Erol Gelenbe · Ricardo Lent *Editors*

Computer and Information Sciences III

27th International Symposium on Computer and Information Sciences

Springer

Computer and Information Sciences III

Erol Gelenbe · Ricardo Lent
Editors

Computer and Information Sciences III

27th International Symposium on Computer and Information Sciences

Springer

Editors
Erol Gelenbe
Department of Electrical and Electronics Engineering
Imperial College
London
UK

Ricardo Lent
Department of Electrical and Electronics Engineering
Imperial College
London
UK

ISBN 978-1-4471-4593-6 ISBN 978-1-4471-4594-3 (eBook)
DOI 10.1007/978-1-4471-4594-3
Springer London Heidelberg New York Dordrecht

Library of Congress Control Number: 2011938586

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The International Symposium on Computer and Information Sciences 2012 held its 27th meeting in Paris on October 3 and 4, 2012, in the prestigious venue of the Institut Henri Poincaré, after having been held for two successive years (2010 and 2011) at the Royal Society in London.

ISCIS 2012 included several invited keynotes and 52 contributed papers that were selected among 81 submissions.

The contents of these proceedings reflect a wide range of relevant topics and timely research within Computer Science and Engineering, in line with the objective of the meeting which is to allow the participants, and the readers of these Proceedings, to obtain a snapshot of work in diverse areas and from many different countries. New areas such as energy management in computer systems and networks, and the use of computer systems to manage energy networks, have recently come to the forefront of research, and these topics are well represented in the Proceedings. Similarly, the security of computer and communication systems is also well-represented in these proceedings, and happens to be one of the most active areas of research.

Each submitted paper was refereed by at least two, often three, or four referees, and the acceptance was based on a numerical ranking provided by the referees themselves. We therefore thank the authors of all submitted papers, as well as the referees and the programme committee members for their hard work and contributions. The resulting Proceedings reflect their collective contributions, and the referee comments have not just selected but also undoubtedly improved the papers that are being presented here.

Erol Gelenbe
Ricardo Lent

Contents

Part I
Smart Systems and Networks

Finite-State Robots in the Land of Rationalia

Arnold L. Rosenberg

Abstract Advancing technologies have enabled simple mobile robots that collaborate to perform complex tasks. Understanding how to achieve such collaboration with *simpler* robots leverages these advances, potentially allowing more robots for a given cost and/or decreasing the cost of deploying a fixed number of robots. This paper is a step toward understanding the algorithmic strengths and weaknesses of robots that are identical mobile *finite-state machines* (*FSM*s)—FSMs being the avatar of "simple" digital computers. We study the ability of (teams of) FSMs to *identify* and *search within* varied-size quadrants of square ($n \times n$) meshes of tiles—such meshes being the avatars of tesselated geographically constrained environments. Each team must accomplish its assigned tasks *scalably*—i.e., in arbitrarily large meshes (equivalently, for arbitrarily large values of n). Each subdivision of a mesh into quadrants is specified via a pair of fractions $\langle \varphi, \psi \rangle$, where $0 < \varphi, \psi < 1$, chosen from a *fixed, finite* repertoire of such pairs. The quadrants specified by the pair $\langle \varphi, \psi \rangle$ are delimited by a horizontal line and a vertical line that cross at *anchor* mesh-tile $v^{(\varphi,\psi)} = \langle \lfloor \varphi(n-1) \rfloor, \lfloor \psi(n-1) \rfloor \rangle$. The current results:

- *A single FSM cannot identify tile $v^{(\varphi,\psi)}$ in meshes of arbitrary sizes, even for a single pair $\langle \varphi, \psi \rangle$—except when $v^{(\varphi,\psi)}$ resides on a mesh-edge.*
- *A pair of identical FSMs can identify tiles $v^{(\varphi_i,\psi_i)}$ in meshes of arbitrary sizes, for arbitrary fixed finite sets of k pairs $\{\langle \varphi_i, \psi_i \rangle\}_{i=1}^{k}$. The pair can sweep each of the resulting quadrants in turn.*
- *Single FSMs can always* verify *(for all pairs and meshes) that all of the tiles of each quadrant are labeled in a way that is unique to that quadrant. This process parallelizes linearly for teams of FSMs.*

Keywords Finite-state mobile robots · Path planning/exploration

A. L. Rosenberg (✉)
College of Computer and Information Science, Northeastern University,
Boston, MA 02115, USA
e-mail: rsnbrg@ccs.neu.edu

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_1,

1 A Motivating Story

MANAGING AGRICULTURE IN RATIONALIA. The state of Rationalia controls its agrarian economy very tightly. Years ago, the state partitioned all arable land into 1×1 *unit-plots* whose (common) physical size is dictated by the demands of the agricultural endeavor: the need to cultivate, plant, and harvest each unit-plot. Each year, after reviewing all farmers' performances (yields, cost efficiencies, etc.), the state aggregates the unit-plots into square plots and allocates to each farmer Φ_i a plot of dimensions $n_i \times n_i$ (measured in unit-plots), where n_i is determined based on Φ_i's past performance. (The unit-plots of each $n_i \times n_i$ plot are indexed from $\langle 0, 0\rangle$ in the northwest corner to $\langle n_i - 1, n_i - 1\rangle$ in the southeast.) Each farmer cultivates crops of the same 4 types, which we label A, B, C, D. (Clerical extensions will handle k crop-types.) As with plot sizes, the government uses past performance to determine how much of each crop-type each farmer should cultivate. Formally, each Φ_i is assigned a fixed pair $\langle \varphi_i, \psi_i\rangle$ of *rational numbers* (what else, given the country's name?), each strictly between 0 and 1. Φ_i's $n_i \times n_i$ plot is then partitioned into quadrants determined by the pair $\langle \varphi_i, \psi_i\rangle$. Each farmer's quadrant NW is devoted to crop-type A, quadrant NE to crop-type B, quadrant SW to crop-type C, and quadrant SE to crop-type D. In detail, each Φ_i's quadrants are specified by passing through her $n_i \times n_i$ plot a horizontal line and a vertical line that cross at the *anchor unit-plot* $v_i = \langle \lfloor \varphi_i(n_i - 1)\rfloor, \lfloor \psi_i(n_i - 1)\rfloor\rangle$, leading to the pattern depicted in Fig. 1. Anchor unit-plot v_i determines where Φ_i's quadrants meet and the crop-types change.

To implement the described system, the government must efficiently partition each farmer's plot into quadrants and sow each quadrant's unit-plots with crops of the appropriate type, achieving the arrangement of Fig. 1. We formalize this organization problem via: the *Anchor-Identification* (*A-I*) *Problem*, which requires the organizing agent(s) to *identify* each farmer's anchor unit-plot; the *Plot-Sweep* (*P-S*) *Problem*, which requires the agent(s) to sweep each of the resulting quadrants, in turn, sowing the authorized type of crop in each. The government faces an additional challenge. Regrettably, some farmers cheat in order to increase their profits, specifically by changing their allocation parameters $\langle \varphi_i, \psi_i\rangle$ in response to the relative profitability of the crop-types. The government must *monitor* each farmer's compliance with her assigned allocation parameters. This is the *Compliance-Checking* (*C-C*) *Problem*.

Complicating the preceding tasks is the government's extreme reluctance to expend money. Therefore, it convened an *elite task force* to determine:

1. How little intelligence do robots need to solve our three Problems?
2. Given the preceding bounds, how few robots suffice to solve the Problems?

	$\lfloor\varphi_i(n_i-1)\rfloor$ columns	$n_i - \lfloor\varphi_i(n_i-1)\rfloor$ columns
$\lfloor\psi_i(n_i-1)\rfloor$ rows	all of type A	all of type B
$n_i - \lfloor\psi_i(n_i-1)\rfloor$ rows	all of type C	all of type D

Fig. 1 The arrangement of Φ_i's crop-types; lengths count unit-plots

The government's operating assumptions are:

- Employing robots would be cheaper than employing humans.
- Less-"intelligent" robots are less expensive to deploy than more capable ones.

In this paper, we play the role of the *elite task force*. We craft a formal setting for the preceding story and study whether robots that have the capabilities (or, "intelligence") of *finite-state machines* (*FSM*s) can accomplish the following formalized versions of the three Problems for arbitrary anchor unit-plots v.

1. *The A-I Problem*: FSM(s) proceed from their initial unit-plots to v.
2. *The P-S Problem*: FSM(s) sweep each quadrant specified by v and label each encountered unit-plot u with its assigned crop-type.
3. *The C-C Problem*: FSM(s) sweep the plot and check that each encountered unit-plot u has the authorized label.

We view FSMs as the lowest level of "intelligence" that might be able to solve the preceding Problems. Informally, we show that:

1. *A single FSM cannot solve the A-I Problem in arbitrary plots—even for a single pair of parameters* $\langle\varphi, \psi\rangle$. *Not obviously, a single FSM can solve the A-I Problem when the anchor unit-plot resides on an edge of the plot.*
2. *A team of* ≥ 2 *identical FSMs can solve the A-I Problem in arbitrary plots, for arbitrary fixed pairs of parameters. Having discovered an anchor unit-plot, the team can solve the P-S Problem for the resulting quadrants.*
3. *A single FSM can solve arbitrary instances of the C-C Problem*; $k > 1$ *identical FSMs can accomplish this k times faster than a single FSM* (*to within rounding*).

2 Technical Background and Related Work

2.1 Technical background. Our model of *FSM-robot* (*FSM*) augments the capabilities of standard finite-state machines (see, e.g., [15]) with the ability to navigate square *meshes* (our story's "plots") of *tiles* (our story's "unit-plots").
Meshes and tiles. Every edge of every tile v is labeled to indicate which of v's potentially four neighbors actually exist. (Labels on tile edges enable FSMs to avoid "falling off" $\mathcal{M}_n$ by moving to a nonexistent tile.) $\mathcal{M}_n$ admits partitions into *quadrants* (labeled NW, NE, SE, SW in clockwise order) that are determined by crossing lines perpendicular to its edges; each partition is determined by an *anchor tile* at which the defining horizontal and vertical line cross.

A single FSM on $\mathcal{M}_n$. At any moment, an FSM $\mathcal{F}$ occupies a single tile of $\mathcal{M}_n$, coresiding with the crop in that tile *but with no other FSM*. At each step, $\mathcal{F}$ can move to any of the (≤ 4) neighbors of its current tile in the primary compass directions: (N)orth, (E)ast, (W)est, (S)outh. (One easily augments $\mathcal{F}$'s move repertoire with any *fixed finite* set of atomic moves.) As $\mathcal{F}$ plans its next move, it *must* consider the label of its current tile—to avoid "falling off" $\mathcal{M}_n$.
Multiple FSMs on $\mathcal{M}_n$. All FSMs operate synchronously, hence, can follow trajectories *in lockstep*. This ability is no less realistic than are human synchronous-start endeavors. FSMs on neighboring tiles can exchange (simple) messages, e.g., "I AM HERE." This enables one FSM to act as an "usher" for others; cf. Sect. 4. FSMs' moves are tightly orchestrated: an FSM attempts to move in direction:

N only at steps $t \equiv 0 \bmod 4$; E only at steps $t \equiv 1 \bmod 4$;
S only at steps $t \equiv 2 \bmod 4$; W only at steps $t \equiv 3 \bmod 4$

(Larger repertoires of atomic moves require larger moduli.) Thereby, *FSMs need never collide!* If several FSMs want to enter a tile from (perforce distinct) neighboring tiles, then one will have permission to enter before the others.

2.2 Algorithmic standards.

- *Algorithms are scalable*. They work on arbitrary-size meshes; FSMs can learn only "finite-state" properties of $\mathcal{M}_n$'s size measures (n, n^2)—e.g., parity.
- *All FSMs are identical*: (a) None has a "name" that renders it unique. (b) All execute the same *finite-state* program; cf. [15, 18].

These standards are often violated in implementations of "ant-like" robots (cf. [8, 11, 17]), where practical simplicity overshadows algorithmic simplicity.

2.3 Related work.

Our study combines ideas from complementary bodies of literature that span several decades. The literature on automata theory and its applications contains studies such as [3–5, 7, 13] that focus on the (in)ability of FSMs to explore graphs with goals such as finding "entrance"-to-"exit" paths or exhaustively visiting all nodes or all edges. Other studies, e.g., [10], focus on algorithms that enable FSMs that populate the tiles of (multidimensional) meshes—the *cellular automaton* model [9]—to synchronize. The robotics literature contains numerous studies—e.g., [1, 2, 8, 17]—that explore ants as a metaphor for simple robots that collaborate to accomplish complex tasks. Cellular automata appear in many application—and implementation-oriented robotic applications of automata-theory [11, 12, 14, 17]. The current study melds the automata-theoretic and robotic points of view by studying FSMs that traverse square meshes, with goals more closely motivated by robotics than automata theory. Our closest precursor is [16], which requires each FSM in a mesh to *park*, i.e., go to its closest corner and organize with other FSMs there into a maximally compact formation.

3 The Anchor-Identification Problem

The *Anchor-Identification* (*A-I*) *Problem* for $\mathcal{M}_n$ is formalized as follows.

Input: A pair of rationals $\langle \varphi, \psi \rangle$, where $0 < \varphi, \psi < 1$
Task: All FSMs on $\mathcal{M}_n$ proceed from their initial tiles to anchor tile $v^{(\varphi,\psi)}$. One FSM ends on $v^{(\varphi,\psi)}$; all others cluster around it.

3.1 The general A-I Problem for one FSM. *A single FSM cannot solve the A-I Problem on arbitrarily large meshes, for any input pair* $\langle \varphi, \psi \rangle$. The proof exploits a result from [16] that exposes the inability of single FSMs to navigate the *interiors* of large meshes, i.e., submeshes that are bounded away from the edges.

3.2 The Edge-Constrained A-I Problem for one FSM. Not obviously, a single FSM *can* solve the variant of the A-I Problem wherein the sought anchor-tile resides on an edge of $\mathcal{M}_n$. Formally, this Problem for FSM $\mathcal{F}$ is:

Input: A rational φ with $0 < \varphi < 1$

Task: $\mathcal{F}$ proceeds from its initial tile to:
- bottom version: $\langle (n-1), \lfloor \varphi(n-1) \rfloor \rangle$
- top version: $\langle 0, \lfloor \varphi(n-1) \rfloor \rangle$
- left version: $\langle \lfloor \varphi(n-1) \rfloor, 0 \rangle$
- right version: $\langle \lfloor \varphi(n-1) \rfloor, (n-1) \rangle$

Theorem 1 *For any fixed rational* $0 < \varphi < 1$*: A single FSM* $\mathcal{F}^{(\varphi)}$ *whose size depends only on* φ *can solve the Edge-Constrained A-I Problem with input* φ *on any mesh* $\mathcal{M}_n$*, within* $O(n)$ *steps.*

Sketch. Let $\varphi = a/b$ for integers $b > a > 0$. $\mathcal{F}^{(\varphi)}$ moves from v_{s} to v_{h}: it goes to $\langle 0, 0 \rangle$ and continues thence to v_{h} via a diagonal walk of *super-steps*. The *bottom-edge* walk has slope $-b/a$; the *right-edge* walk has slope $-a/b$ (Fig. 2). □

3.3 The A-I Problem for teams of ($\geq$2) identical FSMs.

Theorem 2 *Let* $\Psi = \{\langle \varphi_i, \psi_i \rangle\}_{i=1}^{k}$ *be any fixed set of rational pairs, where* $0 < \varphi_i, \psi_j < 1$ *for all* i, j*. One can design an FSM* $\mathcal{F}^{(\Psi)}$ *such that a team of two or more copies of* $\mathcal{F}^{(\Psi)}$ *can solve the A-I Problem in every mesh* $\mathcal{M}_n$*, for every pair* $\langle \varphi, \psi \rangle \in \Psi$*, within* $O(n)$ *synchronous steps.*

Sketch. We use Theorem 1 to design identical FSMs $\mathcal{F}_1$ and $\mathcal{F}_2$ that solve the A-I Problem for a rational pair $\langle \varphi, \psi \rangle$. See Fig. 3. $\mathcal{F}_1$ and $\mathcal{F}_2$ meet at tile $\langle 0, 0 \rangle$, then execute the algorithm of Theorem 1 to go to the projections of anchor $v^{(\varphi,\psi)}$: $\mathcal{F}_1$ goes to the left-edge anchor $v_{\mathrm{h},1}$; $\mathcal{F}_2$ goes (*in lockstep*) to the top-edge anchor $v_{\mathrm{h},2}$. When $\mathcal{F}_1$ reaches $v_{\mathrm{h},1}$ (resp., $\mathcal{F}_2$ reaches $v_{\mathrm{h},2}$), it starts to walk eastward (resp., delays one step, then starts to walk southward). The FSMs halt when they meet: $\mathcal{F}_1$ is then on tile $v^{(\varphi,\psi)}$; $\mathcal{F}_2$ is on $v^{(\varphi,\psi)}$'s northward neighbor. □

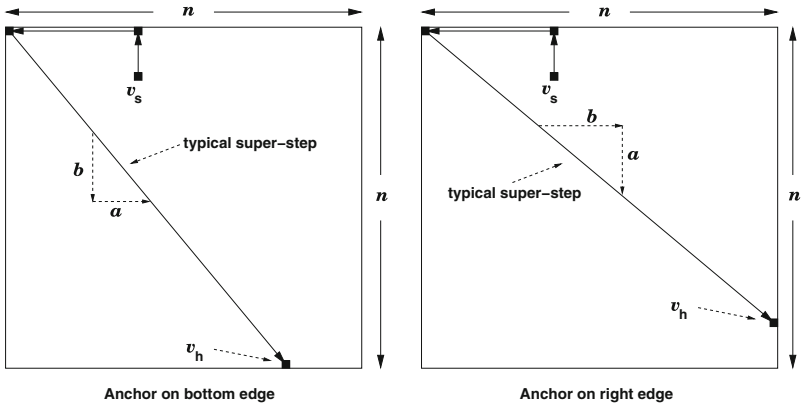


Fig. 2 $\mathcal{F}(\varphi)$ proceeds from tile v_{s} to the anchor tile v_{h} to solve: (*left*) the bottom-edge-Constrained A-I Problem and (*right*) the right-edge version, both with input $\varphi = a/b$

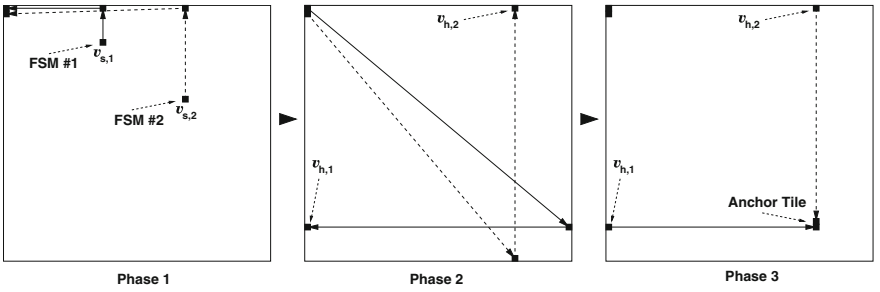


Fig. 3 The 3-phase coordinated trajectories for FSMs $\mathcal{F}_1$ and $\mathcal{F}_2$ to solve the A-I Problem with inputs $\langle \varphi, \psi \rangle$. *Solid lines* show $\mathcal{F}_1$'s trajectory; *dashed lines* show $\mathcal{F}_2$'s

4 The Plot-Sweep Problem

A pair of FSMs sweep through each quadrant specified by anchor tile v, in turn.

Theorem 3 *For any rational pair $\langle \varphi, \psi \rangle$, where $0 < \varphi, \psi < 1$, one can design an FSM $\mathcal{F}^{(\varphi,\psi)}$ such that team of two copies of $\mathcal{F}^{(\varphi,\psi)}$ can solve the P-S Problem in every mesh $\mathcal{M}_n$.*

Sketch. Focus on a sweep of quadrant NE. Once $\mathcal{F}_1$ and $\mathcal{F}_2$ identify anchor tile $v^{(\varphi,\psi)}$, $\mathcal{F}_1$ moves to $v^{(\varphi,\psi)}$ and $\mathcal{F}_2$ to $v^{(\varphi,\psi)}$'s eastward neighbor (stage 0 of Fig. 4). Thence, $\mathcal{F}_1$ climbs the column that extends northward from $v^{(\varphi,\psi)}$ and acts as an "usher" while $\mathcal{F}_2$ threads the rest of the quadrant; see Fig. 4. □

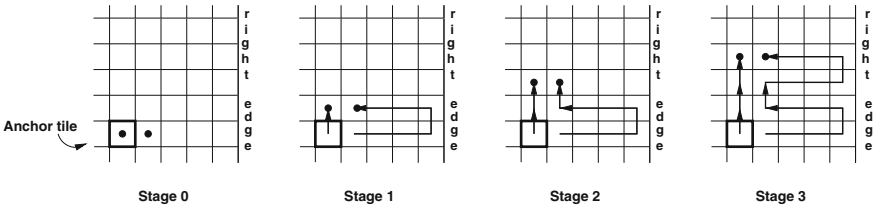


Fig. 4 Starting a sweep of $\mathcal{M}_n$'s NE quadrant. The *left* FSM "ushers" the *right* one

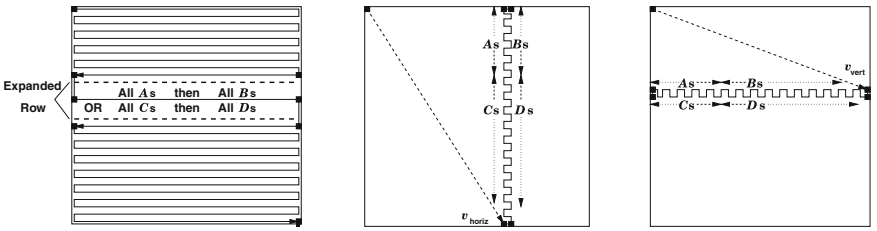


Fig. 5 (*left*) $\mathcal{F}^{(\varphi,\psi)}$ verifies the form of each row's labels: (*left* to *right*) all As followed by all Bs *OR* all Cs followed by all Ds. (*middle*, *right*) $\mathcal{F}^{(\varphi,\psi)}$ verifies that labeled quadrants have the correct horizontal and vertical endpoints

5 Compliance-Checking by a Single FSM

Our final task is the *Compliance-Checking* (*C-C*) *Problem* for $\mathcal{M}_n$:

Input: A pair or rationals $\langle \varphi, \psi \rangle$, where $0 < \varphi, \psi < 1$

Task: The FSM(s) on $\mathcal{M}_n$ perform a sweep from tile $\langle 0, 0 \rangle$, to check that the tile labels have the format illustrated in Fig. 1.

Theorem 4 *A team of $k \geq 1$ identical FSMs can solve the C-C Problem for any pair of rationals $\langle \varphi, \psi \rangle$, in any mesh $\mathcal{M}_n$, within $\frac{1}{k}n^2 + O(n)$ steps.*

Sketch. An FSM $\mathcal{F}^{(\langle \varphi,\psi \rangle)}$ can check that boundaries are as in Fig. 1 in two phases. Using a *row-sweep*, $\mathcal{F}^{(\langle \varphi,\psi \rangle)}$ easily checks that each row's crop-labels belong to $A^\star B^\star$ or to $C^\star D^\star$; see Fig. 5(left) for the case $k = 1$. (Larger teams use the Edge-Constrained A-I algorithm to partition the columns evenly among them.) In the *boundary* phase, $\mathcal{F}^{(\langle \varphi,\psi \rangle)}$ finds the *edge-constrained projections* of anchor tile $v^{(\varphi,\psi)}$, viz., tiles $v_{\text{horiz}} = \langle \lfloor \varphi(n-1) \rfloor, n-1 \rangle$ and $v_{\text{vert}} = \langle n-1, \lfloor \psi(n-1) \rfloor \rangle$. Sawtooth trajectories northward from v_{horiz} and westward from v_{vert} enable $\mathcal{F}^{(\varphi,\psi)}$ to verify all boundaries [Fig. 5(middle, right)]. □

6 Conclusions

6.1 Retrospective. We have gained new understanding of the algorithmic strengths and weaknesses of finite-state robots as they navigate square meshes. We have contrasted the powers of a single FSM versus teams of ≥ 2 identical FSMs on three basic problems, each specified by a pair of fractions $\langle \varphi, \psi \rangle$ that specify the *anchor* tile $v^{(\varphi,\psi)}$ in any $n \times n$ mesh $\mathcal{M}_n$. Each anchor specifies a partition of $\mathcal{M}_n$ into quadrants. The *Anchor-Identification* (*A-I*) Problem has FSMs move to $v^{(\varphi,\psi)}$; the *Plot-Sweep* (*P-S*) Problem has FSMs sweep through each of the quadrants specified by $v^{(\varphi,\psi)}$; the *Compliance-Checking* (*C-C*) Problem has FSMs verify that every tile of $\mathcal{M}_n$ has a label that is unique to its quadrant. Single FSMs cannot solve the A-I or P-S Problems, but they can solve the C-C Problem; teams of ≥ 2 identical FSMs can solve all three problems, and they can speed up the C-C Problem linearly via parallelism. All problem solutions are *scalable*: a single FSM design works for all meshes. FSMs can sometimes use $\mathcal{M}_n$'s edges to *appear* to count to n, even though unbounded counting is impossible. The P-S and C-C Problems combine to show that single FSMs can sometimes *check* patterns that they are unable to generate.

6.2 Sample extensions. (1) One can generalize Theorem 3 to *sweep nonsquare submeshes*; e.g., if the "usher" FSM follows a diagonal trajectory, then the team sweeps a trapezoidal region. (2) One can *personalize the A-I Problem* so that FSM $\mathcal{F}^{(\varphi,\psi)}$ moves to the "copy" of tile $v^{(\varphi,\psi)}$ in $\mathcal{F}^{(\varphi,\psi)}$'s starting quadrant.

6.3 Prospective. The *Parking Problem* for FSMs in [16] focuses on the question "What can FSMs discover about where they reside within $\mathcal{M}_n$?" The current study, especially the A-I Problem, focuses on the question "How well can FSMs discover designated target tiles within $\mathcal{M}_n$?" An obvious goal for future research would be to extend the definitions of "where they reside" and "designated target tile." A valuable source of inspiration are robotic studies such as [1, 8, 11]. Another direction for the future would be to go beyond pure path planning/exploration by designing FSMs that can scalably find and transport "food" and that can avoid obstacles, inspired by, e.g., [2, 6, 8, 11, 14].

References

1. Adler, F., Gordon, D.: Information collection and spread by networks of patrolling ants. Am. Nat. **140**, 373–400 (1992)
2. Basu, P., Redi, J.: Movement control algorithms for realization of fault-tolerant ad hoc robot networks. IEEE Network **18**(4), 36–44 (2004)
3. Bender, M., Slonim, D.: The power of team exploration: two robots can learn unlabeled directed graphs. In: Proceedings of the 35th IEEE Symposium on Foundations of Computer Science, pp. 75–85 (1994)
4. Blum, M., Sakoda, W.: On the capability of finite automata in 2 and 3 dimensional space. In: Proceedings of the 18th IEEE Symposium on Foundations of Computer Science, pp. 147–161 (1977)

5. Budach, L.: On the solution of the labyrinth problem for finite automata. Elektronische Informationsverarbeitung und Kybernetik (EIK) **11**(10–12), 661–672 (1975)
6. Chen, L., Xu, X., Chen, Y., He, P.: A novel FSM clustering algorithm based on Cellular automata. IEEE/WIC/ACM International Conference on Intelligent Agent Technology (2004)
7. Cohen, R., Fraigniaud, P., Ilcinkas, D., Korman, A., Peleg, D.: Label-guided graph exploration by a finite automaton. ACM Trans. Algorithms **4**(4), 1–18 (2008)
8. Geer, D.: Small robots team up to tackle large tasks. IEEE Distrib. Syst. Online **6**(12), 2 (2005)
9. Goles, E., Martinez, S. (eds.): Cellular Automata and Complex Systems. Kluwer, Amsterdam (1999)
10. Gruska, J., La Torre, S., Parente, M.: Optimal time and communication solutions of firing squad synchronization problems on square arrays, toruses and rings. In: Calude, C.S, Calude, E., Dinneen, M.J. (eds.) Developments in Language Theory. Lecture Notes in Computer Science, vol. 3340, pp. 200–211. Springer, Heidelberg (2004)
11. http://www.kivasystems.com/
12. Marchese, F.: Cellular automata in robot path planning. In: Proceedings of the EUROBOT'96, pp. 116–125 (1996)
13. Müller, H.: Endliche automaten und labyrinthe. Elektronische Informationsverarbeitung und Kybernetik (EIK) **11**(10–12), 661–672 (1975)
14. Rosenberg, A.L.: Cellular ANTomata: path planning and exploration in constrained geographical environments. Adv. Complex Syst. (2012) (to appear)
15. Rosenberg, A.L.: The Pillars of Computation Theory: State, Encoding, Nondeterminism. Universitext Series. Springer, Heidelberg (2009)
16. Rosenberg, A.L.: Ants in parking lots. In: Proceedigs of the 16th International Conference on Parallel Computing (EURO-PAR'10), Part II. Lecture Notes in Computer Science, vol. 6272, pp. 400–411. Springer, Heidelberg (2010)
17. Russell, R.: Heat trails as short-lived navigational markers for mobile robots. In: Proceedings of the International Conference on Robotics and Automation, pp. 3534–3539 (1997)
18. Spezzano, G., Talia, D.: The CARPET programming environment for solving scientific problems on parallel computers. Parallel Distrib. Comput. Pract. **1**, 49–61 (1998)

Cognitive Packets in Large Virtual Networks

Ricardo Lent and Erol Gelenbe

Abstract Network testbeds are useful for protocol performance evaluation. They overcome most challenges commonly involved in live network experimentation, retaining fair realism and repeatability under controllable conditions. However, large-scale testbeds are difficult to set up due to limited resources available to most experimenters. In this paper, we explore the use of hardware virtualisation as an experimental tool to improve resource efficiency, allowing to boost the effective number of nodes available for network testing. By virtualising network routers and links, a cluster environment with off-the-shelf equipment can host hundreds of virtual routers for large-scale network testing. We apply this technique to construct a 800-node Cognitive Packet Networks (CPN) testbed that provides insight into the benefits and limitations of the approach.

1 Introduction

Testbed experimentation with network prototypes can complement, and sometimes replace modelling and simulation studies [9], bringing a closer-to-reality alternative to evaluation studies in network research. Repeatable experiments can stress working conditions for protocol implementations to verify operation aspects and performance limitations. Cognitive packet network (CPN) routing [3] has been evaluated in several testbeds [4, 5, 7], to verify the effectiveness of the approach. These tests have provided useful feedback to improve the prototype and algorithm, and

R. Lent (✉) · E. Gelenbe
Department of Electrical and Electronic Engineering,
Intelligent Systems and Networks, Imperial College,
London SW7 2AZ, UK
e-mail: r.lent@imperial.ac.uk

E. Gelenbe
e-mail: e.gelenbe@imperial.ac.uk

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_2,

inspiration for new research. However, tests have been limited so far to relatively small networks, of tens of nodes (less than 100 in all cases) connected with a proportional number of links. Tests on larger networks are always desirable, not only because they permit observations of the network operation under conditions closer to production systems, but also useful for debugging and measuring practical limits of given implementations. Despite its drawbacks for network performance testing (e.g., unrealistic packet delays), virtualisation can be a viable alternative for creating a high number of virtual devices, which could be used for large-scale network testing under certain conditions. It is interesting to note that forms of virtualisation have been supported by networks for a while through diverse techniques, such as Virtual LANs (VLANs) and tunneling (e.g, L2TP, PPTP). Virtual machine managers on the other hand, such as VMware, VirtualBox and QEMU/KVM, support network virtualisation for guest machines providing emulated network domains. Some recent proposals have suggested conceptual extensions to virtualisation to improve the control and management of physical networks [2, 11]. Unfortunately, these techniques are of limited use to handle in practice the thousands of links and nodes that a large-scale testbed might require. For example, in the case of VLANs, the VLAN ID (12 bits) in IEEE 802.1p limits the identification of 4,096 VLANs. In practice, most commercial switches support about 250 VLANs. Similarly, current virtual machine hypervisors usually limit the number of virtual networks to 8. While likely enough to satisfy the networking needs of most sites, these numbers are clearly too modest for our purposes. On the other hand, several initiatives, such as FIRE (http://www.ict-fire.eu) and GENI [1] seek to build remotely accessible large experimental facilities. A few notable examples of existing facilities are Emulab [15] and PlanetLab. Having remote access to testbeds is an advantage but their shared nature creates restrictions to the kind and size of experiments that could be carried out.

Recent work has addressed optimisation issues resulting from embedding virtual topologies in physical ones [17]. For a single site, a simpler approach is to leverage platform virtualisation to create router instances in a similar way machine instances can be created with an hypervisor. Unlike the approach in [10], our plan is to enforce logical links without tunnels which provides greater scalability. Naturally, because of the virtual nature of the network, packet traveling times may not accurately follow those on a physical network [16]. Therefore, a careful selection of network metrics will be needed to produce meaningful results from experiments under virtualisation. For clarity, we will elaborate our approach to network testing with virtualisation within the specific context of Cognitive Packet Networks and apply it to evaluate the network under router misbehaviour. The approach is however easily applicable to a wider range of network tests.

2 Workflows in a Virtual Network

Setting up a new virtual network involves two well-defined parts: creating nodes and enforcing the virtual topology (i.e., creating edges). For the experiment, we make use of machine virtualisation to instantiate network nodes given that the existing CPN

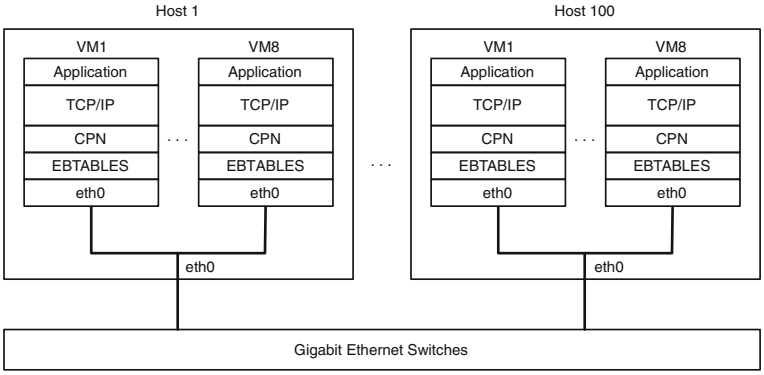


Fig. 1 Virtual network testbed for CPN

prototype works as a Linux kernel module. The module requires a virtualisation platform able to create virtual machine (VM) instances that are capable of running a Linux kernel. Because of the reduced size of the prototype—about 64 KB, Linux and the CPN module can easily run in virtual machines with RAM disks. Using RAM disks can greatly simplify the deployment process because no disk images would need to be created across the physical testbed. Via the Preboot Execution Environment (PXE), machines instances can quickly obtain both a kernel with the CPN extension and a RAM disk image from a server machine. Alternatively, at node creation time, full disk images must be created (typically, replicated from a master copy). In practice, we developed UNIX scripts to automate virtual machine creation with full pre-copied images. The script allowed a rapid creation of virtual routers, at about 100 machines per minute.

Assuming the use of a single physical network for virtual machine interconnection with bridged networking, edge virtualisation can be implemented through packet filtering, similarly to [12]. The only constraint is that virtual machines would need to be created with a consistent and predictable MAC address, which could be enforced in the VM creation process. An arbitrary network topology can then be implemented through bride firewalling, by setting appropriate filtering rules at each node to replicate the desired topology. Our approach relies on `EBTABLES`, which can enforce firewall rules at the MAC layer. The resulting process becomes transparent to all networking layers above the MAC layer, including the CPN layer (see Fig. 1). We were able to test this approach with hundreds of firewall rules without a noticeable drop in performance.

2.1 A 800-Router CPN Testbed

The topology was modeled after the Internet autonomous-system (AS) graph collected by the Internet maps [18] project. We selected a map of 2009 because of its

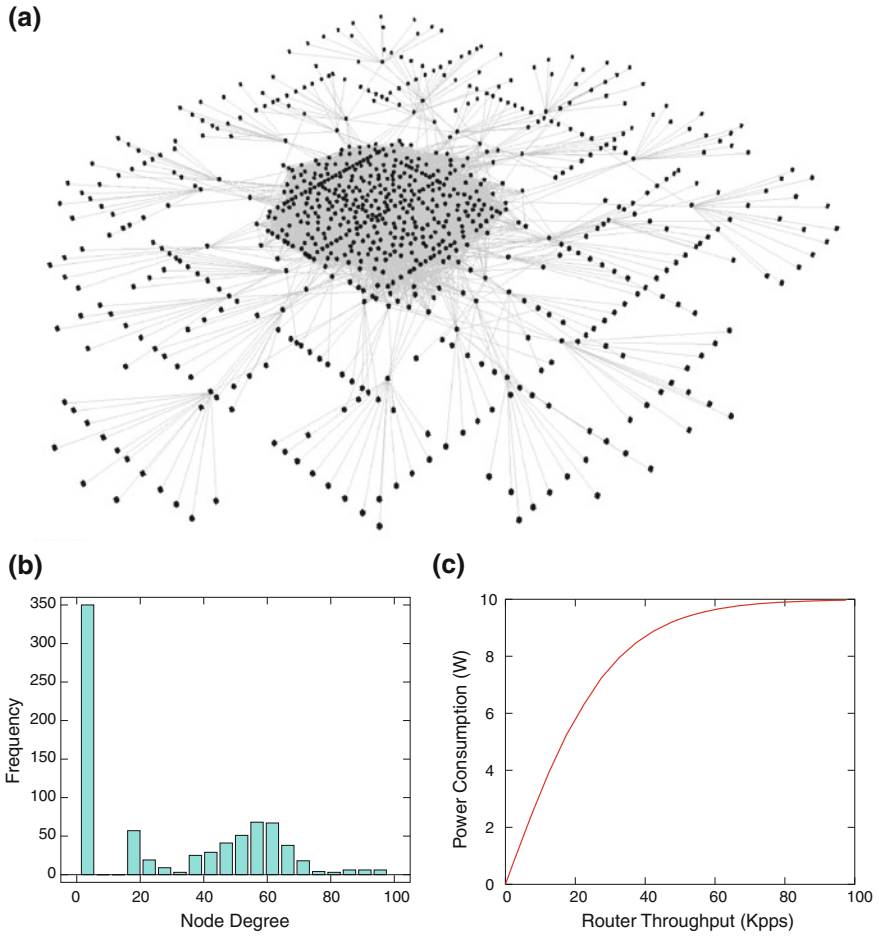


Fig. 2 A 800-node CPN testbed. **a** Topology. **b** Node degree distribution. **c** Power profile of routers

smaller size compared to current maps. To define the topology, we included all Tier 1 and large Internet Service Provider nodes from the map collection, each represented by a CPN router. Nodes with very large number of neighbors were transformed into mesh subnets for practical reasons (i.e., because of the maximum number of neighbours that a CPN node could handle). In addition, 350 end nodes were connected to randomly selected routers. The whole network consisted of 800 nodes and 11,762 links forming the topology shown in Fig. 2a. Node degree (i.e., number of neighbours) frequency is plotted in Fig. 2b. Clearly, end hosts (350 of them) have degree 1 and routers have varying number of neighbours, up to 100, which was the limit assumed in the large node division. The physical system consisted of 100 identical machines (Fig. 1): 8-core Xeon X3450 2.67 GHz and 8 GB of RAM with Gigabit

Ethernet connections through Cisco Nexus 2148T switches forming a star topology with 10 Gbps connections to a central switch—a Cisco Nexus 5010. Each physical machine hosted exactly 8 virtual machines (VM) with a VirtualBox hypervisor. Each VM was configured as a single-core machine with 256 MB of RAM. Identical copies of a disk image containing an extended Ubuntu Linux kernel with a CPN implementation were attached to each virtual machine. The disk image also contained a topology file and scripts to enforce a MAC-layer filtering based on the MAC address of each node and expected neighbours listed in the topology file. Therefore, each node configuration was customized based on its MAC address assigned at VM creation time. MAC addresses followed a certain convention which allowed node to use them in determining their IP address (for control) and CPN address (for testing). In addition, each CPN router was associated with a (simulated) power consumption dependent on the packet total throughput handled by the node as depicted in Fig. 2c. Such response can be measured by profiling the system with a power measuring tool [13].

2.2 Studying Router Misbehavior

To evaluate the testbed, we applied it to measure CPN performance under router misbehaviour, which could result from nodes that have been compromised by an intruder. Such nodes would behave in an undesirable manner, affecting user flows. We are interested in quantifying how such flows could be affected. To conduct the evaluation, we considered either 50, 100 or 150 concurrent flows for any given experiment run, randomly selecting nodes from the set of 350 end nodes that we defined when creating the network topology. Similarly, we select a certain number of routers to misbehave during the experiment to be either 5, 23, 45, 90 or 180 for any particular run. In addition, we define a target rate for each of the flows to be 10, 50, 100, 150, 200, 250 or 300 pps with 1 KB-long packets. All flows are UDP without retransmission at upper layers. Two types of router misbehaviour were examined. In the first case, misbehaving routers drop all incoming traffic (including smart packets), so that those routers could potentially break network connectivity. This is the simplest case of network intrusion where the attacker has managed to disable the router. In a second case, we consider that the misbehaving router will send user packets to the worst possible next hop as a possible consequence of a more elaborate attack. Because multiple virtual machines share the same physical resources (e.g., the same network port), latency measurements could not reflect the actual values on a corresponding physical network. We have selected power consumption in nodes [6–8] as a metric for the routing goal of smart packets. Power consumption in real routers has a significant idle component and a dynamic component, being the latter quite relevant for software routers [14]. For testing purposes, we zeroed the idle component (see Fig. 2c) to direct the routing optimisation towards the lowest dynamic consumption due to user flows. Measurements for the first type of router misbehaviour are depicted in Fig. 3 comparing the level of average power consumption observed in the network under

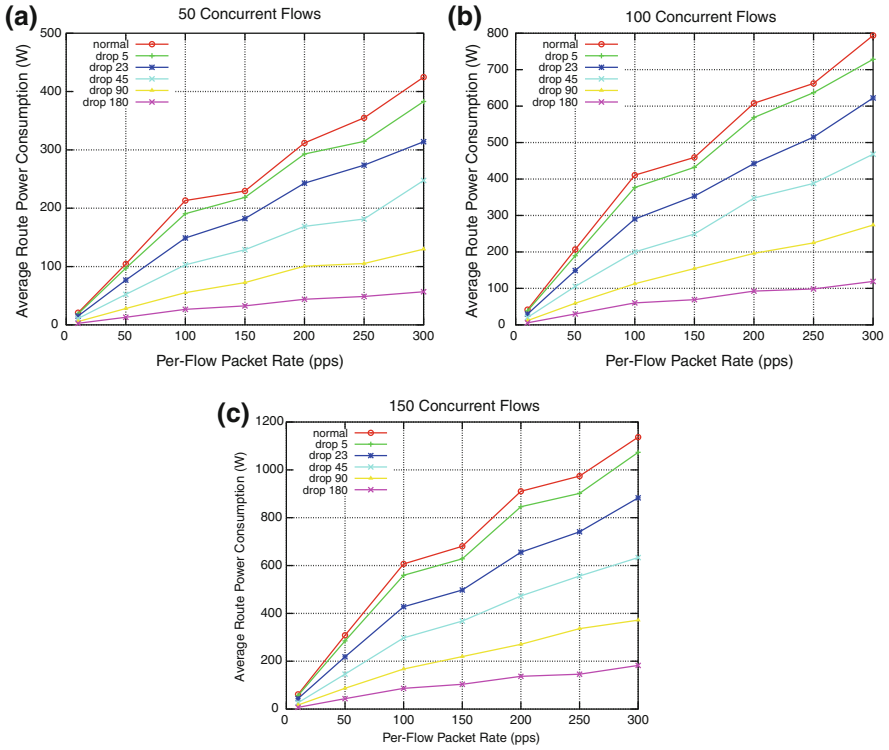


Fig. 3 Average power consumption of paths

different number of misbehaving routers. The normal behaviour is also included (top curve) for comparison purposes. Randomly choosing routers to drop packets may cause network partition, which explains the high packet loss that can be observed in cases of drop misbehaviour.

A higher number of packet-drop nodes tends to diminish the overall network power consumption because of the higher level of packet loss as can be inferred from Fig. 4b. This is because no packet retransmission was considered. On the other hand, we observed path lengths to be approximately similar across different combinations of number of flows. However, there was a significant difference in the path length of packets between the two types of router misbehaviour, likely expected given the differences in the effective network topology.

It is important to note that the purpose of the tests was mainly the validation of the CPN prototype in a large network setting rather than providing a specific solution to a network intrusion problem. Nevertheless, we observed that CPN could easily handle to some success (all but network partitions), these types of router misbehavior without any change in the algorithm. In the case of packet drops, smart packets could avoid misbehaving nodes because they are quickly removed from the path discovering process whenever they move to a compromised router. Similarly, smart packets tried

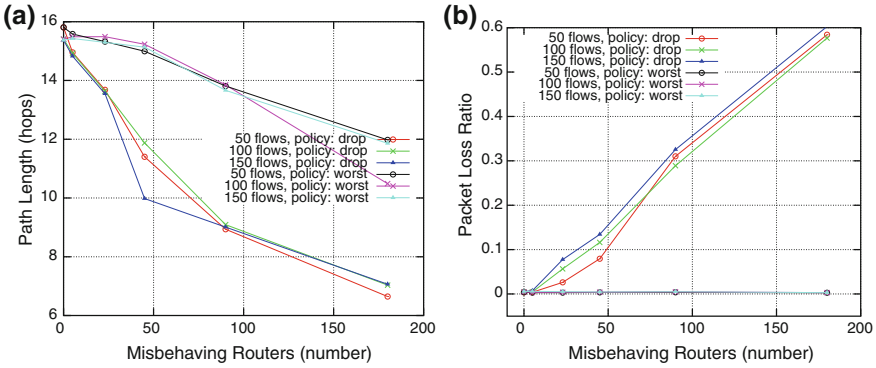


Fig. 4 Average path length and packet loss ratio. **a**, **b** System A

to avoid routers which forward packets to the worst next hop because of their low rating reflected in mailboxes.

3 Conclusions

We have successfully tested CPN in a large network of 800 routers which suggests a viable direction to create even larger testbeds in the future. Two practical considerations resulted from our experiments. First, the CPN implementation was able to handle up to 250 concurrent neighbours per node. This limit is mainly due to the use of static memory allocation in the CPN kernel implementation. While the neighbour table implementation could be modified to make use of dynamic memory allocation and increase this limit, we observed that the main bottleneck was rather the high memory usage of the random neural network (see [5]) which requires two matrices of size the square number of neighbours. A possible solution is to dynamically assign neighbours to the matrix, for example, in a least-recently used fashion, so as to limit its size by only considering the most used (or successful) neighbours in the table. A future work will address these implementation issues in further detail.

Acknowledgments This work has been possible in part thanks to the support provided by the UK Technology Strategy Board/EPSRC SATURN Project, the FP7 FIT4Green Project and the computing facilities at Northrup-Grumman UK Ltd.

References

1. Anderson, T., Reiter, M.K.: Geni: global environment for network innovations distributed services working group (2006). http://www.homes.cs.washington.edu/~tom/support/GDD-06-24.pdf
2. Casado, M., Koponen, T., Ramanathan, R., Shenker, S.: Virtualizing the network forwarding plane. In: Proceedings of the Workshop on Programmable Routers for Extensible Services of Tomorrow, PRESTO '10, pp. 8:1–8:6. ACM, New York (2010)

3. Gelenbe, E.: Cognitive packet network. US Patent 6,804,201, 2004
4. Gelenbe, E.: Steps towards self-aware networks. Commun. ACM **52**(7), 66–75 (2009)
5. Gelenbe, E., Lent, R., Xu, Z.: Design and performance of cognitive packet networks. Perform. Eval. **46**(2, 3), 155–176 (2001)
6. Gelenbe, E., Mahmoodi, T.: Energy-aware routing in the cognitive packet network. In: International Conference on Smart Grids, Green Communications, and IT Energy-Aware Technologies (Energy 2011), Venice, Italy 2011
7. Gelenbe, E., Morfopoulou, C.: A framewok for energy aware routing in packet networks. Comput. J. **54**(6), 850–859 (2011)
8. Gelenbe, E., Morfopoulou, C.: Gradient optimisation for network power consumption. In: GreenNets 2011, The First International Conference on Green Communications and Networking, Colmar, France 2011
9. Gelenbe, E., Stafylopatis, A.: Global behavior of homogeneous random neural systems. Appl. Math. Model. **15**(10), 534–541 (1991). doi:10.1016/0307-904X(91)90055-T
10. Jiang, X., Xu, D.: Vbet: a vm-based emulation testbed. In: Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research, MoMeTools '03, pp. 95–104. ACM, New York (2003)
11. Keller, E., Rexford, J.: The "platform as a service" model for networking. In: Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking, INM/WREN'10, p. 4. USENIX Association, Berkeley (2010)
12. Lent, R.: Design of a MANET testbed management system. Comput. J. **49**(4), 171–179 (2006)
13. Lent, R.: A sensor network to profile the electrical power consumption of computer networks. In: Proceedings of GLOBECOM Workshops (GC Wkshps), pp. 1433–1437, Miami, 2010. doi:10.1109/GLOCOMW.2010.5700175
14. Lent, R.: Simulating the power consumption of computer networks. In: Proceedings of the 15th IEEE International Workshop on Computer Aided Modeling Analysis and Design of Communication Links and Networks (IEEE CAMAD), Miami, 2010
15. White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S., Newbold, M., Hibler, M., Barb, C., Joglekar, A.: An integrated experimental environment for distributed systems and networks. In: Proceedings of the Fifth Symposium on Operating Systems Design and Implementation, pp. 255–270. USENIX Association, Boston (2002)
16. Whiteaker, J., Schneider, F., Teixeira, R.: Explaining packet delays under virtualization. SIGCOMM Comput. Commun. Rev. **41**(1), 38–44 (2011)
17. Xin, Y., Baldine, I., Mandal, A., Heermann, C., Chase, J., Yumerefendi, A.: Embedding virtual topologies in networked clouds. In: Proceedings of the 6th International Conference on Future Internet Technologies, CFI '11, pp. 26–29. ACM, New York (2011)
18. Zhang, B., Liu, R., Massey, D., Zhang, L.: Collecting the internet as-level topology. SIGCOMM Comput. Commun. Rev. **35**, 53–61 (2005). doi:10.1145/1052812.1052825. http://www.doi.acm.org/10.1145/1052812.1052825

A Novel Unsupervised Method for Securing BGP Against Routing Hijacks

Georgios Theodoridis, Orestis Tsigkas and Dimitrios Tzovaras

Abstract In this paper, a BGP hijack detection mechanism is presented. The proposed methodology is utterly unsupervised and no assumptions are made whatsoever, but it is developed upon the extraction of two novel features related to the frequency of appearance and the geographic deviation of each intermediate AS towards a given destination country. The technique is tested under a real-world case of BGP hijack and the efficiency of the features and the corresponding proximity measures is assessed. It is proven that the proposed approach is capable of decisively capturing such events of malicious routing path anomalies.

Keywords BGP · Hijack · Security · Detection · Routing · Anomalies

1 Introduction

According to its fully decentralized structure, Internet is established as the sum of numerous administrative regions called Autonomous Systems (ASes), which are interconnected through the existing backbone on the basis of the interdomain routing protocol, i.e. Border Gateway Protocol (BGP) [1]. BGP is responsible for maintaining and communicating the routing directives among the Internet comprising entities and hence defining the actual operational network topology.

This work has been partially supported by the European Commission through project FP7-ICT-257495-VIS-SENSE funded by the seventh framework program. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission.

G. Theodoridis (✉) · O. Tsigkas · D. Tzovaras
Centre for Research and Technology Hellas, Information Technologies Institute,
6th km Charilaou-Thermi Road, P.O. Box 60361, 57001 Thessaloniki, Greece
e-mail: gtheo@iti.gr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_3,

Routing paths undergo continuous alterations as a result of hardware failures and the varying inter-AS relationships. In consequence, the volume of BGP activity is significantly increased, while phenomena of intense BGP disturbances also appear at high frequency. Additionally, due to the fact that it is developed upon the concept of mutual trust, BGP suffers from inherent security vulnerabilities, which can severely compromise its functionality and the integrity of the routing paths. In this context, cyber attacks against BGP have nowadays emerged as one of the most prominent Internet threats. Hence, given the key role of BGP in conjunction with the utmost importance of Internet, it becomes an urging necessity to develop the adequate mechanism that would allow for the effective detection and root cause analysis of potentially malicious BGP anomalies.

In this context, significant research activity has been drawn on the detection of BGP anomalies [2]. Ballani et al. perform a thorough study of the prefix hijacking mechanisms with special emphasis on the interception cases [3], while Gao aims at inferring the inter-AS relationships by solely utilizing the raw BGP data [4]. One of the most common approaches is based on the extraction and unsupervised statistical analysis of the most appropriate features [5, 6]. Alternatively, in a semi-supervised learning manner, data mining upon well known BGP behavior is performed, so as to calibrate the detection algorithms [7, 8]. Several monitoring systems have also been developed that build a view of the AS/prefix topology and report on any alterations against this state of Refs. [9, 10]. Furthermore, the *Argus* system focuses specifically on cases of traffic blackholing, by combining info from both the control and the data-plane [11].

Towards this ultimate goal of decisively capturing and attributing any abnormal routing alterations, a completely novel methodology is hereby presented, which introduces two significant novelties. First, it is completely unsupervised, requiring no a-priori knowledge of the BGP dynamics. Second, it manages to efficiently make virtue of the underlying geo-spatial coherence of the Internet routing information by grouping the BGP activity on a per destination-country basis; to this aim two new feature related to each AS's frequency of appearance and geographic divergence are extracted.

The paper is structured as follows. Section 2 describes the proposed methodology in detail, while in Sect. 3 the technique is implemented and evaluated in a real-world scenario. Finally, Sect. 4 summarizes the paper.

2 Proposed Methodology for AS-Path Hijack Detection

A primary threat against BGP is related to the case that a malicious AS announces itself as an intermediate hop towards an already occupied prefix. As a result, all the IP traffic of the victim prefix is compelled to traverse the attacking AS, in order to be either blackholed or intercepted.

More precisely, let $\mathbf{W^{M,p}}$ be the initial AS-Path connecting the monitoring AS (M) with the destination AS (O^p) that owns prefix p and let $\mathbf{W'^{M,p}}$ be an alternative AS-Path announced at a later time instant.

$$\mathbf{W^{M,p}} = \{M, A_1^{M,p}, \ldots, A_K^{M,p}, O^p\} \tag{1}$$

$$\mathbf{W'^{M,p}} = \{M, A_1'^{M,p}, \ldots, A_{K'}'^{M,p}, O^p\} \tag{2}$$

Then, $\mathbf{F^M}(\mathbf{W}, \mathbf{W}')$ denotes the set of non-common ASes for $\mathbf{W^{M,p}}$ and $\mathbf{W'^{M,p}}$:

$$\mathbf{F^M}(\mathbf{W}, \mathbf{W}') = \mathbf{W^{M,p}} \cup \mathbf{W'^{M,p}} - \mathbf{W^{M,p}} \cap \mathbf{W'^{M,p}} \tag{3}$$

An AS-Path anomaly is identified if and only if $\mathbf{F^M}(\mathbf{W}, \mathbf{W}') \neq \emptyset$, i.e. there is at least one non-common intermediate AS between the competing routes. Each non-common AS, $A_h \in \mathbf{F^M}(\mathbf{W}, \mathbf{W}')$, is suspicious of performing BGP hijacking. If $A_h \in \mathbf{AP^{M,p'}}$, the later announcement is the root cause of the hijacking. On the contrary, if $A_h \in \mathbf{AP^{M,p}}$, then the BGP anomaly event is related to an attempt of the prefix's owner (O^p) to restore the normal path.

Hence, in order to be able to evaluate the normality of an AS-Path alteration, it is necessary to assess the legitimacy of each AS's appearance within the announced AS-Paths. To this end, two primary features are extracted, while, different similarity measures are utilized for each one of them, in order to capture the root causes of the BGP activity. In more detail, let $\mathbf{C} = \{C_1, \ldots, C_D\}$ be the set of all the country-entities identified in Internet. Then, for each country $C_d \in \mathbf{C}$, $\mathbf{I^d}$ is introduced as the set of all the intermediate ASes that are traversed in order to reach C_d from M for any prefix $p \in \mathbf{P^d}$, where $\mathbf{P^d}$ is the set of all the prefixes hosted by ASes located in C_d and $C(X)$ denotes the country of origin of AS X.

$$\mathbf{I^d} = \bigcup_{k=1}^{K} \left\{A_k^{M,p}\right\}, \quad \forall p \in \mathbf{P^d}, \quad C(A_k^{M,p}) \neq C(M), \quad C(A_k^{M,p}) \neq C^d \tag{4}$$

Namely, $\mathbf{I^d} = \{I_1^d, \ldots, I_Q^d\}$ comprises the Q different ASes that have ever appeared as intermediate hops between $C(M)$ and C^d. Although, the sequence of intermediate hops towards each prefix's owner AS is dependent on the monitoring point, the notation M will be hereafter omitted for ease of reference.

Furthermore, $\mathbf{U^d}$ is introduced as the set of all the countries that are recorded as intermediate hops towards hosts of C_d, i.e. $\mathbf{U^d}$ is the set of all the countries of origin of the ASes comprising $\mathbf{I^d}$.

$$\mathbf{U^d} = \{U_1^d, \ldots, U_Z^d\} = \bigcup_{q=1}^{Q} \left\{C(I_q^d)\right\}, \quad \mathbf{U^d} \subseteq \mathbf{C} \tag{5}$$

The driving notion behind the definition of $\mathbf{I^d}$ and $\mathbf{U^d}$ resides in the necessity to quantitatively define the legitimacy of an AS's occurrence as an intermediate hop. More concisely, in order to draw safe conclusions concerning the legitimacy of an announced AS-Path, it would optimally be required to retain the complete history of the specific prefix and all the involved ASes. Nevertheless, the frequency of a route's announcement cannot be regarded as a safe criterion for judging its normality, since any alternative route announced for the first time would be always condemned to be labelled as suspicious. Moreover, in the case that the competing paths are interchangeably announced, the respective routes will be found to present comparably high probability values and thus the event would be disregarded, despite the fact that such phenomena may correspond to repeating hijacks-responses. Additionally, an analysis performed at per prefix/AS level would impose substantial memory and processing overhead.

On the contrary, by aggregating the BGP activity on per country level, the proposed methodology exploits the inherent geo-spatial coherence of the Internet infrastructure and routing policies. Specifically, following the inter-AS agreements, for every destination-country there is a finite, semi-constant set of Intermediate-Countries (ICs) that provide its connectivity with the rest of the world. Hence, any path alterations involving ICs that are not common for the specific destination-country are bound to raise significant suspicions of interception. Additionally, special emphasis must be laid on the fact that such malicious Internet activity is usually carried out by (or through) remote hosts, in order to decrease the probability of being tracked down as well as to escape any legal actions and countermeasures. In this context, the two features that form the cornerstone of the proposed technique are presented below.

2.1 Probability of an IC's Appearance

For every country $C_d \in \mathbf{C}$, the vector $\mathbf{V_C^d}$ is estimated, which contains the number of appearances of each IC across the path towards hosts of the destination-country C^d.

$$\mathbf{V_C^d} = \begin{bmatrix} N(U_1^d) \\ \vdots \\ N(U_Z^d) \end{bmatrix} \tag{6}$$

Utilizing $\mathbf{V_C^d}$, *the probability* $(B(U_z^d))$ *of a country's appearance in a path towards destination-country* C_d is introduced, in order to allow for the quantitative assessment of the legitimacy of an AS's appearance.

$$B(U_z^d) = \frac{N(U_z^d)}{\sum_{z=1}^{Z}\{N(U_z^d)\}}, \quad \forall U_z^d \in \mathbf{U^d} \tag{7}$$

The calculated probability is equal to the conditional probability that an AS X from country $C(X)$ appears within a routing path for a prefix hosted in C_d $\left(B(U_z^d) = PR\left[X \in \mathbf{AP^p} | (C(O^p) = C_d)\right]\right)$ and hence it is equal to the fraction of the $C(X)$'s appearances towards country C_d against the aggregate appearances of all the ICs for C_d. $B(U_z^d)$ is a measure of how frequently U_z^d serves as an intermediate hop for C_d and therefore how commonly expected is for U_z^d to appear in a BGP announcement that refers to a C_d host.

2.2 Geographic Disparity of a Route's ASes

Routing algorithms generally opt for the path with the lowest latency and thus the minimization of the end-to-end geographic distance is a primary routing objective. Hence, apart from specific agreements/policies or infrastructural malfunctions, there is no operational reason for selecting routes that significantly diverge from the direct route. Furthermore, cyber criminal activity is anticipated to originate from remote countries with favourable legal system and/or international relationships. Consequently, profound geographic divergence along the routing path can be safely regarded as enough evidence for raising an alarm. In this context, in order to numerically approach the geographic anomaly imposed by the appearance of an AS along a BGP route, two measures are defined.

Geographic length introduced by each IC. $\forall U_z^d \in \mathbf{U^d}$, the geographic length $(L(U_z^d))$ introduced by U_z^d in reference to the ideal direct path is defined as:

$$L(U_z^d) = \frac{L(C(M), U_z^d) + L(U_z^d, C_d)}{L(C(M), C_d)} \tag{8}$$

where $L(C_x, C_y)$ is the geographic distance between countries C_x and C_y. $L(U_z^d)$ is the geographic length of the $C(M) \rightarrow U_z^d \rightarrow C_d$ path, normalized against the length of the direct source-destination link. The estimation of $L(U_z^d)$ allows for tracing the countries and thus the corresponding ASes that prominently diverge from the expected path.

Z-Score of the an IC's geographic length. According to the definition of the Z-Score,

$$S^L(U_z^d) = \frac{L(U_z^d) - E[L(U_z^d)]}{\sigma[L(U_z^d)]} \tag{9}$$

where $E[L(U_z^d)]$ and $\sigma[L(U_z^d)]$ are respectively the mean value of all $L(U_z^d), \forall U_z^d \in \mathbf{U^d}$. The statistical analysis of potential geographic anomalies on the basis of Z-Score, allows to assess each IC's role in comparison with the general deviations of the routing path for the specific destination-country under investigation. In particular,

- $S^L(U_z^d) < (>)\ 0$: The geographic divergence introduced by U_z^d is lower (higher) than the average distance of all the perceived routing paths between the $C(M)$ and C_d.
- $|S^L(U_z^d)| \uparrow$ AND $S^L(U_z^d) > 0$: U_z^d's geographic location significantly distances from the average route, while there are only few alternative countries towards C_d that are located far away from the direct route.

2.3 Implementation Issues

Despite its solid theoretical background, the aforementioned methodology is associated with a real-world implementation complication. A substantial fraction of the intermediate hops along an announced AS-Path are higher-tier ASes that provide connectivity across multiple countries/continents. However, despite their global presence, higher-tier ASes are uniquely identified by a sole country of origin, which usually coincides with the location of the enterprise's headquarters. Therefore, including the higher-tier ASes in the calculation of $\mathbf{U^d}$ shall utterly distort the overall statistical analysis.

Let us assume the example of AS-Path {AS15469, AS174, AS6762, AS8966, AS13224}. By including AS174 (tier-1 AS) in $\mathbf{I^d}$ (4), it will be erroneously taken for granted that it is usual for IP traffic originating from Switzerland (CH) to traverse USA (US) so as to reach Kenya (KE). As a result, the efficiency of the BGP hijack detection mechanism will be severely degraded, since: (i) announcements that include higher-tier ASes established in US would be erroneously considered as suspicious due to the high $L(US)$ value and (ii) BGP hijacks executed by ASes located in countries hosting higher-tier ASes would remain unobserved.

In this respect, a subset $\mathbf{I'^d}$ of $\mathbf{I^d}$ is defined that comprises all the ASes that belong to $\mathbf{I^d}$ and which are not classified as higher-tier ASes. Correspondingly, $\mathbf{U'^d}$ is also defined on the basis of $\mathbf{I'^d}$ (5). Eventually, $B(U_z^d)$, $L(U_z^d)$ and $S^L(U_z^d)$ are calculated for the set $\mathbf{U'^d}$. However, for simplicity, the initial notation will be kept. Finally, it must be underlined that this exclusion of the higher-tier ASes does not compromise the reliability of the BGP hijack mechanism, since higher-tier ASes are not expected to deploy criminal activity.

2.4 Transition to AS-Path Anomaly Level

The metrics $B(U_z^d)$, $L(U_z^d)$ and $S^L(U_z^d)$ are defined on a per IC level. Thus, in order to study BGP hijacks, it is necessary to implement the proposed methodology at AS-Path anomaly level. According to (3), each AS-Path anomaly comprises two competing AS-Paths, while in turn, each AS-Path is identified by a sequence of ICs. In this context, let $\mathbf{F_j}$ be the set of non-common ASes $\mathbf{F}(\mathbf{W}, \mathbf{W'})$ involved in the

AS-Path anomaly j. Then, considering that suspicions of malicious BGP activity are raised for ASes with low values of $B(U_z^d)$, and high values of $L(U_z^d)$ and $S^L(U_z^d)$, the corresponding scores at AS-Path level are defined:

- Country Appearance Probability per AS-Path anomaly (CAP)

$$CAP_j = min\{B(C(A_f))\}, \quad \forall A_f \in \mathbf{F_j} \tag{10}$$

- Country geographic length per AS-Path anomaly (CGL)

$$CGL_j = max\{L(C(A_f))\}, \quad \forall A_f \in \mathbf{F_j} \tag{11}$$

- Z-Score of country geographic length per AS-Path anomaly ($CGLZ$)

$$CGLZ_j = max\{S^L(C(A_f))\}, \quad \forall A_f \in \mathbf{F_j} \tag{12}$$

where $A_f \in \mathbf{I^d}$ and $C(A_f) \in \mathbf{U^d}$ and $\mathbf{J}$ are all the monitored AS-Path anomalies.

3 Evaluation Under Real-World BGP Hijack Events

On August 20, 2011, a Russian telecommunication company, hereafter referred to as victim-AS (AS_V), reported to the North American Network Operators Group (NANOG) that five of its prefixes had been hijacked. The prefixes' ownership was not affected, but forged routes were announced that dictated any traffic to traverse through the hijacking AS (AS_H), which is located in US, for interception purposes. As a countermeasure, AS_V responded on August 24, by announcing longer subprefixes with the correct paths. All AS-Path anomalies taking place on that date are recorded and, for each one, the values of CAP, CGL and $CGLZ$ are calculated. The response of the AS_V triggers an AS-Path anomaly as a result of the juxtaposition of the new legitimate route (rather straight path) against the existing path (detour through US) previously forged by AS_H. The raw BGP data are obtained through the RIPE repository [12] and AS15469, which is situated in CH (Vantage Point *rrc00*), is chosen as the monitoring point (M).

The distributions of CAP and CGL are presented in Fig. 1, so as to assess their capability to efficiently discriminate the bulk of AS-Path anomalies. From Fig. 1a it becomes apparent that the vast majority of the AS-Path anomalies involve rather common ICs, while, according to Fig. 1b, almost 98 % of the AS-Path anomalies involve ICs that do not introduce additional geographic path length higher than the direct source-destination distance ($CGL < 2$). Figure 2a, b presents the scatter plot of CAP against CGL and $CGLZ$ respectively. With the solid black triangle it is marked the AS-Path anomaly that corresponds to the incident under investigation. As it becomes apparent, taking into account the CAP in conjunction with $CGLZ$ of all the monitored BGP path alterations, the proposed methodology is capable of

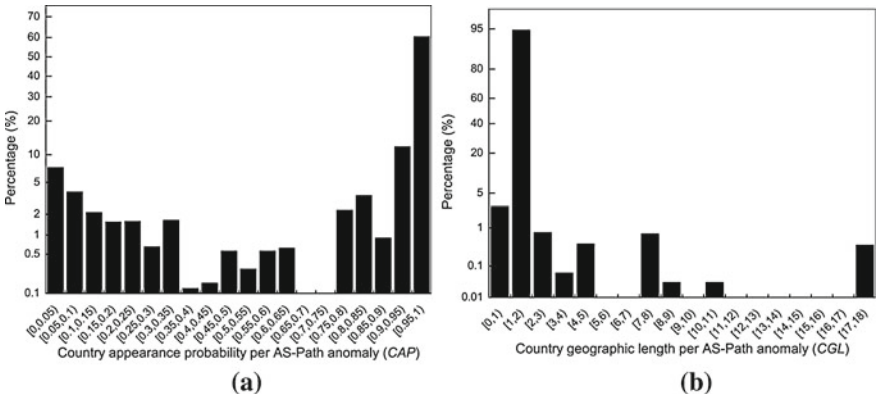


Fig. 1 Distribution of (**a**) *CAP* and (**b**) *CGL*

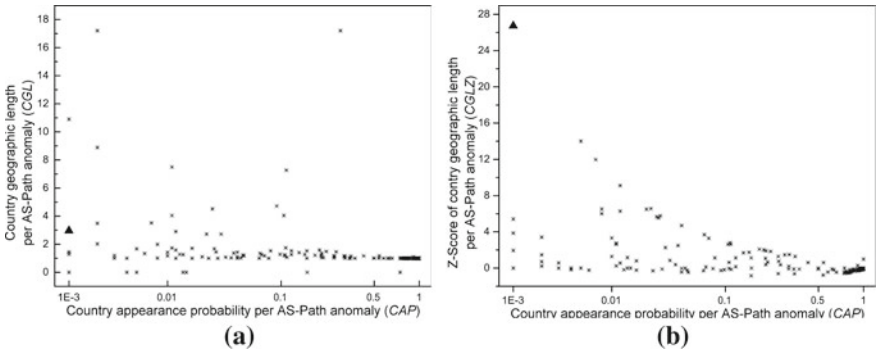


Fig. 2 Scatter plots between proximity measures of different features. **a** Scatter plot of *CAP* against *CGL*. **b** Scatter plot of *CAP* against *CGLZ*

decisively pinpointing the BGP hijack. $CGLZ$ is chosen instead of CGL since it incorporates the overall distribution of the routes' geographic length.

4 Conclusions

In this paper, a fully-unsupervised methodology for the detection of malicious AS-Path anomalies has been described. The novelty of the proposed technique lies within the basic notion of statistically analysing the BGP activity on a per hosting-country basis, so as firstly to make virtue of the inherent geo-spatial coherence of the Internet infrastructural functionality and secondly to formulate a solid background regarding the normal routing status. In this context, two completely novel features are extracted and corresponding proximity measures are defined, while no assump-

tions whatsoever are being introduced. The presented mechanism is implemented and evaluated against a positively known event of AS-Path hijack that has been publicly reported. Through this real-world case study, the behavior of each feature and metric are studied and assessed thoroughly and eventually the efficiency of the proposed methodology in capturing BGP attacks is proven.

References

1. Rekhter, Y., Li, T.: A border gateway protocol 4 (bgp-4). IETF RFC 1771. http://www.ietf.org/rfc/rfc1771.txt
2. Sriram, K., Borchert, O., Kim, O., Gleichmann, P., Montgomery, D.: A comparative analysis of bgp anomaly detection and robustness algorithms. In: Proceedings of the CATCH '09, (March 2009)
3. Ballani, H., Francis, P., Zhang, X.: A study of prefix hijacking and interception in the internet. In: Proceedings of the SIGCOMM '07, Kyoto (2007)
4. Gao, L.: On inferring autonomous system relationships in the internet. IEEE/ACM Trans. Netw. **9**(6), 733–745 (2001)
5. Deshpande, S., Thottan, M., Ho, T.K., Sikdar, B.: An online mechanism for bgp instability detection and analysis. IEEE Trans. Comput. **58**(11), 3296–3304 (2009)
6. Zhang, K., Yen, A., Zhao, X., Massey, D., Felix Wu, S., Zhang, L.: On detection of anomalous routing dynamics in bgp. Lecture Notes in Computer Science. Springer, Berlin (2004)
7. Li, J., Dou, D., Wu, Z., Kim, S., Agarwal, V.: An internet routing forensics framework for discovering rules of abnormal bgp events. In: Proceedings of the SIGCOMM '05, (Oct. 2005)
8. de Urbina Cazenave, I., Kosluk, E., Ganiz, M.: An anomaly detection framework for bgp. In: Proceedings of the INISTA '11, (June 2011)
9. Lad, M., Massey, D., Pei, D., Wu, Y., Zhang, B., Zhang, L.: PHAS: A prefix hijack alert system. USENIX Security Symposium, In (2006)
10. Qiu, J., Gao, L., Ranjan, S., Nucci, A.: Detecting bogus bgp route information: going beyond prefix hijacking. In: Proceedings of the SecureComm '07, (Sept. 2007)
11. Xiang, Y., Wang, Z., Yin, X., Wu, J.: Argus: An accurate and agile system to detecting ip prefix hijacking. In: Proceedings of the IEEE ICNP '11, (Oct. 2011)
12. Ripe, NCC. http://www.ripe.net/datatools/stats/ris/ris-raw-data

Learning Equilibria in Games by Stochastic Distributed Algorithms

Olivier Bournez and Johanne Cohen

Abstract We consider a family of stochastic distributed dynamics to learn equilibria in games, that we prove to correspond to an Ordinary Differential Equation (ODE). We focus then on a class of stochastic dynamics where this ODE turns out to be related to multipopulation replicator dynamics.Using facts known about convergence of this ODE, we discuss the convergence of the initial stochastic dynamics. For general games, there might be non-convergence, but when the convergence of the ODE holds, considered stochastic algorithms converge towards Nash equilibria. For games admitting a multiaffine Lyapunov function, we prove that this Lyapunov function is a super-martingale over the stochastic dynamics and that the stochastic dynamics converge. This leads a way to provide bounds on their time of convergence by martingale arguments. This applies in particular for many classes of games considered in literature, including several load balancing games and congestion games.

1 Introduction

Consider a scenario where agents learn from their experiments, by small adjustments. This might be for example about choosing their telephone companies, or about their portfolio investments. We are interested in understanding when the whole market can converge towards rational situations, i.e. Nash equilibria in the sense of game

The authors are supported in part by the ANR SHAMAN project.

O. Bournez
LIX-CNRS, 91128 Palaiseau Cedex, France
e-mail: bournez@lix.polytechnique.fr

J. Cohen (✉)
PRiSM-CNRS Univ. Versailles, 45 av. des Etats-Unis, 78000 Versailles, France
e-mail: Johanne.Cohen@prism.uvsq.fr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_4,

theory. This is natural to expect dynamics of adjustments to be stochastic, and fully distributed, since we expect agents to adapt their strategies based on their local knowledge of the market.

Several such dynamics of adjustments have been considered recently in the literature. Up to our knowledge, this has been done mainly for deterministic dynamics or best-response based dynamics: computing a best response requires a global description of the market. Stochastic variations, avoiding a global description, have been considered. However, considered dynamics are somehow rather ad-hoc, in order to get efficient convergence time bounds. We want to consider here more general dynamics related to (possibly perturbed) replicator dynamics, and discuss when one may expect convergence.

Basic game theory framework. Let $[n] = \{1, \ldots, n\}$ be the set of players. Every player i has a set $\mathscr{S}_i$ of *pure strategies*. Let m_i be the cardinal of $\mathscr{S}_i$. A *mixed strategy* $q_i = (q_{i,1}, q_{i,2}, \ldots, q_{i,m_1})$ corresponds to a probability distribution over pure strategies: pure strategy ℓ is chosen with probability $q_{i,\ell} \in [0, 1]$, with $\sum_{\ell=1}^{m_i} q_{i,\ell} = 1$. Let K_i be the simplex of mixed strategies for player i. Any pure strategy ℓ can be considered as mixed strategy e_ℓ, where vector e_ℓ denotes the unit probability vector with ℓth component unity, hence as a corner of K_i.

Let $K = \prod_{i=1}^{n} K_i$ be the space of all mixed strategies. A *strategy profile* $Q = (q_1, \ldots, q_n) \in K$ specifies the strategies of all players: q_i corresponds to the mixed strategy of player i. In game theory, we often write $Q = (q_i, Q_{-i})$, where Q_{-i} denotes the vector of the strategies played by all other players. We admit games whose payoffs may be random: we assume that each player i gets a random *cost* of expected value $c_i(Q)$. In particular, the expected cost for player i for playing the pure strategy e_ℓ is denoted by $c_i(e_\ell, Q_{-i})$.

Some classes of games. Several games where players' costs are based on the shared usage of a common set of resources $[m] = \{1, 2, \ldots, m\}$ where each resource $1 \leq r \leq m$ has an associated nondecreasing cost function denoted by $C_r : [n] \to \mathbb{R}$, have been considered in algorithmic game theory literature.

In *load balancing games* [9], the machines are the resources, and the players (task) choose a machine to execute: each player i has a weight w_i. The cost for player i under profile of pure strategies (assignment) $Q = (q_1, \ldots, q_n)$ corresponds to $c_i(Q) = C_{q_i}(\lambda_{q_i}(Q))$, where $\lambda_r(Q)$ is the load of machine r: $\lambda_r(Q) = \sum_{j:q_j=r} w_j$. In *congestion games* [13], the players compete for subsets of $[m]$. Hence, the pure strategy space $\mathscr{S}_i$ of player i is a subset of $2^{[m]}$ and a pure strategy $q_i \in Q$ for player i is a subset of $[m]$ resources. The cost of player i under profile of pure strategies Q corresponds to $c_i(Q) = \sum_{r \in q_i} C_r(\lambda_r(Q))$ where $\lambda_r(Q)$ is the number of q_j with $r \in q_j$.

Ordinal and potential games. All these classes of games can be related to potential games introduced by [11]: A game is an *ordinal potential game* if there exists some function ϕ from *pure* strategies to $\mathbb{R}$ such that for all pure strategies Q_{-i}, q_i, and q_i', one has $c_i(q_i, Q_{-i}) - c_i(q_i', Q_{-i}) > 0$ iff $\phi(q_i, Q_{-i}) - \phi(q_i', Q_{-i}) > 0$. It is an *an exact potential game* if for all pure strategies Q_{-i}, q_i, and q_i', one has $c_i(q_i, Q_{-i}) - c_i(q_i', Q_{-i}) = \phi(q_i, Q_{-i}) - \phi(q_i', Q_{-i})$.

2 Stochastic Learning Algorithms

We consider fully distributed algorithms of the following form where b is a positive real parameter close to 0.

Let $Q(t) = (q_1(t), \ldots, q_n(t)) \in K$ denote the state of all players at instant t. Our interest is in the asymptotic behavior of $Q(t)$, and its possible convergence to Nash equilibria. Functions $F_i^b(c_i(t), s_i(t), q_i(t))$ is defined as generic as possible, maintaining that the $q_i(t)$ always stays validity probability vectors. We only assume that $G_i(Q) = \lim_{b\to 0} \mathbb{E}[F_i^b(c_i(t), s_i(t), q_i(t))|Q(t)]$ is always defined and that G_i is continuous.

- Initially, $q_i(0) \in K_i$ can be any vector of probability, for all i.
- At each round t,
  - Any player i selects strategy $\ell \in \mathscr{S}_i$ with probability $q_{i,\ell}(t)$. This leads to a cost $c_i(t)$ for player i.
  - Select some player $i(t)$: player $i(t)$ is selected with probability p_i, with $\sum_{i=1}^n p_i = 1$.

  This player $i(t)$ updates $q_i(t)$ as follows: $q_i(t+1) = q_i(t) + bF_i^b(c_i(t), s_i(t), q_i(t))$;
  Any other player keeps $q_i(t)$ unchanged: $q_i(t+1) = q_i(t)$.

Results. In the general case (Theorem 1), any stochastic algorithm in the considered class converges (see in [2]) weakly towards solutions of initial value problem (ordinary differential equation (ODE)) $\frac{dq_i}{dt} = p_i G_i(Q)$, given $Q(0)$.

A *replicator-like dynamics* F_i^b is a dynamic where $F_i^b(c_i(t), s_i(t), q_i(t)) = \gamma(c_i(t))\,(q_i(t) - e_{s_i(t)}) + \mathscr{O}(b)$, where $\gamma : \mathbb{R} \to [0, 1]$ is some *decreasing* function with value in $[0, 1]$. We assume all costs to be positive, by linearity of expectation then all costs must be bounded by some constant M, and we can take $\gamma(x) = \frac{M-x}{M}$.

We can admit randomly perturbed dynamics: $\mathscr{O}(b)$ denotes some perturbation that stay of order of b. A *perturbed replicator-like dynamic* is of the form

$$F_i^b(r_i(t), s_i(t), q_i(t)) = \mathscr{O}(b) + \begin{cases} \gamma(r_i(t))(q_i(t) - e_{s_i(t)}) & \text{with probability } \alpha \\ b(q_i(t) - e_{s_j}) & \text{with probability } 1-\alpha, \\ & \text{where } j \in \{1, \ldots, m_i\} \\ & \text{is chosen uniformly,} \end{cases}$$

where $0 < \alpha < 1$ is some constant.

We prove that such dynamics have a mean-field approximation which is isomorphic to a multipopulation replicator dynamics. We claim (Theorem 2), that for general games, *if* there is convergence of the mean-field approximation, then stable limit points will correspond to Nash equilibria. Notice, that there is no reason that the convergence of mean-field approximation holds for generic games. We note (Theorem 3)

that the ordinal games are *Lyapunov* games: their mean-field limit approximation admits some Lyapunov function. Furthermore, we show that for Lyapunov games with multiaffine Lyapunov function, the Lyapunov function is a super-martingale over stochastic dynamics. Finally, we deduce results on the convergence of stochastic algorithms for this class.

For lack of space, we refer to [2] for missing proofs.

Related work. A potential game always have a pure Nash equilibrium: since ordinal potential function, that can take only a finite number of values, is strictly decreasing in any sequence of pure strategies strict best response moves, such a sequence must be finite and must lead to a Nash equilibrium [13].This is clear that an (exact) potential game is an ordinal potential game. Congestion games, and hence load balancing games are known to be particular potential games [13].

For load-balancing games, the bounds on the convergence time of best-response dynamics have been investigated in [5]. Since players play in turns, this is often called the *Elementary Stepwise System*. Other results of convergence in this model, have been investigated in [7, 10], but they require some global knowledge of the system in order to determine what next move to choose. A Stochastic version of best-response dynamics has been investigated in [1]. For congestion games, the problem of finding pure Nash equilibria is PLS-complete [8]. Efficient convergence of best-response dynamics to approximate Nash equilibria in particular symmetric congestion games have been investigated in [3] in the case where each resource cost function satisfies a *bounded jump assumption*.

All previous discussions are about best-response dynamics. A stochastic dynamic, not elementary stepwise like ours, but close to those considered in this paper, has been partially investigated in [12] for general games and for potential games: It is proved to be weakly convergent to solutions of a multipopulation replicator equation. Some of our arguments follow theirs, but notice that their convergence result (Theorem 3.1) is incorrect: convergence may happen towards non-Nash (unstable) stationary points. Furthermore, this is not clear that any super-martingale argument holds for such dynamics, as our proof relies on the fact that the dynamics is elementary stepwise.

Replicator equations have been deeply studied in evolutionary game theory [15]. Evolutionary game theory has been applied to routing problems in the Wardrop traffic model in [6]. Potential games have been generalized to continuous player sets in [14]. They have be shown to lead to multipopulation replicator equations, and since our dynamics are not about continuous player sets, but lead to similar dynamics, we borrow several constructions from [14]. No time convergence discussion is done in [14]. Moreover, in [4], a replicator equation for the routing games and for particular allocation games are studied to converge to a pure Nash equilibrium.

3 Mean-Field Approximation For Generic Algorithms

We focus on the evolution of $Q(t)$, where $Q(t) = (q_1(t), \ldots, q_n(t))$ denotes the strategy profile at instant t in the stochastic algorithm. Clearly, $Q(t)$ is an homogeneous Markov chain. Define $\Delta Q(t)$ as $\Delta Q(t) = Q(t+1) - Q(t)$, and $\Delta q_i(t)$ as

$q_i(t+1) - q_i(t)$. We can write

$$\mathbb{E}[\ \Delta q_i(t)\ |Q(t)\ ] = bp_i\mathbb{E}[\ F_i^b(c_i(t), s_i(t), q_i(t))\ |Q(t)\ ], \tag{1}$$

with $G_i(Q) = \lim_{b\to 0}\mathbb{E}[\ F_i^b(c_i(t), s_i(t), q_i(t))\ |Q(t)\ ]$ assumed to be continuous under our hypotheses.

Convergence of the stochastic algorithms towards ODEs defining their mean-field limit approximation can be formalized as follows: Consider the piecewise-linear interpolation $Q^b(.)$ of $Q(t)$ defined by $Q^b(t) = Q(\lfloor t/b\rfloor) + (t/b - \lfloor t/b\rfloor)$ $(Q(\lfloor t/b+1\rfloor) - Q(\lfloor t/b\rfloor))$. Function $Q^b(.)$ belongs to the space of all functions from $\mathbb{R}$ into K which are right continuous and have left hand limits (*cad-lag functions*). Now consider the sequence $\{Q^b(.) : b > 0\}$. We are interested in the limit $Q(.)$ of this sequence when $b \to 0$. Recall that a family of random variable $(Y_t)_{t\in\mathbb{R}}$ weakly converges (see in [2]) to a random variable Y, if $E[h(X_t)]$ converges to $E[h(Y)]$ for each bounded and continuous function h.

Theorem 1 *The sequence of interpolated processes $\{Q^b(.)\}$ converges weakly, when $b \to 0$, to $Q(.)$, which is the solution of initial value problem*

$$\frac{dq_i}{dt} = p_i G_i(Q), i = 1, \ldots, n, \quad with\ Q(0) = Q^b(0). \tag{2}$$

4 General Games and Replicator-Like Dynamics

Now, we restrict to (possibly perturbed) replicator-like dynamics, as defined in page 3. For any such dynamic (full details in [2]), Eq. (2) leads to the following ordinary differential equation which turns out to be (a rescaling of) (multipopulation) classical replicator dynamic

$$\frac{dq_{i,\ell}}{dt} = p_i q_{i,\ell}(c_i(q_i,\, Q_{-i}) - c_i(e_\ell,\, Q_{-i})), \tag{3}$$

whose limit points are related to Nash equilibria (see in [2]). Using properties of dynamics (3), we get:

Theorem 2 *For general games, for any replicator-like or perturbed replicator-like dynamic, the sequence of interpolated processes $\{Q^b(.)\}$ converges weakly, as $b \to 0$, to the unique deterministic solution of dynamic (3) with $Q(0) = Q^b(0)$. If the mean-field approximation dynamic (3) converges, its stable limit points correspond to Nash equilibria of the game.*

More precisely (see in [2]), the following are true for solutions of dynamic (3): (i) All Nash equilibria are stationary points. (ii) All stable stationary points are Nash equilibria. (iii) However, (unstable) stationary points can include some non-Nash equilibria.

Actually, all corners of simplex K are stationary points, as well as, from the form of (3), more generally any state Q in which all strategies in its support perform equally well. Such a state Q is not a Nash equilibrium as soon as there is an not used strategy (i.e. outside of the support) that performs better.

Unstable limit stationary points may exist for the mean-field approximation. Consider for example a dynamics that leave on some face of K where some well-performing strategy is never used.To avoid "bad" (non-Nash equilibrium, hence unstable) stationary points, following the idea of penalty functions for interior point methods, one can use as in Appendix A.3 of [14] some patches on the dynamics that would guarantee Non-complacency (see in [2]). *Non-Complacency (NC)* is the following property: $G(Q) = 0$ implies that Q is a Nash equilibrium (3) (i.e. stationarity implies Nash).

For general games, we get that the limit for $b \to 0$ is some ordinary differential equation whose stable limit points, when $t \to \infty$, *IF* there exist, can only be Nash equilibria. Hence, *IF* there is convergence of the ordinary differential equation, then one expects the previous stochastic algorithms to learn equilibria.

5 Lyapunov Games, Ordinal and Potential Games

Since general games have no reason to converge, we propose now to restrict to games for which replicator equation dynamic or more generally general dynamics (2) is provably convergent.

Definition 1 (**Lyapunov Game**). We say that a game has a *Lyapunov function* (with respect to a particular dynamic (2) over K), or that the game is Lyapunov, if there exists some non-negative $\mathscr{C}^1$ function $F : K \to \mathbb{R}$ such that for all i, ℓ and Q, whenever $G(Q) \neq 0$,

$$\sum_{i,\ell} p_i \frac{\partial F}{\partial q_{i,\ell}}(Q) G_{i,\ell}(Q) < 0. \tag{4}$$

Lyapunov games include ordinal potential games : we will say that a Lyapunov function $F : K \to \mathbb{R}$ is multiaffine, if it is defined as polynomial in all its variables, it is of degree 1 in each variable, and none of its monomials are of the form $q_{i,\ell} q_{i,\ell'}$.

Theorem 3 *An ordinal potential game is a Lyapunov game with respect to dynamics* (3)*. Furthermore, its has some multiaffine Lyapunov function.*

If ϕ is the potential of the ordinal potential game, then one can take its expectation $F(Q) = \mathbb{E}[\ \phi(Q) \mid$ players play pure strategies according to $Q\ ]$ as a Lyapunov function with respect to dynamics (3). The following class of games have been introduced [12, 14].

Definition 2 (**Potential Game** [14]). A game is called a *continuous potential game* if there exists a $\mathscr{C}^1$ function $F : K \to \mathbb{R}$ such that for all i, ℓ and Q,

$$\frac{\partial F}{\partial q_{i,\ell}}(Q) = c_i(e_\ell, Q). \tag{5}$$

Proposition 1 *A continuous potential game is a Lyapunov game with respect to dynamics* (3)*. Its has some multiaffine Lyapunov function.*

Proposition 2 *An (exact) potential game of potential ϕ leads to a continuous potential game with $F(Q) = \mathbb{E}[\ \phi(Q)\ ]$, and conversely, the restriction of F of class $\mathscr{C}^2$ to pure strategies of a potential in the sense of above definition leads to an (exact) potential.*

A Lyapunov game can have some non-multiaffine potential function, hence not all Lyapunov games with respect to dynamics (3) are ordinal games. The interest of Lyapunov functions is that they provide convergence. Observing that all previous classes are Lyapunov games with respect to dynamics (3), this gives the full interest of this corollary.

Corollary 1 *In a Lyapunov game with respect to general dynamics* (3)*, whatever the initial condition is, the solutions of mean-field approximation* (2) *will converge. The stable limit points are Nash equilibria.*

6 Replicator Dynamics for Multiaffine Lyapunov Games

Fortunately, this is possible to go further, observing that many of the previous classes turn out to have a multiaffine Lyapunov function. The key observation is the following (the proof mainly relies on the fact that second order terms are null for multiaffine functions).

Lemma 1 *When F is a multiaffine Lyapunov function,*

$$\mathbb{E}[\ \Delta F(Q(t+1))\ |Q(t)\ ] = \sum_{i=1}^{n}\sum_{\ell=1}^{m_i} \frac{\partial F}{\partial q_{i,\ell}}(Q(t))\mathbb{E}[\ \Delta q_{i,\ell}\ |Q(t)\ ], \tag{6}$$

where $\Delta F(t) = F(Q(t+1)) - F(Q(t))$.

Notice that for Lyapunov game with a multiaffine Lyapunov function F, with respect to Dynamic (3) (this include ordinal, and hence potential games from above discussion), the points Q^* realizing the minimum value F^* of F over compact K must correspond to Nash equilibria.

Fortunately, this is possible to get bounds on the expected time of convergence (see in [2]): we write $L(\mu)$ for the subset of states Q on which $F(Q) \leq \mu$.

Definition 3 (ε-**Nash equilibrium**). Let $\varepsilon \geq 0$. A state Q is some ε-Nash equilibrium iff for all $1 \leq i \leq n$, $1 \leq \ell \leq m_i$, we have $c_i(e_\ell, Q_{-i}) \geq (1-\varepsilon)c_i(q_i, Q_{-i})$.

Theorem 4 *Consider a Lyapunov game with a multiaffine Lyapunov function F, with respect to* (3)*. This includes ordinal, and hence potential games from above discussion. Taking $b = \mathcal{O}(\varepsilon)$, whatever the initial state of the stochastic algorithm is, it will almost surely reach some ε-Nash equilibrium. Furthermore, it will do it in a random time whose expectation $T(\varepsilon)$ satisfies $T(\varepsilon) \leq \mathcal{O}(\frac{F(Q(0))}{\varepsilon})$.*

References

1. Berenbrink, P., Friedetzky, T., Goldberg, L.A., Goldberg, P., Hu, Z., Martin, R.: Distributed selfish load balancing. In: Proceedings of the SODA, pp. 354–363. ACM, New York (2006)
2. Bournez, O., Cohen, J.: Learning equilibria in games by stochastic distributed algorithms. Université de Versailles, CoRR, abs/0907.1916, (2009)
3. Chien, S., Sinclair, A.: Convergence to approximate nash equilibria in congestion games. In: Proceedings of the SODA, pp. 169–178. (2007)
4. Coucheney, P., Touati, C., Gaujal, B.: Fair and efficient user-network association algorithm for multi-technology wireless networks, In: Proceedings of the INFOCOM (2009)
5. Even-Dar, E., Kesselman, A., Mansour, Y.: Convergence time to Nash equilibrium in load balancing. ACM Trans. Algorithms **3**(3), 84–92 (2007)
6. Fischer, S., Räcke, H., Vöcking, B.: Fast convergence to Wardrop equilibria by adaptive sampling methods. In: Proceedings of the STOC, pp. 653–662. (2006)
7. Goldberg, P.W.: Bounds for the convergence rate of randomized local search in a multiplayer load-balancing game. In: Proceedings of the PODC '04, pp. 131–140. (2004)
8. Johnson, D.S., Papadimitriou, C.H., Yannakakis, M.: How easy is local search? J. Comput. Syst. Sci. **37**(1), 79–100 (1988)
9. Koutsoupias, E., Papadimitriou, C.: Worst-case equilibria. In: Symposium on, Theoretical Computer Science (STACS'99), pp. 404–413. (1999)
10. Libman, L., Orda, A.: Atomic resource sharing in noncooperative networks. Telecommun. Syst. **17**(4), 385–409 (2001)
11. Monderer, D., Shapley, L.S.: Potential games. Games Econ. behav. **14**(1), 124–143 (1996)
12. Thathachar, M.A.L., Sastry, P.S., Phansalkar, V.V.: Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information. IEEE trans. Syst. man Cybern. **24**(5), 769–777 (1994)
13. Rosenthal, R.W.: A class of games possessing pure-strategy Nash equilibria. Int. J. Game Theory **2**(1), 65–67 (1973)
14. Sandholm, W.H.: Potential games with continuous player sets. J. Econ. Theory **97**(1), 81–108 (2001)
15. Weibull, J.W.: Evolutionary Game Theory. The MIT Press,Cambridge (1995)

Autonomic Management of Cloud-Based Systems: The Service Provider Perspective

Emiliano Casalicchio and Luca Silvestri

Abstract The complexity of Cloud systems poses new infrastructure and application management challenges. One of the common goals of the research community, practitioners and vendors is to design self-adaptable solutions capable to react to unpredictable workload fluctuations and changing utility principles. This paper analyzes the problem from the perspective of an application service provider that uses a cloud infrastructure to achieve scalable provisioning of its services in the respect of QoS constraints. We designed and implemented two autonomic cloud resource management architectures running five different resource provisioning algorithms. The implemented testbed has been evaluated under a realistic workload based on Wikipedia access traces.

1 Introduction

The extreme complexity of cloud systems calls for novel adaptive management solutions for scalable, maintainable, cost-effective cloud provision, at all software stack layers. Cloud infrastructures and services management can be operated from different perspectives possibly with conflicting goals. Cloud providers are interested in maximizing resources utilization minimizing management costs. Customers of cloud services, like Application Service Providers (ASPs), want to fulfill service level agreements (SLAs) stipulated with users minimizing the costs for buying virtualized resources.

One of the promises of cloud computing is to facilitate ASPs in starting up and providing their services avoiding the costs to build, manage and maintain a data

E. Casalicchio (✉) · L. Silvestri
University of Tor Vergata, Rome, Italy
e-mail: casalicchio@ing.uniroma2.it

L. Silvestri
e-mail: silvestri@ing.uniroma2.it

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_5,

center infrastructure. Therefore, to realize this promise it is fundamental to investigate models, architectures and policies to support QoS provisioning at service level. The service provisioning problem (or application-level resource provisioning) problem is related to the management of cloud resources and services from the end-user (e.g. ASP) viewpoint. In this paper we address this challenge considering how an autonomic service provisioning architecture can be realized and how the resource and service management can be planned (and actuated).

The concept that cloud customers should be empowered with their own dynamic controller, outside of the cloud platform or as an extension/service of the cloud platform itself, and that such controller should allow the definition of policies to automatically dimension the number of instances and containers, thus reducing application costs and guaranteeing SLAs, has been introduced in [8] and [9]. Moreover, [10] proposes a cloud auto-scaling system for scientific applications.

In line with this trend is the idea, proposed in [2], that also cloud systems should be organized according to the autonomic loop MAPE-K [7] (Monitor, Analyze, Plan, Execute, and Knowledge). In such a way, cloud-based applications should be able to automatically react to changing components, workload, and environmental conditions while minimizing operating costs and preventing SLA violations. Analyzing solutions proposed in literature we identified mainly works focused on the analysis and planning phases (e.g. [1, 6, 9, 10]) and others focused on the monitoring phase (e.g. [4]).

In a previous work [3] we performed a first survey on IaaS providers features respect to their use in the implementation of solutions for autonomic management of computational resources in cloud infrastructures. The main results of our analysis were that no IaaS provider delivers all the services necessary to implement an autonomic service management solution. Only Amazon Web Service (AWS) delivers, even with some important limitations, an integrated autonomic service management solution. Therefore we proposed four architectures (Extreme ASP control, Full ASP control, Partial ASP Control, and Limited ASP control) that allow to implement autonomic management of cloud resources and that provide different degrees of customization and control over the autonomic loop phases.

This paper extend our previous results as follows. First, we provided an implementation of the Partial ASP Control, and Limited ASP control architectures using features and services of the Amazon Elastic Compute Cloud (EC2)[1] Infrastructure. Then, we proposed and implemented five reactive resource allocation policies. Finally, we set up a real testbed and we evaluated, under a realistic workload, the proposed architectures and policies.

The following sections will present: the autonomic architecture implementation and the VM allocation policies considered (Sect. 2); the workload, the testbed implementation and the experimental results (Sect. 3). Finally, Sect. 4 gives concludes the paper.

[1] This research is partially supported by a grant from the AWS in Education program, 2010–2011 (http://www.ce.uniroma2.it/cloud/.

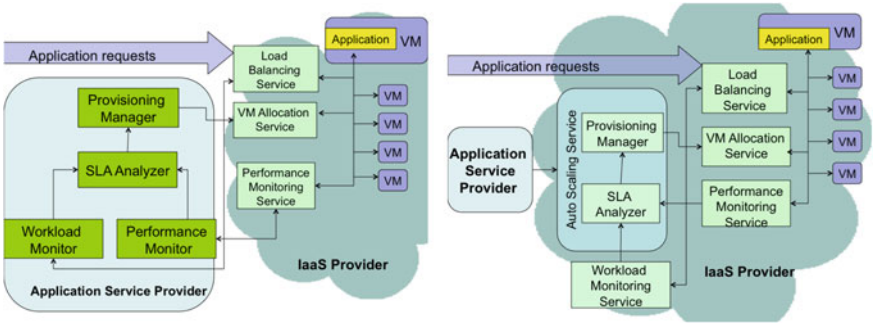


Fig. 1 The Partial ASP control architecture (on the *left*) and the Limited ASP control architecture (on the *right*)

2 Implementation

In this section we describe how we implemented the Partial and Limited ASP control architectures (see Fig. 1) using Amazon EC2. The Limited ASP control architectures uses the Amazon Auto Scaling service to perform adaptation. In this simple architecture the ASP has only to properly setup the Auto Scaling parameters specifying thresholds and adaptive actions. In the Partial ASP control architecture, as shown in Fig. 1, the ASP has the total control on the Analysis and Planning phases of the autonomic cycle. Single VMs are EC2 on-demand instances placed behind an Elastic Load Balancer in one or more availability zones inside the same EC2 Region. The target web application is completely replicated on every VM and data are centralized. This simplifying assumption allows us to consider the application as a whole without taking account of interactions between its layers (otherwise resource replication and distribution have to be considered for every layer) and data synchronization issues.

Incoming requests are managed by the Elastic Load Balancer that distributes them among active VMs using the Least Loaded policy. The Performance Monitoring Service is realized through Amazon CloudWatch, a web service that allows to collect, analyze, and view system and application metrics. The various components of the autonomic architecture we implemented (Performance and Workload Monitor, SLA Analyzer and Provisioning Manager) have been realized as Java modules running under JRE 1.6 and using AWS SDK for Java APIs to interact with EC2 services.

2.1 VMs Allocation Policies

In the Partial ASP control implementation we use a simple reactive allocation policy, named *Reactive 1 step early* (r-1), that works as follows. r-1 observes the requests arrivals and computes the average arrival rate over time slots of 1–5 min. Measured

the average arrival rate λ_{i-1} for the time slot $i-1$, the arrival rate for the time slot i is estimated as $\hat{\lambda}_i = (1+a)\lambda_{i-1}$, where $a > 0$.

Computed the estimated arrival rate and supposing that the service rate μ and the maximum response time allowed R_{max} are known, the minimum number of VMs $x_{i,\min}$ needed to guarantee R_{max} is determined by $x_{i,\min} = \frac{\hat{\lambda}_i R_{max}}{\mu R_{max} - 1}$.

The deallocation policy is the following: A VMs is deallocated only if no more needed, that is if at the beginning of time slot i the number of allocated VMs is greater then $x_{i,\min}$, and if the billing period (typically 1 h) is expired (or is going to expire in few minutes). The assumption behind this deallocation policy is that does not make sense to deallocate a resources for which we already paid.

The Limited ASP control solution is directly implemented using the Amazon Auto Scaling service. Amazon Auto Scaling offers the possibility to define allocation/deallocation strategies based on CloudWatch metrics values. In particular, it allows to define alarms on every CloudWatch metric and to decide what action to take when the threshold specified by the alarm is violated. In the specific we set up the following four policies: *Utilization-based, One alarm* (UT-1Al), *Utilization-based, Two alarms* (UT-2Al), *Latency-based, One alarm* (LAT-1Al), *Latency-based, Two alarms* (LAT-2Al).

In all the policies we defined, the possibility to take an adaptation action is evaluated periodically (every 1 or 5 min). For the r-1 policy timing is directly managed by the SLA analyzer we implemented. For the threshold based policies we used a cooldown interval (i.e. the minimum time interval between two adaptation actions) equals to the evaluation period.

3 Experimental Evaluation

The evaluation of the proposed architectures has been done instrumenting the system with the allocation policies described in Sect. 2.1 and comparing the system capability to satisfy a given Service Level Objective (SLO), the system responsiveness (i.e. the capability to promptly react to unexpected load variations), and the allocation costs of the implemented policies. We do not define any specific metric for the responsiveness but estimated it through the analysis of the response time and allocated VMs time series.

The metrics used for comparison are the following: *Latency* (or Response time) collected by the CloudWatch monitor at the load balancer; *Number of Allocated VMs*, evaluated by CloudWatch; *VMs Allocation Cost*, obtained analyzing the log of VMs allocation and deallocation actions. Both system implementations have been evaluated under two different load conditions (see Sect. 3.1) but, due to the lack of space, only results obtained under the bursty workload are reported. The fairness of our comparison is given by the fact that both the architectures are implemented using the Amazon EC2 infrastructure.

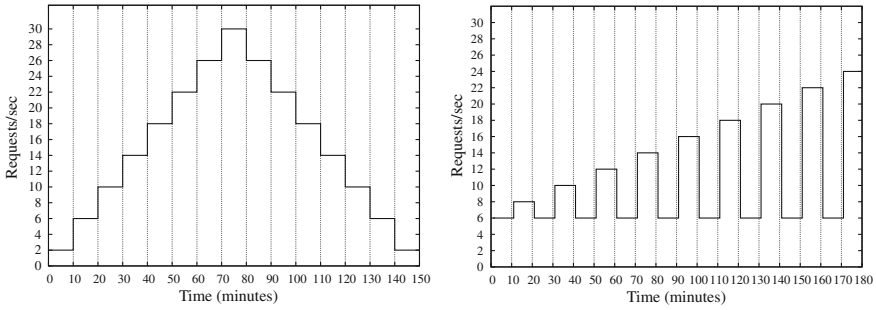


Fig. 2 An example of stress-test workload generated using MediaWiki and Httperf. On the *left* the smooth ramp workload and on the *right* the bursty workload

From the experimental results emerged that (see Sect. 3.2): having full control over the resource management plan has its advantages in term of achievable performances and reduced resource utilization costs; the proposed allocation policy, r-1, outperforms the threshold based policies implemented using the Auto Scaling service; Latency and responsiveness benefit of shorter performance and workload evaluation periods if the allocation/deallocation decision is taken periodically rather than triggered by events.

3.1 Workload Generation and Testbed Setup

Right now, benchmarking cloud services is an open issue and no standard solutions are available. Therefore, we decided to create our own workload generator assembling: the WikiBench [12] benchmark, scaled Wikipedia traces to generate the requests [11], and httperf [5] to control the request generation rate and statistics collection. In this way we have been able to generate not only traffic with a smooth and periodical variation of intensity (*smooth ramp workload*) but also controlled bursts of requests ranging from the 33 % to the 300 % of the base traffic (*bursty workload*), as shown in Fig. 2. We implemented our testbed by means of the Amazon EC2 infrastructure and services. We used the following resources: from 1 to 10 Amazon EC2 m1.small instances (each VM replicates and executes the front-end part of the MediaWiki web application); one Amazon EC2 m1.large instance implementing the database server that runs the MediaWiki back-end (this solution avoid multi-tier load balancing and data consistency problems); the Amazon Elastic Load Balancer to distribute incoming traffic among the active VMs; an EC2 m1.small instance, located in the same availability zone of the application and back-end servers, to run the workload generator; an EC2 m1.small instance to run the components of the Partial ASP control architecture we implemented. The VMs, the database, the load balancer and the workload generators run all in the same availability zone (us-east-1a). Placing the workload generators and the servers in the same availability zone we tried to

isolate the components of the total response time due to queuing and computation times reducing to the minimum the effects of the network latency.

To evaluate the capacity of a single VM and the workload intensity that a VM is capable to handle we proceeded as follows. First, empirically, we evaluated that an EC2 m1.small serves four requests per second with an utilization of about 62 %. Second, considering a simple M/M/1 model and that the system is stable ($\rho = \lambda/\mu < 1$) we evaluated the average service rate of a single VM as 6.45 requests per second. Finally, supposing the ASP has to guarantee a maximum response time (the Service Level Objective) of 0.5 s, we were capable to determine the minimum number of VMs needed to handle a given load and to satisfy the SLO. The minimum number of VMs is evaluated using an M/G/1/PS queuing model.

Moreover, empirically we evaluated that the EC2 m1.large instance used to host the MediaWiki database is capable to guarantee that, for the level of traffic we submit to it (up to 40 requests per second), the back-end server never represents the system bottleneck.

3.2 Performance and Responsiveness Analysis

In a set of experiments we compared all the allocation policies under the bursty workload (see Fig. 2). We considered two cases, namely *1*-min and *5*-min. In the former case the allocation/deallocation decision is taken every minute, in the latter case every 5 min. Results for the first case are shown in Figs. 3 and 4. The first observation is about the responsiveness that range from 2 to 10 min when the burst exceeds the 130 % of base workload (i.e. starting from time 70 min). The empirical CDF plot shows that r-1 is capable to satisfy the SLO in more then the 95 % of observation. The UT-1Al policy follows with a SLO satisfaction in more than the 87 % of the cases. All the other policies offer worst performances (from 79 to 82 %). The allocation cost is reported in Table 1. The r-1 allocation strategy allows to save about the 32 % versus the UT-2Al, LAT-2Al and LAT-1Al policies, and about the 17 % compared to the UT-1Al policy. When allocation/deallocation decisions are taken every 5 min the system is more stable as shown by the lower allocation costs (see Table 1). However, for the threshold based policies, the lower allocation cost impacts negatively on the latency, the responsiveness and, therefore, the capability to satisfy the SLO (see Fig. 5). Only the r-1 ($a = 0.1$) policy performs better than in the 1-min case, with a probability to satisfy the SLO equal to 1. The r-1 ($a = 0.05$) had almost the same behavior. All the other policies are characterized by the probability of the system to satisfy the SLO ranging from 0.7 to 0.83 (see Fig. 5), while in the 1-min case the same probability was greater than 0.79. Moreover, observing Latency CDF (Fig. 5), it is immediate to note that the distribution of the response time has a longer queue than in the 1-min case (Fig. 4). This means that the system did not react quickly to workload fluctuations and, when a burst of request arrived, ASP users experimented an higher response time for a longer time period.

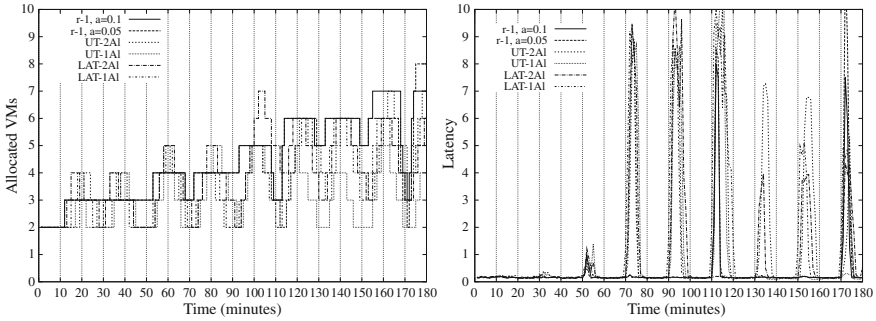


Fig. 3 Trend of the allocated VMs (*left*) and of the Latency (*right*) in the 1-min case

Fig. 4 The empirical CDF of the Latency for case 1-min. Here we show a zoom for Latency values less the 1 s. All the CDF policies converge to 1 on the whole experiment

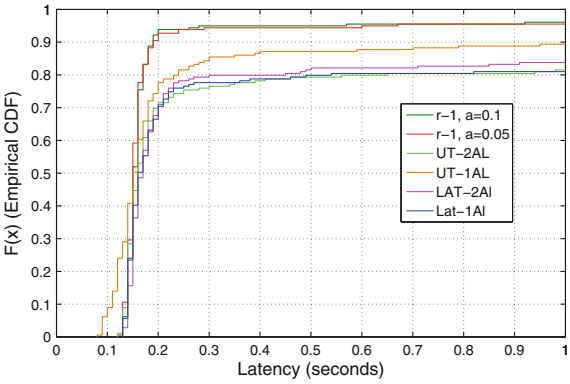


Table 1 The allocation cost for the two cases 1-min (left) and 5-min (right)

Allocation policies	Total cost	Cost/hour	Allocation policies	Total cost	Cost/hour
r-1, a = 0.1	1.615	0.538	r-1, a = 0.1	1.53	0.51
r-1, a = 0.05	1.615	0.538	r-1, a = 0.05	1.445	0.481
UT-2Al	2.38	0.793	UT-2Al	1.445	0.481
UT-1Al	1.955	0.651	UT-1Al	1.955	0.51
LAT-2Al	2.465	0.821	LAT-2Al	1.275	0.425
LAT-1Al	2.295	0.765	LAT-1Al	1.19	0.39

Finally, we remark that the allocation cost decrease for all the threshold based allocation policies and in the specific the Latency based allocation policies allow to save about the 20 % of the allocation cost (compared to the other strategies).

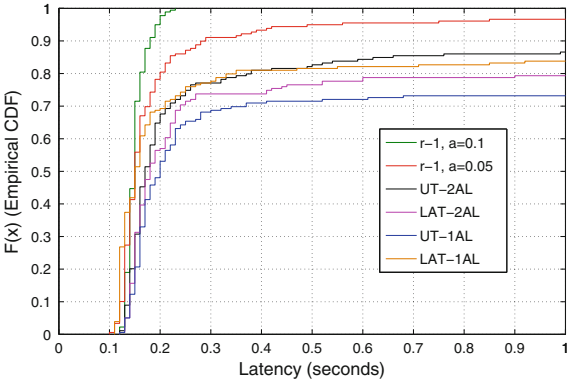


Fig. 5 The empirical CDF of the Latency for case 5-min. Here we show a zoom for Latency values less the 1 s. All the CDF policies converge to 1 on the whole experiment

4 Conclusions

This paper brought to the community several contributions. We provided an implementation of two autonomic service provisioning architectures using Amazon EC2, we set up a real testbed and evaluated, under a realistic workload, the proposed architectures and five reactive allocation policies.

The main result from the conducted experiments is that a user-defined policy allows to have better performance and responsiveness than IaaS providers Autoscaling services, sometimes at a lower price.

References

1. Andrzejak, A., Kondo, D., Yi, S.: Decision model for cloud computing under sla constraints. In: 2012 IEEE 20th international symposium on modeling, analysis and simulation of computer and telecommunication systems, pp. 257–266, 2010 18th annual IEEE/ACM international symposium on modeling, analysis and simulation of computer and telecommunication systems (2010). http://www.computer.org/csdl/proceedings/mascots/2010/4197/00/index.html
2. Brandic, I.: Towards self-manageable cloud services. Comput. Softw. Appl. Conf. Annu. Int. **2**, 128–133 (2009)
3. Casalicchio, E., Silvestri, L.: Architectures for autonomic service management in cloud-based systems. Proceedings of IEEE SCS-MoCS, In (2011)
4. Gong, Z., Gu, X., John, W.: Press: predictive elastic resource scaling for cloud systems. In: Proceedings of IEEE NSM pp. 9–16 (2010).
5. Httperf: The httperf http load generator. http://code.google.com/p/httperf/
6. Hu, Y., Wong, J., Iszlai, G., Litoiu, M.: Resource provisioning for cloud computing. In: Proceedings of ACM CASCON, pp. 101–111 (2009).
7. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. IEE Comput. **36**(1), 41–50 (2003)
8. Lim, H.C., Babu, S., Chase, J.S., Parekh, S.S.: Automated control in cloud computing: challenges and opportunities. In: Proceedings of ACM ACDC, pp. 13–18 (2009).
9. Litoiu, M., Woodside, M., Wong, J., Ng, J., Iszlai, G.: A business driven cloud optimization architecture. In: Proceedings of 2010 ACM SAC, pp. 380–385 (2010).

10. Mao, M., Li, J., Humphrey, M.: Cloud auto-scaling with deadline and budget constraints. In: Proceedings of the 11th ACM/IEEE Grid (2010).
11. Urdaneta, G., Pierre, G., van Steen, M.: Wikipedia workload analysis for decentralized hosting. Comput. Netw. **53**(11), 1830–1845 (2009)
12. Wikibench, the realistic web hosting benchmark. http://www.wikibench.eu/

Part II
Green IT, Energy and Networks

Measuring Energy Efficiency Practices in Mature Data Center: A Maturity Model Approach

Edward Curry, Gerard Conway, Brian Donnellan, Charles Sheridan and Keith Ellis

Abstract Power usage within a Data Center (DC) goes beyond the direct power needs of servers to include networking, cooling, lighting and facilities management. Data centers range from closet-sized operations, drawing a few kilowatts (kW), to mega-sized facilities, consuming tens of megawatts (MWs). In almost all cases, independent of size there exists significant potential to improve both the economic and environmental bottom line of data centers by improve their energy efficiency, however a number of challenges exist. This paper describes the resulting maturity model, which offers a comprehensive value-based method for organizing, evaluating, planning, and improving the energy efficiency of mature data centers. The development process for the maturity model is discussed, detailing the role of design science in its definition.

1 Introduction

It is estimated that Information and Communication Technology (ICT) is responsible for at least 2 % of global greenhouse gas (GHG) emissions with Data Centers (DC)'s accounting for 14 % of the ICT footprint [1] According to McKinsey & Co. [2] the world's 44 million servers consume 0.5 % of all electricity and produce 0.2 % of carbon dioxide emissions, or 80 Mt a year. The emissions generated for all the DC's

E. Curry
Digital Enterprise Research Institute National University of Ireland, Galway, Ireland

G. Conway (✉) · B. Donnellan
Innovation Value Institute National University of Ireland, Maynooth, Ireland
e-mail: gerard.conway@nuim.ie

C. Sheridan · K. Ellis
Intel Labs Europe, Intel Corporation, Princeton, USA

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_6,

globally are approaching the emissions levels of entire countries like Argentina or the Netherlands that are ranked 142nd and 146th respectively. Given a business as usual scenario greenhouse gas emissions from DC are projected to more than double from 2007 levels by 2020 [1].

The efficient operation of a data center requires a diverse range of knowledge and skills from a large ecosystem of stakeholders. A Data Center (DC) requires expertise from engineering (including electrical, civil, mechanical, software and electronic) to accountancy to systems management. Working in a DC may require knowledge of, the laws of thermodynamics and OpEx charge-back accounting principles in the same day. To address this issue, a consortium of leading organizations from industry, the nonprofit sector, and academia has developed and tested a framework for systematically improving Energy Efficiency (EE) capabilities within mature DCs.

The Innovation Value Institute (IVI; http://ivi.nuim.ie) consortium used an open-innovation model of collaboration; engaging academia and industry in scholarly work to create a data center energy efficiency assessment. It offers a comprehensive value-based model for organizing, evaluating, planning, and managing DC capabilities for energy efficiency, and it fits within the IVI's IT-Capability Maturity Framework (IT-CMF). The objective of the assessment is to provide a high-level assessment of maturity for IT managers with responsibility for DC operations. This paper describes the Data Center Energy Efficiency maturity model.

2 Data Center Energy Consumption

With power densities of more than 100 times that of a typical office building, energy use is a central issue for DCs. The US Environmental Protection Agency estimates that servers and DCs are responsible for up to 1.5 % of the total US electricity consumption [3] or roughly 0.5 % of US GHG emissions for 2007. Massive growth in the volumes of computing equipment, and the associated growth in areas such as cooling equipment has lead to increased energy usage and power densities within data centers. If growth continues in line with demand, the world will be using 122 million servers in 2020. Trends toward cloud computing have the potential to further increase the demand for DC-based computing services.

Power usage within a DC goes beyond the direct power needs of servers to include networking, cooling lighting and facilities management with power draw for data centers ranging from a few kWs for a rack of servers to several tens of MWs for large facilities. While the exact breakdown of power usage will vary between individual DCs, illustrates the examination of one DC where up to 88.8 % of the power consumed by the DC was not used on computation; for every 100 W supplied to the data center only 11.2 W were used for computation [3].

Air conditioners, power converters and power transmission can use almost half of the electricity in the datacenter, the IDC estimates that DC energy costs will be higher than equipment costs by 2015 [4]. The cost of operating a data center goes beyond just the economic bottom line; there is also an environmental cost. By 2020, the

net footprint for data centers is predicted to be 259 MtCO2e [1]. There is significant potential to improve both the economic and environmental bottom line of data centers by improve their energy efficiency, however a number of challenges exist.

3 Energy Efficiency within Mature Data Centers

Data centers are complex eco-systems that interconnect elements of the ICT, electrical and mechanical fields of engineering. Energy efficient data center operations require a holistic approach to both IT and facilities energy management. Organizations face many challenges in developing and driving their overall DC EE strategies and programs: the complexity of the subject and its rapid evolution, the lack of DC EE subject-matter expertise, the lack of relevant instrumentation and information within mature DCs, the need for new metrics and measures and the need for multiple stakeholder engagement and agreement (IT, facilities, business users). With electricity costs the dominant operating expense of a DC it is vital to maximize the efficiency to reduce both the environmental and economic cost. DC and IT leaders frequently can't find satisfactory answers to questions such as: What is the utilization of the DC? Is the infrastructure provisioned appropriate for the business requirements? How energy efficient is the DC? Where are the inefficiencies? Are there clear measurable goals and objectives for DC EE? What more could be done to contribute to DC EE goals? What is the roadmap for DC EE improvements?

4 The Need for a Data Center Energy Efficiency Maturity Model

Maturity models are tools that have been used to improve many capabilities within organizations, from Business Process Management (BPM) [5] to Software Engineering (CMMI) [6]. Typically, these models consist of a set of levels that describe how well the behaviors, practices, and processes of an organization can reliably produce required outcomes. They can have multiple uses within an organization, from defining a place to start, to providing a foundation to build a common language and shared vision, to defining priorities and roadmaps. If defined by a community the model can capture the collective knowledge of the community's prior experiences. A maturity model could also be used as an assessment tool and benchmark for the comparative assessments of the capabilities of different organizations.

Maturity models have also been developed to support the management of IT organizations. The Innovation Value Institute (IVI; http://ivi.nuim.ie), a consortium of leading organizations from industry (including, Microsoft, Intel, SAP, Chevron, Cisco, Boston Consultancy Group, Ernst & Young, and Fujitsu) the nonprofits sector, and academia, have developed the IT-Capability Maturity Framework (IT-CMF)

[7]. The IT-CMF provides a high-level process capability maturity framework for managing the IT function within an organization to deliver greater value from IT by assessing and improving a broad range of management practices. The framework identifies critical IT capabilities and defines maturity models for each process. A core function of the IT-CMF is to act as an assessment tool and a management system.

There is a need to improve the behaviors, practices and processes within data centers to deliver greater energy efficiency. To address the issue, the IVI consortium has extended the IT-CMF with a maturity model for systematically assessing and improving DC capabilities for energy efficiency.

5 Design Methodology

The IVI consortium engages in an open-innovation model of collaboration, engaging academia and industry in scholarly work to create the DE EE Maturity Model. As maturity addresses a number of different interrelated concerns including strategy, information management, facilities managements, the approach taken adheres to design science guidelines. The development of the model was undertaken using a design process [8] with defined review stages and development activities based on the design science research guidelines advocated by Hevner et al. [9].

During the design process, researchers participate together with practitioners within research teams to research and develop the model. The research team interviewed multiple DC stakeholders to capture the views of key domain experts and to understand current practice and barriers to improving DC EE. The team widely consulted the relevant literature, both industrial and academic, on DC EE. Industrial best practices including the EU code of conduct for DE EE and the work of the Green Grid on metrics were incorporated. Once the model was developed it was piloted within a number of DCs with learning and feedback incorporated into subsequent versions. The core of the resulting maturity model for DC EE provides a management system with associated improvement roadmaps that guide senior IT and business management in selecting strategies to continuously improve, develop, and manage the DC capability for EE.

6 Maturity Model for Energy Efficiency in Mature Data Center

The Data Center Energy Efficiency model offers a comprehensive value-based model for organizing, evaluating, planning, and managing DC EE capabilities. The model fits within the IT-Capability Maturity Framework (IT-CMF) [7] and is aligned with the broader Sustainable ICT critical process maturity model [10].

Table 1 Capability building blocks of energy efficient data centers

Category	Capability	Description
Mgmt.	Organizational structure	How the data center and its energy efficiency is managed, who is responsible for running the DC and how integrated are: IT, facilities, and the business
	Policy	The policies in place for energy efficiency within the DC and how they are aligned across the enterprise
	Manageability and metering	The metering use by IT and facilities to improve understanding and manageability of energy usage
Operations	IT infrastructure and services	The management of IT equipment and services to ensure energy efficiency
Building	Internal air and cooling	The internal air management techniques employed
	Cooling plant	The design and management of the cooling system
	Power infrastructure	The management of power generation, conditioning and delivery systems to maximize energy efficiency

The DC EE assessment methodology determines how different DC capabilities are contributing to energy efficiency goals and objectives. This gap analysis between what energy efficiency targets are, and what they are actually achieving, positions the model as a management tool for aligning relevant capabilities with EE objectives [11]. The model focuses on the execution of four key actions to improve the management EE in the DC: define the goals and objectives for the DC EE program, understand the current DC capability maturity level, systematically develop and manage the DC capability building blocks, assess and manage DC progress over time. In the remainder of this section we outline the main components of the model in more detail and discuss the assessment approach.

6.1 Capability Building Blocks

The maturity model consists of seven capability building blocks (see Table 1) across the following three categories: Management, including the organizational structure, policy, and metering of the DC; Operations, which includes the efficient management of existing and new IT equipment and services; Building, which covers how air management, cooling and power infrastructure are managed to increase energy efficiency.

Table 2 Level 1: Initial Data Center Energy Efficiency

Capability	Maturity
Organizational structure	No formal organizational structure expected at this level
Policy	No formal EE policies in place
Manageability and metering	IT has no specific energy related metrics or metering capability in place. Facilities have limited building level metering. Shared metering systems are used that require manual readings for different support infrastructure. No visualization medium is used to improve manageability of energy data
IT infrastructure and services	Ad-hoc
Internal air and cooling	Ad-hoc design and operation. Typically peripheral/perimeter air-cooling is utilized within the DC. Air supply temperatures are significantly cooler than needed ($\sim$13$-$15 °C) to compensate for a high level of air mixing. No air management techniques are in place. In chilled water DCs Leaving Water Temperature (LWT) at the cooling plant is lower than needed ($\sim$4$-$5 °C).
Cooling plant	Cooling is supplied based on resilience. Energy conservation is not a major consideration. DC predominantly relies on refrigeration cycle and Co-efficient Of Performance (COP) is typically $<$3. Cooling infrastructure is oversized relative to current IT load. Configuration is static (i.e. pumps at fixed speed)

6.2 Assessment Approach

The assessment begins with an online survey of DC stakeholders to understand their individual assessments of the maturity and importance of these capabilities. The survey takes no more than 30 min to complete. Typically, a range of individuals who are involved in, or accountable for, EE for the DC complete the survey. A series of targeted interviews with key stakeholders augments the survey to understand key business priorities and energy efficiency drivers, successes achieved, and initiatives taken or planned. Interviews last between 60 and 90 min; they are used to support the survey data. In addition to helping organizations understand their current maturity level, the initial assessment provides insight into the value placed on each capability, which will undoubtedly vary according to each organization's strategy and objectives. The assessment also provides valuable insight into the similarities and differences in how key stakeholders view both the importance and maturity of individual capabilities, as well as the overall vision for success. Understanding the current levels of maturity and strategic importance lets an organization quickly identify gaps in capabilities. This is the foundation for developing a meaningful action plan [12].

Table 3 Level 2: Basic Data Center Energy Efficiency

Capability	Maturity
Organizational structure	EE is considered by IT and facilities but with a siloed or disjointed approach to energy management
Policy	IT policies have limited consideration for decommissioning, consolidation, refresh, efficient storage allocation, and virtualization. Facility policy position EE as an operational decision criterion. Both IT and Facilities equipment purchase and upgrade decisions have energy related criterion. Policy creation is siloed
Manageability and metering	IT understands IT electrical load and can meter at the UPS level. Some larger facilities infrastructure has individual metering installed such as the chiller units. Static power capping and other fixed strategies are in place. A useable granularity of data is in place i.e. ever hour or better. Basic information systems exist for energy data analysis and decision support. Metrics such as kWh consumed and/or PUE / DCiE are defined and are used to dive effective decisions. Basic measurements in place for air temp, Humidity (HUM), and pressure
IT infrastructure and services	Defined IT landing procedures in place that considers EE. Auditing and decommissioning of unused or un-valued equipment. Appropriate storage allocation and some level of server consolidation in place. Existing server energy saving features are utilized. Processor P state energy saving modes are enabled.
Internal air and cooling	IT equipment inlet supply temperatures lower than the low end of the ASHRAE recommendations due to air mixing (at time of writing $\sim$16$-$18 °C). In chilled water DCs LWT is slightly higher than L1 due to supply air temp increase. IT equipment is oriented in a cold aisle hot aisle configuration. Blanking panels within racks are utilized and gaps in floor are sealed. HUM utilizes a fixed set point. Auditing for improvement opportunities are conducted such as inlet temp
Cooling plant	Refrigeration infrastructure is appropriately sized or strategies are in place to align cooling capacity and demand. Typically COP is $\sim$3$-$4. But refrigeration is still predominately required. Limited economization utilized
Power infrastructure	UPS is more effectively sized or strategies employed to appropriately align demand and capacity relative to existing IT load. Efficiency is typically $\sim$90 % at 50 % load or above

6.3 Assessing and Managing Progress

With the initial assessment complete, organizations will have a clear view of current capability and key areas for action and improvement. However, to further develop the capability, the organization should assess and manage progress over time by using the

Table 4 Level 3: Intermediate Data Center Energy Efficiency

Capability	Maturity
Organizational structure	EE is inherent in policies and the management of the DC and takes account of the interrelationship of IT and Facilities. Both disciplines share knowledge openly and work together to improve EE of the business offering
Policy	IT server landing policies take account of facility cooling requirements. Cable and air management best practice are defined. Policy moves towards increased virtualization. There is an efficient and effective customer requirement program in place that matches service to needs. Facilities have defined improvement roadmap that targets sustainable operations. There is a resilience review policy/audit. External best practices are systematically reviewed and internalized. Best practice policies exist for new DC design and build
Manageability and metering	IT have granular understanding of IT electrical load through metering at the power distribution unit/board level. Facilities have an increased level of support infrastructure metered. Granularity of metering is 15 min or better. DC information systems combine the IT and facilities energy data with dashboards supporting decision-making. Environmental data both internal and externally is integrated into dashboard (i.e. weather data affects on air-cooled chillers). Basic remote intelligent control of IT and facilities infrastructure is available
IT infrastructure and services	Comprehensive consolidation program in place. DC has begun to move some legacy services to virtualized environments. IT services are moving towards delivering valued performance per watt
Internal air and cooling	Air inlet supply temperature at lower end of the ASHRAE recommendations (at time of writing ~20−21 °C). In addition to blanking/sealing an optimized diffuser layout has been implemented, ideally based on CFM or similar study. Row based cooling utilized or perimeter. Additional air barriers are used to minimize air mixing. CRAC/CRAH units are powered down or throttled back to match air capacity to heat load to save on motor fan consumption. HUM uses a tight floating set point range. LWT is increased in line with the server supply temp
Cooling plant	COP is typically ~4−6. Both standalone and systemic efficiency are considered. Pumps have Variable Frequency Drives (VFDs). Cooling coils have an economization mode that turns off refrigeration cycle if outside conditions permit. Partial wetside/airside economization is increasingly utilized, ~50 % of the year
Power infrastructure	UPS is correctly sized. The UPS is typically ~93 % efficient or above at 50 % load. There is an optimal number of PDUs

assessment results to develop a roadmap and action plan and add a yearly/half-yearly follow-up assessment to the overall DC energy efficiency management process to measure over time both progress and the value delivered from improving energy efficiency. Agreeing on stakeholder ownership for each priority area is critical to developing both short-term and long-term action plans for improvement. The assessment results can be used to prioritize the opportunities for quick wins—that is, those capabilities that have smaller gaps between current and desired maturity.

Table 5 Level 4: Advanced Data Center Energy Efficiency

Capability	Maturity
Organizational structure	Holistic management approach that understands the symbiotic relationships between IT and Facilities, and leverages heuristics in both domains. Operational decisions balance sustainability, resilience, and business needs
Policy	Policies reflect a harmonized process based approach. Best practices are disseminated within the enterprise. IT has a default virtualization and exception justification policy. EE is a criterion in terms of service offerings and purchases. For existing facilities CapEx funding programs/policy for upgrading existing infrastructure are in place. EE begins at the software level to combat code bloating and wasteful service provisioning. The DC function engages and influences with regard to external best practice. Best practice eco-design policies including climate and location in decisions for new-build or re-engineering of DCs. Built environment best practice are incorporated with relevant third party accreditation
Manageability and metering	IT has rack and server level consumption data together with environmental data like temperature and HUM. Facilities infrastructure is completely metered from an electrical standpoint. Environment data internally and externally is totally integrated along with the full electrical consumption data. Any water consumption is assessed and is considered as part of sustainability goals, metrics such Water Usage Effectiveness (WUE) are considered. A pervasive level of dynamic control is in place that leverages virtualized environments to spin up/down servers/services. Facilities are dynamically matched to IT load with VFD, pump and motor, intelligent control logic utilized
IT infrastructure and services	Virtualization is the default practice in terms of server and storage service provision. IT moves toward machine readable Service Level Agreements (SLAs)
Internal air and cooling	Air inlet supply temperature at middle of ASHRAE recommendations (at time of writing $\sim$22$-$24 °C). Full air segregation in place. Cold aisle/ hot aisle/ chimney cabinets/in-rack cooling utilized. CRAC/CRAH units have VFDs and controlled based on dynamic metering. DC utilizes a widened floating set point range for HUM. LWT is $\sim$10$-$12 °C. Wetside/airside economization is utilized
Cooling plant	Where refrigeration is utilized all fans and pumps have VFDs. COP is typically $\sim$6 or greater and normal operation is economization mode $\sim$75 % of the year. Wetside economization/evaporation direct free air-cooling is utilized
Power infrastructure	UPS has a $\sim$94/95 % or better efficiency at $\sim$50 % load. UPS is modular. The UPS is correctly sized to the data center load. Redundancy is appropriate for the criticality of the load. Rack PDU's are efficient with less than 3 % loss. Non-critical load maybe removed off the UPS. Renewable energy resources like solar are considered for lighting systems or heat blocks for back-up generators, etc

Table 6 Level 5: Optimizing Data Center Energy Efficiency

Capability	Maturity
Organizational structure	A team led by a senior executive has responsibility for EE across the enterprise. Sustainability is an inherent value and capability within the enterprise
Policy	Energy efficiency policy is a continuum from the enterprise-level to the software code level and everything in-between. Policies influence beyond the enterprise boundaries requiring vendors and suppliers to incorporate sustainable practices. DC function actively engages in industry bodies and forums regarding best practice and influences EE throughout the enterprise supply-chain
Manageability and metering	IT can measure electrical load at service-level, matching consumption to useful work done. Facilities infrastructure and IT infrastructure is completed metered and optimized automation is in place. Environmental data is completely integrated and all cause and effect relationships are understood. Management dashboards are highly effective decision tools that incorporate energy, resilience, and business variables. Metering and control capabilities extend beyond the DC with real-time integration with other building infrastructure or smart grid infrastructure in terms of energy management, energy re-use, trading, etc
IT infrastructure and services	DC environment is almost exclusively virtualized. Dynamic service management allows transferable workloads. Power usage is matched to valued workload. The enterprise can move service between its data center estate. Cloud based services and energy service level agreements are in place
Internal air and cooling	Inlet temperature is run at the top end of ASHARE recommendations or higher (at time of writing ~25−27 °C). An optimal floating HUM set point is used. DC normal operational mode is 'free-cooling' economization
Cooling plant	Evaporative cooling (Wetside economization), or direct free air-cooling is utilized in climates that permit. Direct touch cooling utilized. Heat recovery systems utilized where appropriate; redundancy has little effect on EE
Power infrastructure	UPS is modular and the most efficient for the given DC, ~96 % efficient at ~40 % load or less. Flywheel may be used for storage, if appropriate. Appropriate non-critical applications are on mains-only. Renewable energy sources are utilized with onsite self-generation where possible. Natural or LED lighting used. A move to direct current design is considered

6.4 Maturity Levels

To get an understanding of each maturity level, the following section outlines what is required at each level and the characteristic of each capability. It is important to note that a DC can be at different levels of maturity for each capability.

The first maturity level is *Initial* and it is characterized by the absence of any formal EE practices or processes within IT or it management structure (Table 2).

The *Basic* level of maturity is the start of some formal management structure with a base level of understanding of the impact of EE within the DC (Table 3).

The *Intermediate* level of maturity shows that all of the major components are in place for the efficient management of the EE within the data center (Table 4).

The fourth level of maturity is *Advanced* which is characterized by a consistent and coordinated approach that is above the industrial average and where there is continuous improvement (Table 5).

The final and highest level of maturity is O*ptimizing* where policies, procedures and standards for EE are set at the highest possible level (Table 6).

References

1. Webb, M.: SMART 2020: Enabling the Low Carbon Economy in the Information Age. The Climate Group, London (2008)
2. Forrest, W., Kaplan, J.M.: Data centers: how to cut carbon emissions and costs. McKinsey Bus. Technol. **14**(6), (2008)
3. U.S. Environmental Protection Agency ENERGY STAR Program: Report to congress on server and data center energy efficiency public law 109–431. Environ. Prot. **109**, 431 (2007)
4. Martinez, N., Bahloul, K.: Green IT Barometer European Organisations and the Business Imperatives of Deploying a Green and Sustainable IT Strategy. IDC, Leiden (2008)
5. Rosemann, M., De Bruin, T.: Towards a business process management maturity model. In: Proceedings of the 13th European Conference on Information Systems, vol. 53, no. 1, pp. 521–532 (2005)
6. Paulk, M.C., Curtis, B., Chrissis, M.B., Weber, C.V.: The capability maturity model for software. Softw. Eng. Proj. Manag. **10**, 1–26 (1993)
7. Curley, M.: Managing Information Technology for Business Value: Practical Strategies for IT and Business Managers. Intel Press, Hillsboro (2004)
8. Donnellan, B., Helfert, M.: The IT-CMF: a practical application of design science. In: Proceedings of the DESRIST 2010. Lecture Notes in Computer Science, vol. 6105. Springer, Heidelberg (2010)
9. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Q. **28**(1), 75–105 (2004)
10. Donnellan, B., Sheridan, C., Curry, E.: A capability maturity framework for sustainable information and communication technology. IT Prof. **13**(1), 33–40 (2011)
11. Curry, E., Guyon, B., Sheridan, C., Donnellan, B.: Developing an sustainable IT capability: lessons from Intel's journey. MIS Q. Executive **11**(2), 61–74 (2012)
12. Curry, E., Donnellan, B.: Understanding the maturity of sustainable ICT. In: Vom, B.J., Seidel, S., Recker, J. (eds.) Green Business Process Management—Towards the Sustainable Enterprise, pp. 203–216. Springer, New York (2012)

Using Energy Criteria to Admit Flows in a Wired Network

Georgia Sakellari, Christina Morfopoulou, Toktam Mahmoodi and Erol Gelenbe

Abstract Admission control in wired networks has been traditionally used as a way to control traffic congestion and guarantee quality of service. Here, we propose an admission control mechanism which aims to keep the power consumption at the lowest possible level by restricting the more energy-demanding users. This work relies on the fact that power consumption of networking devices, and of the network as a whole, is not proportional to the carried traffic, as would be the ideal case [1]. As a result some operating regions may be more efficient than others and "jumps" may arise in power consumption when new traffic is added in the network. The proposed mechanism aims to keep power consumption in the lowest possible power consumption level, hopping to the next level only when necessary.

G. Sakellari (✉)
School of Architecture, Computer Science Field,
Computing and Engineering, University of East London,
London E16 2RD, UK
e-mail: g.sakellari@uel.ac.uk

C. Morfopoulou · E. Gelenbe
ISN Group, Electrical & Electronic Engineering Department,
Imperial College London, London SW7 2BT, UK
e-mail: christina.morfopoulou08@imperial.ac.uk

E. Gelenbe
e-mail: e.gelenbe@imperial.ac.uk

T. Mahmoodi
Division of Engineering, Centre for Telecoms Research,
Kings College London, London WC2R 2LS, UK
e-mail: toktam.2.mahmoodi@kcl.ac.uk

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_7,

1 Introduction

The carbon imprint of ICT technologies is estimated to be over 2 % of the world total, similar to that of air travel [7]. Yet, research on the energy consumption of ICT systems and its backbone, the wired network infrastructure, is still at an early stage. In this paper we acknowledge the fact that the behaviour of power consumption in today's networks is not proportional to the carried traffic, as has been identified to be the ideal case [1], though, several techniques are being examined as solutions to increase proportionality in future devices. But even if these techniques are applied, this would result in a distinct number of possible operation states, thus a multi step power profile, close to the ideal fully proportional case. Another way investigated in the literature in order to increase energy efficiency of future devices is by putting links or nodes into a sleep state [3].

The implementation of such solutions in network devices will lead to a more complicated behaviour of the power consumption of a network with relation to the carried traffic. Some operating regions may be more efficient than others while "jumps" may arise in power consumption when new traffic is added in the network. The mechanism proposed in this paper acknowledges these changes and aims to keep power consumption in the lowest possible level, by avoiding the more energy demanding operating regions. More specifically, in our experiments we examine the potential savings in energy by using the case of a multi-step power profile in each network node. The admission control mechanism then aims to keep the power consumption at the lowest possible level by restricting the more energy-demanding users and by hopping to the next power consumption level only when necessary. Performance investigations show savings up to 17 % in the total network power consumption revealing that this idea of admission control can be of large importance on top of energy saving mechanisms of future network devices.

2 Previous Work

2.1 Admission Control

Admission control in wired networks has been traditionally used as a way to control traffic congestion and guarantee QoS [6, 13]. The metrics considered in the decision of whether to accept a new flow into a network are mainly bitrate, delay, packet loss and jitter [2, 11]. To the best of our knowledge admission control has never been used as a tool to restrict user entrance in a wired network in order to minimise energy consumption. However, the concept of admitting users according to their power consumption has been used in wireless networks where flows are accepted based on the estimated residual energy or the transmit power of the nodes along a routing path [4, 5].

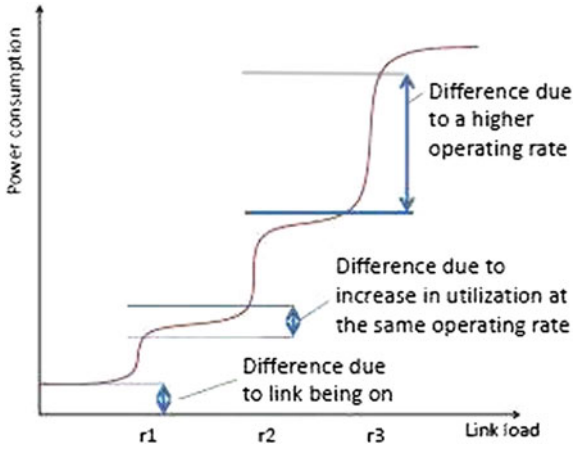


Fig. 1 Predicted power consumption of a device versus load [15]

2.2 Power Consumption in Energy-Aware Networks

Energy proportionality is examined in [14] where the authors explore the potential savings of hardware capable of supporting N performance states, each corresponding to a different link rate. They state that in general, operating a device at a lower frequency can enable dramatic reductions in energy consumption. Also, operating at a lower frequency allows the use of Dynamic Voltage Scaling (DVS) that reduces the operating voltage. DVS is already common in general purpose processors for these reasons and is particularly appealing given that reducing the voltage has a dramatic effect (quadratic decrease) on energy consumption. The technique of Adaptive Link Rate (ALR) assumes that individual links can switch performance states adapting to the carried traffic. Hence the savings that are obtained apply directly to the consumption at the links and interface cards of a network element. Adaptive Link Rate and Dynamic Voltage Scaling is also examined in an energy-aware online technique, proposed in [15], which aims to reduce energy consumption of the backbone internet by spreading the load among multiple paths. Their proposed technique is based on the assumption that the hardware is designed to automatically switch to one of four possible operating rates according to its load and that the power consumption of the hardware would follow the curve shown in Fig. 1.

As described in the survey [3] these techniques of DVS and ALR are widely proposed in order to enable energy efficiency in networks. The result of the application of these techniques would be a multi step profile much closer to the ideal case of energy proportional as shown in Fig. 2, where energy coarsely adapts to the load. Our work builds upon these proposals for future design of networking devices. In this paper we examine the case of multistep power profiles of the network devices, though the same mechanism could be implemented in the single step case (sleep state) or any set of given non-linear power profiles.

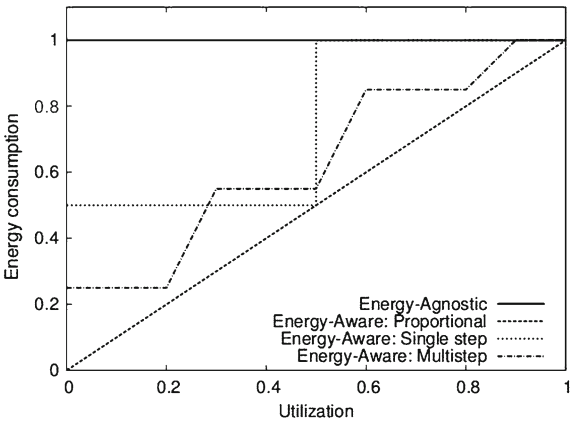


Fig. 2 Power consumption for proportional and non proportional cases [3]

3 Energy Aware Admission Control Mechanism

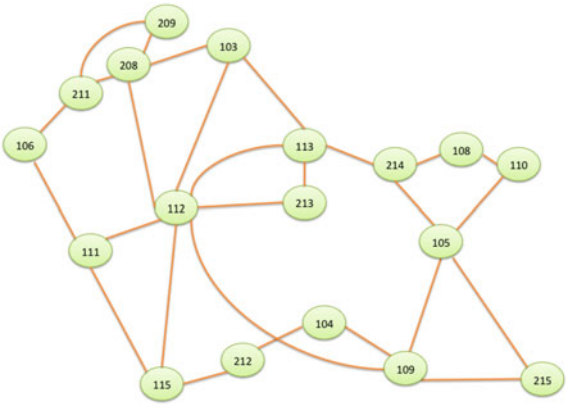
As discussed in the previous section, the proposed algorithm assumes a non linear power consumption behaviour of the network with relation to the carried traffic, where "plateaus" and "jumps" may arise. Since some of the operation areas are more power efficient than others, by using admission control we could reduce the total network energy consumption. Our proposed centralised Energy Aware Admission Control (EAAC) mechanism follows the steps described next:

1. A new user i informs the EAAC about its source s_i, destination d_i and demanded bandwidth bw_i. It also sets a maximum time limit w_i that the user is willing to wait until it is admitted into the network.
2. The EAAC calculates the minimum hop path π_i from s_i to d_i and collects the information about the current power consumption of the nodes n on this path.
3. Using the known power profile and the bandwidth of the flow, it estimates the increase in power consumption after the acceptance of the new flow.

$$\delta P = \sum_{n \in \pi_i} p_n(\lambda_n + bw_i) - \sum_{n \in \pi_i} p_n(\lambda_n) \tag{1}$$

   where p_n is the instantaneous power consumption of node n and λ_n is the current packet rate of the node n on path π_i.
4. If the estimated wattage increase δP is smaller than a fixed value Δ the flow is accepted and admitted into the network (Δ is the threshold in increasing the power consumption that is acceptable by the EAAC). If not, the new flow is sent to a waiting queue. Note that the flows are stored in the waiting queue in a ascending order of their remaining wait time.
5. If a new flow arrives while the mechanism is busy estimating the δP of the previous flow, it joins a request queue. The mechanism checks in first-come-

Fig. 3 Network topology



first-served order the flows in the request queue, going back to step 2. If no flow waits in the request queue, the mechanism picks a waiting user from the waiting queue and follows the same process from step 2.

6. If the waiting time of a flow w_i expires, the flow is immediately admitted into the network, irrelevantly of its estimated power increase.

4 Experiments

4.1 Configuration of the Experiments

In order to evaluate our mechanisms we conducted our experiments on the real testbed located at Imperial College London. Our testbed consists of 18 PC-based routers and we assume that the power consumption profile of these machines has a step-like behaviour as shown in Fig. 4. For less than 100 packets/s a minimal power consumption of 10 W is assumed.

The topology of our experimental testbed is shown in Fig. 3. In the experiments we had 4 users corresponding to 4 Source-Destination (S-D) pairs independently making requests to send traffic into the network. In order to avoid having more than one users requesting to enter the network at the same time, each flow enters a queue ("request queue") at the data gathering point. Thus all users from all source nodes will queue there in order to enter the network. After making a request, each user waits for a random time $intertime$ and then makes a request again. We set this random $intertime$ among requests, so as to have different rates for the arrivals.

Our experiments covered two cases, one with EAAC fully enabled and one with the admission control enabled (flows are queued up) but where the flows are always accepted. The second approach was chosen over not having admission control at all, in order to study the efficiency of our algorithm under the same conditions, since it

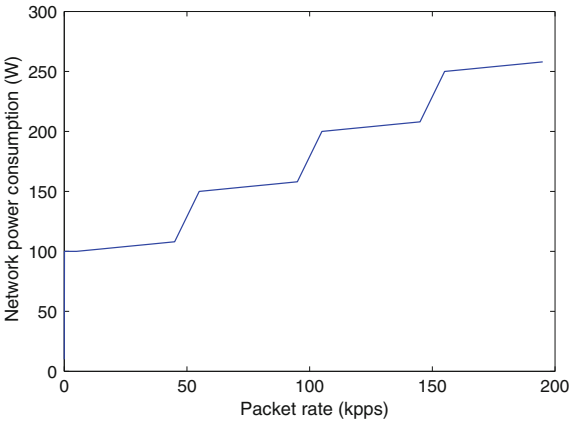


Fig. 4 Step-like router power profile used in the experiments

is a centralised algorithm, and not allowing all users to enter the system at the same time, by itself, contributes to reducing the number of flows entering the network.

We point out that the experiments reported here are not carried out on a standard test-bed that runs the Internet Protocol. Instead, the router software is written for a QoS aware protocol called the "Cognitive Packet Network" (CPN) [9]. However, while CPN can collect energy and QoS information and modify paths so as to minimise such metrics, our experiments were run on a test-bed that uses CPN but which selected all paths to be fixed minimum-hop paths. Furthermore, delays are measured via pinging and energy consumption of the nodes is estimated from the power profile. Therefore we think that the results we obtain will mimic the energy and delay characteristics that one would obtain in a standard IP network.

4.2 Power Consumption

In this experiment we have 4 source-destination pairs (103, 209), (108, 212), (111, 214) and (209, 215). We assume that new flows are generated every $intertime$ seconds, where $intertime$ is randomly distributed between 10 and 40 s. We also assume a random flow duration of 10–30 s and a randomly distributed bandwidth request of 1–10 Mbps. The packet size is set to 100 bytes. Finally, we assume that all the users are willing to wait up to 30 s before they are admitted into the network.

We ran the experiment with our EAAC and without (accepting in the network every new flow). New flows are generated for 300 s. Note that for the EAAC after the 300 s we accept all the flows in order to compare the total energy spent for serving the same amount of users in the network.

The total network power consumption over time, for both cases, is shown in Fig. 5, where the dashed lines correspond to the average values. As we can observe from this figure, there is an average power saving of around 17 %. Note that all the presented

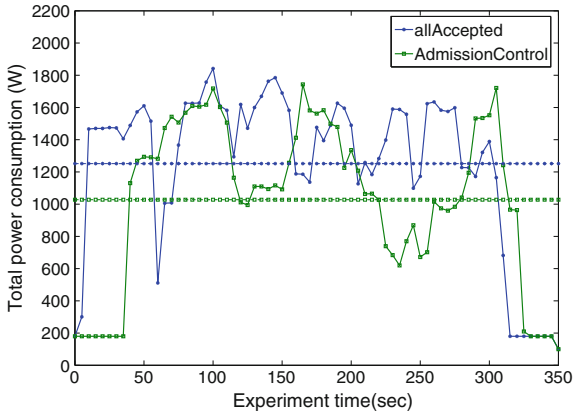


Fig. 5 Network power consumption results for the admission control and no admission control case

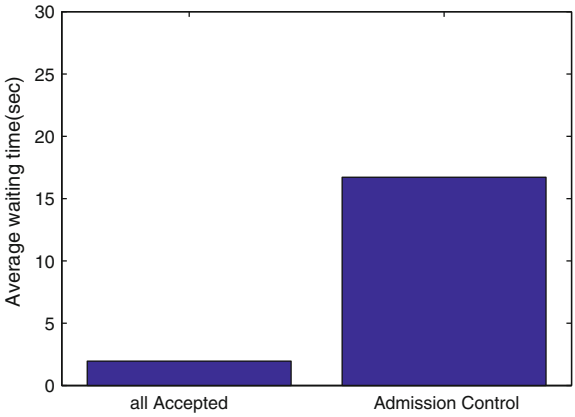


Fig. 6 User average admission waiting time for the admission control and no admission control case

results are averaged over three runs of the experiment. Also, the energy consumption is higher for the EAAC in the last part of the experiment since after the 300th s the EAAC accepts all remaining flows, regardless their energy consumption or waiting times. In Fig. 6 the average waiting time of the users is plotted with and without the EAAC. As expected, the energy saving comes at a cost of delaying the users before they are admitted into the network.

4.3 The Impact of the Δ Value and the Effect of Delaying Traffic

In order to study the effect of the threshold value Δ, we load the network with higher rate of flows' arrivals, i.e. the $intertime$ is set randomly between 10 and 20 s. Several values for Δ are examined and their impact on the energy saving is plotted

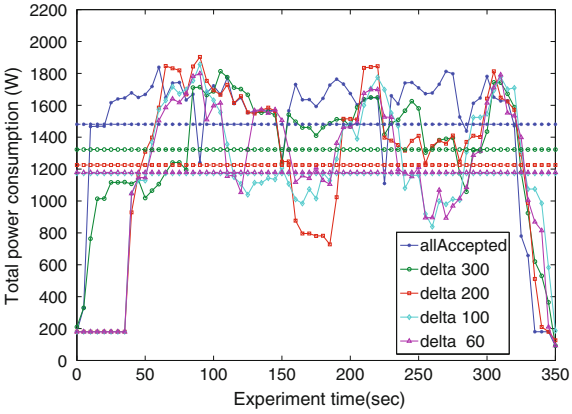


Fig. 7 Network power consumption results for several values of the admission threshold value Δ

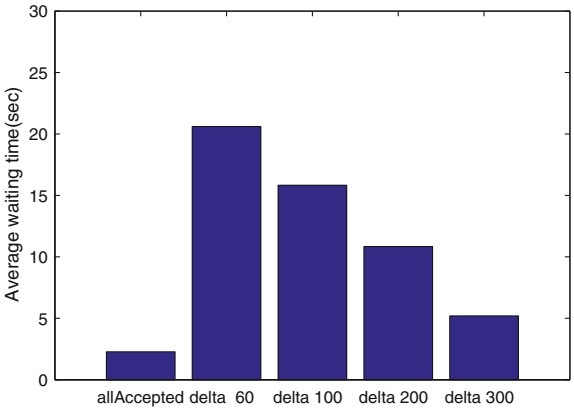


Fig. 8 User average admission waiting time for several values of the admission threshold value Δ

in Fig. 7. These results suggest that the best energy saving is for Δ equal to 60 and 100. Further observation from Fig. 8 reveals that the more strict the Δ value is, the greater the average waiting time is. The results plotted in Fig. 9 clearly show this effect on the total number of admitted flows in the network. Thus, as we can see in Fig. 9, the more strict the Δ value is, the less flows are admitted into the network, and in all cases, the EAAC always accepts less flows compared to the no admission control case.

The value of Δ could affect the efficiency of our proposed method and should therefore be selected carefully. As we can see from Figs. 7 and 8, if the Δ value is too large, the admission control admits almost all new flows and the savings are negligible. On the other hand, if the Δ value is selected to be too small, the admission control will be very strict and users will be obliged to wait until their waiting times expire. The selection of the most appropriate value is not straightforward and should be carefully examined. Our future work will involve finding the optimal value of Δ under several conditions. For example, a careful selection could be made based

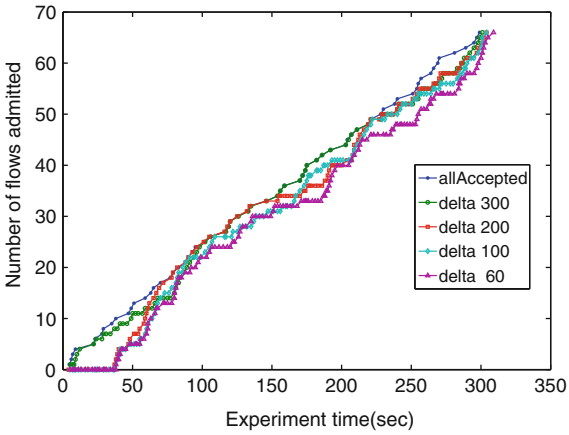


Fig. 9 Number of admitted flows in the network for the for several values of the admission threshold value Δ

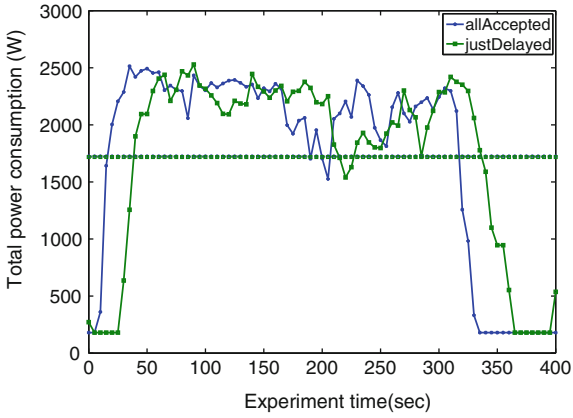


Fig. 10 Network power consumption of no admission control compared to delaying of flows

on the power profile of the nodes. Additionally, this value could also be readjusted online, based on the observed power savings and waiting times.

In order to see whether energy savings result from just delaying the flows, we run an experiment where all new flows are delayed up to their maximum waiting time with maximum waiting times w_j selected uniformly in [10, 40] s. The result is compared to the case where all flows are accepted as soon as they arrive, see Fig. 10, and we observe that initially the power consumption is lower but the overall average is the same. Thus there is no energy saving in just delaying the traffic.

5 Future Work

We have proposed a novel Energy Aware Admission Control mechanism. To the best of our knowledge there is no previous work on Admission Control to improve energy efficiency in wired networks. The experiments and results described here show the

effectiveness of the method and reveal room for potential energy savings, but these energy savings come at the expense of increased waiting delay for users before being admitted into the network. Additional memory transfers and packet processing at the sources can have energy consequences that will be considered and evaluated in future work so that the overall impact of admission control can be considered both on total delay and total energy consumed for the flows and the network. Another aspect that we plan to pursue is the use of analytical models to predict and optimise the energy and delay related to the admission process; mathematical performance modeling tools are well established [8, 12] and will be applied to this particular problem in future work as we have done for energy-aware routing [10].

References

1. Barroso, L., Holzle, U.: The case for energy-proportional computing. Computer **40**(12), 33–37 (2007). doi:10.1109/MC.2007.443
2. Bianchi, G., Borgonovo, F., Capone, A., Fratta, L., Petrioli, C.: Endpoint admission control with delay variation measurements for QoS in IP networks. Comput. Commun. Rev. **32**(2), 61–69 (2002)
3. Bianzino, A., Chaudet, C., Rossi, D., Rougier, J.: A survey of green networking research communications surveys tutorials, IEEE **PP**(99), 1–18 (2010). doi:10.1109/SURV.2011.113010.00106
4. Dilip Kumar, S., Vijaya Kumar, B.: Eaac: energy-aware admission control scheme for ad hoc networks. Int. J. Wirel. Netw. Commun. **1**(2), 201–219 (2009)
5. El-Dolil, S., Al-Nahari, A., Desouky, M., Abd El-Samie, F.S.: Uplink power based admission control in multi-cell wcdma networks with heterogeneous traffic. Prog. Electromagn. Res. B **1**, 115–134 (2008). doi:10.2528/PIERB07101302
6. Floyd, S.: Comments on measurement-based admissions control for controlled-load services. Lawrence Berkeley National Laboratory, Berkeley, CA. Tech. Rep. (1996)
7. Gartner, I.: Gartner estimates ICT industry accounts for 2 percent of global CO_2 emissions (2007). www.gartner.com/it/page.jsp?id=503867
8. Gelenbe, E.: A unified approach to the evaluation of a class of replacement algorithms. IEEE Trans. Comput. **22**(6), 611–618 (1973)
9. Gelenbe, E.: Steps towards self-aware networks. Commun. ACM **52**(7), 66–75 (2009)
10. Gelenbe, E., Morfopoulou, C.: A framewok for energy aware routing in packet networks. Comput. J. **54**(6), 850–859 (2011)
11. Gelenbe, E., Sakellari, G., D' Arienzo, M.: Admission of QoS aware users in a smart network. ACM Trans. Auton Adapt. Syst. **3**(1), 4:1–4:28 (2008)
12. Gelenbe, E., Stafylopatis, A.: Global behavior of homogeneous random neural systems. Appl. Math. Model. **15**(10), 534–541 (1991). doi:10.1016/0307-904X(91)90055-T
13. Lima, S., Carvalho, P., Freitas, V.: Admission control in multiservice IP networks: Architectural Issues and Trends. IEEE Commun. Mag. **45**(4), 114–121 (2007)
14. Nedevschi, S., Popa, L., Iannaccone, G., Ratnasamy, S., Wetherall, D.: Reducing network energy consumption via sleeping and rate-adaptation. In: Proceedings of NSDI'08 5th Symposium on Networked Systems Design and Implementation, pp. 323–336, USENIX Association, Berkeley, CA, USA (2008)
15. Vasic, N., Kostic, D.: Energy-aware traffic engineering. Technical Report, EPFL (2008)

Cost and Benefits of Denser Topologies for the Smart Grid

Giuliano Andrea Pagani and Marco Aiello

Abstract The Smart Grid promises to reshape how electricity is generated, distributed, and used. More delocalized generation based on renewable sources will transform end-users into prosumers (producers and consumers) of energy. These will require electric and supporting ICT infrastructures to be able to openly access the energy market. In this paper, we focus on the electric infrastructure issue related to the Smart Grid topic. We consider network models from the literature of Complex Network Analysis and evaluate their ability to be used for the Distribution Grid to reduce the cost of electricity distribution based on topological property. Our initial conclusion is that denser topologies are helpful to reach the goal. However, the cost of realizing such topologies in terms of cabling is not negligible, as we show.

1 Introduction

The Power Grid has been designed over the years as a hierarchical mono-directional infrastructure with large generation facilities and distribution infrastructure that reaches the end-users. In recent decades, however, unbundling tendencies have begun to change the energy market. Unbundling in the electricity sector proposes to add more players to the market as producers, sellers, or distributors of energy. The goal is to promote competition and innovation in the sector together with better tariffs and services for the consumer [11]. In addition, the availability of affordable small-scale generation facilities (e.g., photovoltaic panels and small wind turbines) shifts

G. A. Pagani (✉) · M. Aiello
Johann Bernoulli Institute for Mathematics and Computer Science,
University of Groningen, Nijenborgh 9, 9747 AGGroningen,
The Netherlands
e-mail: g.a.pagani@rug.nl

M. Aiello
e-mail: m.aiello@rug.nl

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_8,

the generation towards the periphery of the infrastructure [13]. Such trends combined lead to the emergence of a new figure in the energy panorama: the *prosumer*. This term characterizes the new actors, who are both producers and consumers of energy, operating in this scenario. They are increasing in number, and will most likely demand a market with total freedom for energy trading.

With generation moving massively to a local scale, the Power Grid will require an update to evolve into a more efficient and information-driven system, a fact that we take as a defining characteristic of the Smart Grid to come. In particular, the Medium and Low Voltage layers of the Grid are likely to be affected by the energy produced and consumed by prosumers. Therefore, we predict that the current Medium and Low Voltage Grid will be an enabler or a repressor for the transition to an electricity system mainly based on prosumers. The Grid and its electricity distribution cost will determine the success of energy exchanges at the local level.

Based on our previous analysis of the topology of the Dutch Grid and the identification of a relationship between costs for electricity distribution and topology [16], in this paper we take a closer look at the cost and benefits of realizing the Medium and Low Voltage Grid with denser type of network (i.e., a network with an increased number of connections) compared to the current infrastructure. We place particular emphasis on assessing the cost of the current Medium and Low Voltage Grid based on actual cable pricing information. To evaluate the benefits of networks denser than the current one, we use statistical topological metrics that are associated with electricity distribution costs.

The paper is organized as follows. In Sect. 2, we introduce Complex Network Analysis (CNA), our main tool for topological investigation and the principles followed in designing denser networks. Section 3 focuses on the costs of realizing denser electrical Grids and the accompanying benefits. The main related work of the literature is summarized in Sect. 4. The conclusion of the paper is provided in Sect. 5.

2 Complex Network Analysis and the Power Grid

Complex Network Analysis is a branch of Graph Theory taking its root in the early studies of Erdős and Rényi [7] on random graphs and considering statistical structural properties of very large graphs. The first systematic studies appeared in the late 1990s, e.g. [3, 21], having the goal of looking at the properties of large networks with complex systems behavior. Since then, Complex Network Analysis has been used in many diverse fields of knowledge, from biology to chemistry, from linguistics to social sciences, from computer networks and the web to virus spreading, to logistics and also inter-banking systems [2]. Man-made infrastructures are especially interesting to study under the Complex Network Analysis lenses, especially when they are large-scale and grow in a decentralized and independent fashion, thus not being the result of a global design, but rather of many local autonomous designs.

CNA techniques that have been applied to the Power Grid, mainly focused on the reliability of the High Voltage Grid. The studies appeared after blackouts of important electricity infrastructures (e.g., U.S. Grid and Italian Grid), it is thus not surprising they mainly focused on reliability issues [1, 4, 5].

In our own previous study [16], we analyzed the Dutch Medium and Low Voltage Grid. The main findings are as follows: first, the network samples show a small average connectivity (average node degree $\langle k \rangle = 2.009$ for the Low Voltage and $\langle k \rangle = 2.129$ for the Medium Voltage); second, it is possible to roughly categorize the networks based on the number of nodes; and third, there is no clear evidence of a specific topological structure belonging to a well-known model in the 24 samples analyzed. This last point refers to the absence of characteristics that possess other complex networks from biology or technology, such as Scale-Free or Small-World properties. In our follow-up study [17], we considered network models that have proven successful in showing salient characteristics of technological networks and we analyzed possible topological evolutions of the current Grid to see which topology is best suited for supporting local-scale energy exchange. For each model, we considered three values of increasing average node degree ($\langle k \rangle = 2$, $\langle k \rangle = 4$, and $\langle k \rangle = 6$) to study the effects of increasing connectivity on the performance of the network. As a general result, we see that just considering the topology provides benefits to the efficiency of the network in fundamental aspects such as characteristic path length, clustering coefficient, and network reliability against disruption. Naturally, creating denser topologies for a physical infrastructure translates into higher cable deployment costs.

3 Economic Considerations

Traditionally, the problem of evaluating the expansion of an electrical system is a complex task that involves both the use of modeling, usually based on operation research optimization techniques and linear programming [8], as well the experience and vision of experts. However, with more distributed generating facilities at local scale, traditional methods have limits and need to be modified or updated to take into account the new scenario the Smart Grid brings into play. The models that we have analyzed in [17] also need to be evaluated from the economic point of view. How much will it cost to build electrical infrastructures according to these models? What is the actual cost of adding a physical edge to the topology?

One important difference between a physical infrastructure such as the Power Grid and the WWW or social networks is the physical presence of cables that connect the Medium Voltage substations or Low Voltage end-users' generating units. While establishing a link from a Web page to another one is free, each increase in connectivity in the Power Grid implies costs in order to build or adapt the substation or end-user premise involved, as well as costs for the cables required for the connection. To assess these costs in the Medium and Low Voltage infrastructure,

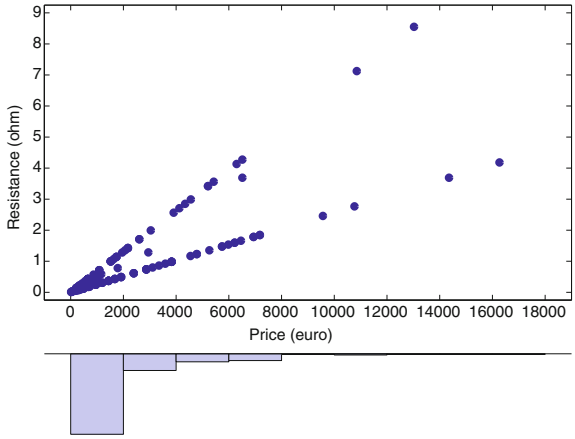


Fig. 1 Price-resistance pairs joint plot

we consider a simple relation where the cost of cabling and cost of substations are added:

$$C_{impl} = \sum_{j=1}^{N} Ssc_j + \sum_{i=1}^{M} Cc_i \tag{1}$$

where C_{impl} stands for cost for implementation, Ssc_j is the adaptation cost for the substation j and Cc_i is the cost for the cable i. The cost of the cable can be expressed as a linear function of the distance the cable i covers: $Cc_i = C_{uc_i} \cdot l_i$, where C_{uc_i} is the cable cost per unit of length and l_i is the length of the cable. There are several types of cables used for power transmission and distribution with varying physical characteristics and costs. In addition, the cost for installation can vary significantly [14]. In the present work, to provide an initial estimate, we simply consider cabling costs and ignore substation ones. While the former are directly tied to the topology and length of the links, the latter pricing is too dependent on other factors (e.g., different equipment in the substation). As a source of data for cable type and pricing, we have been provided (courtesy of Enexis B.V. the Netherlands) with cable characteristics and prices, together with topological information, for 11 network samples belonging to the Low Voltage network and 13 samples belonging to the Medium Voltage network of the Northern Netherlands.

The length of the cables plays an important role for both total resistance (therefore losses) and price. If one considers the correlation between the price and resistance, high values are found using Spearman's rank correlation coefficient, shown in Table 24 in [17]. For generating synthetic networks it is especially important to obtain values for both the properties of cables that are similar to the ones used in practice. A plot of the two variables characterizing each cable reveals that the majority of the samples concentrate in the lower tails of the joint distribution. Figure 1 shows the relation between the price and resistance where the values concentrate in the lower corner of $price \times resistance$. In the chart in Fig. 1, two distinct lines deviate

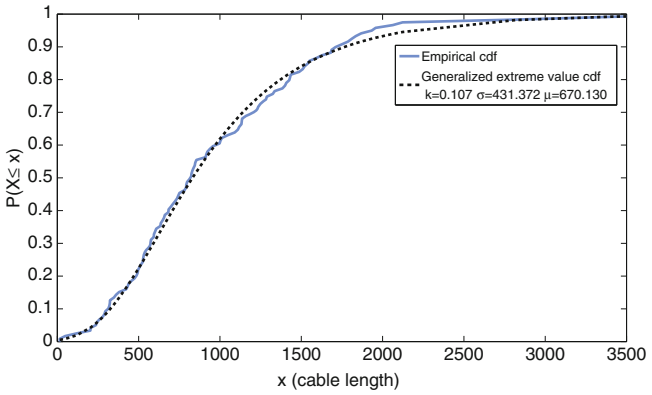


Fig. 2 Cumulative distribution function for cable length for cable type "3x1x70al" in Northern Netherlands medium voltage

from the lower left corner. They represent the two main types of cables to be used in that sample of the Low Voltage network to cover different distances and result in increasing price and resistance for longer lines. The problem of extracting cable properties can be, however, approached in another way: *evaluate for each type of cable (i.e., physical property and technology) used in a certain category of sample belonging to the Medium or Low Voltage network (Small, Medium and Large based on the number of nodes* [17]), *how the lengths of the cables used are distributed.* In fact, given a certain type of cable and its length, all other relevant properties for our analysis are then available (i.e., cable total resistance, total cost and supported current).

When fitting the distribution of lengths to cable types belonging to Low Voltage and Medium Voltage, one notes a rapid decay in the probability distribution, with the majority of lengths for the Low Voltage cable types on the order of tens of meters, and Medium Voltage cables in the hundreds of meters. Fitting the length to a statistical probability distribution gives a good approximation for the Low Voltage cable lengths as exponential distributions $\left(y = f_X(x;\mu) = \frac{1}{\mu} e^{\frac{-x}{\mu}}\right)$, while for Medium Voltage cable lengths, the generalized extreme value distribution fits best $\left(y = f_X(x;k,\mu,\sigma) = \frac{1}{\sigma}\left(1 + k\frac{x-\mu}{\sigma}\right)^{-1-\frac{1}{k}} \exp\left\{-(1 + k\frac{x-\mu}{\sigma})^{-\frac{1}{k}}\right\}\right)$; these hypotheses are supported by the Kolmogorov-Smirnov test results. An example is shown in Fig. 2.

Assume that, statistically speaking, the distribution of the lengths for each type of cable in the synthetic networks is the same as in the physical samples. Therefore, once we know the probability of using a certain type of cable i ($p_{cable_i} = \frac{\#cable_i}{\sum_k \#cable_k}$ where $\#cable_i$ is the number of occurrences of cable type i in a certain network sample) that has a certain cost and resistance per meter and a specific current supported, we can estimate the cables that are used in the synthetic samples together with their properties.

Table 1 Cabling cost for $\langle k \rangle \approx 2$ synthetic samples

Sample type	Size	Cost (thousand euro)
Low voltage—small	≈20	≈30
Low voltage—medium	≈90	≈78
Low voltage—large	≈200	≈449
Medium voltage—small	≈250	≈32000
Medium voltage—medium	≈500	≈42000
Medium voltage—large	≈1000	≈43000

Given the information about cable prices, it is possible to estimate the cost for realizing a network with a certain connectivity and to determine whether such networks are able to lower the (economic) barrier towards decentralized energy trading. The results for Low Voltage and Medium Voltage networks for *Small*, *Medium* and *Large* types with an average node degree $\langle k \rangle \approx 2$ are shown in Table 1. The results for $\langle k \rangle \approx 4$ and $\langle k \rangle \approx 6$ are about two and three times more expensive since there is an increase in the number of edges by the same quantity. The small difference in costs between the *Medium* and *Large* types of networks for Medium Voltage is related mainly to the different technologies of cable types that are used for these types of networks.

The cost in realizing infrastructures with more connectivity compared to the current infrastructures is not the only aspect of comparison. It is essential to show how this additional connectivity provides benefits in the form of a decrease of the costs of electricity distribution. In our previous work [16] we defined two sets of metrics (α and β) to assess the topological aspects that influence the cost of electricity. In particular, α considers the aspects that are related to losses in the network, while β deals with reliability and capacity properties of the network. In order to compare on the same basis (i.e., considering α and β metrics) the physical samples of the Northern Netherlands and generated networks, it is essential to associate to the generated networks realistic physical properties such as resistance and supported current. These properties can be extracted from each physical sample (i.e., Medium or Low Voltage and its *Small*, *Medium* or *Large* category) and associated to the corresponding generated synthetic samples. This mapping can be done with the assumption that, statistically, the properties of cables in the new networks (i.e., synthetic) will remain the same as in the current networks (i.e., physical samples). To enable this mapping the statistical analysis of the physical samples shown above is the necessary tool.

The comparison for the electricity cost based on the topological parameters for Low Voltage networks is shown in Fig. 3. Red dots in the $\alpha \times \beta$ plane represent the Northern Netherlands samples while the white diamonds represent the generated Small-World networks. Small-World has been chosen for the comparison since it is the network model that scores best in the pure topological comparison [17]. One sees that when the connectivity is sufficiently high (i.e., $\langle k \rangle \approx 4$), the synthetic samples score better than the physical ones. On average for the α metric, the improvement is about 50 % compared to the Netherlands samples, while about 60 % when the connectivity is increased to $\langle k \rangle \approx 6$. Considering the β metric, the improvement are

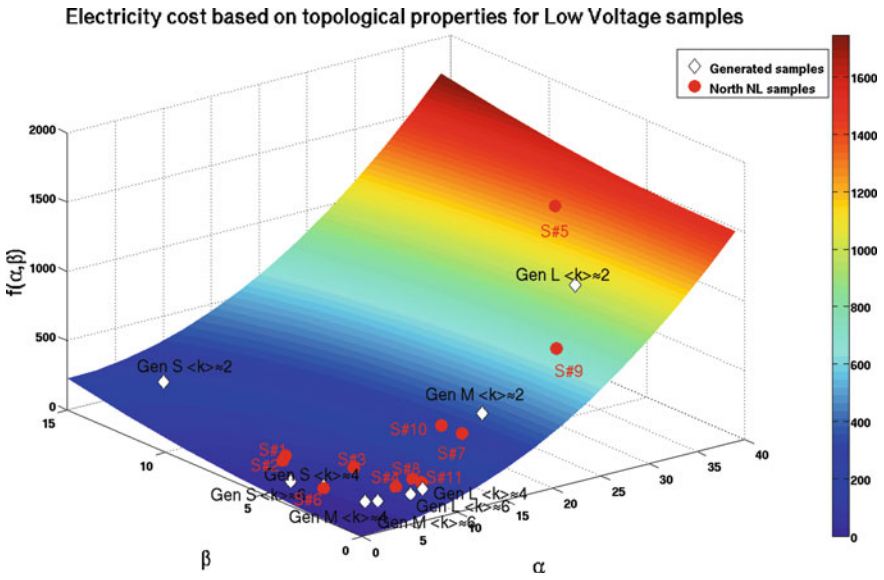


Fig. 3 Comparison of the transport cost between synthetic and real low voltage grids

30 and 40 % for the $\langle k \rangle \approx 4$ and $\langle k \rangle \approx 6$ situations, respectively. Similar considerations apply to the Medium Voltage samples with improvements that reach up to 60 % compared to the physical samples when higher connectivity is added ($\langle k \rangle \approx 6$).

4 Related Work

CNA for network growth and evolution is mainly used in the field of physics, considering man-made or natural networks [6] or social networks [12]. The Internet has also been the subject of investigation in its topological evolution [19]. The main aim of these works is to present and describe the evolution of such networks, rather than offering a design tool for the infrastructure. Approaches that apply CNA to the Power Grid essentially investigate the current Grids analyzing their topological properties. Most of the work focuses on either considering the membership of a network in a certain category or on evaluating the reliability and tolerance to failures of the network [15]. Only very few studies consider the improvement of Power Grids taking into account the addition of few power lines and their topological benefit [10, 18]. Wang et al. [20] applied Complex Network Analysis to analyze the Smart Grid mainly to understand the communication infrastructure and network topologies needed to support decentralized control. However, these publications consider once again only the High Voltage Grids. In practice, electrical engineers consider several aspects: economics, environment, feasibility, and concurrent safety [9]. Taking into

consideration all these aspects makes the task of planning a complex decision problem with multiple objectives.

5 Concluding Remarks

The Smart Grid promises a new approach to energy generation and distribution where the Medium and Low Voltage Grid will change role and importance. In fact, the topology of the network plays an important role, in influencing the costs of electricity distribution. We have proposed network models from the literature of Complex Network Analysis and investigated how topologies with increased connectivity (i.e., higher node degree) could be beneficial in lowering those parameters that influence the price of distributing electricity. On the other hand, we note an the increase in costs for denser topologies. Our approach does replace the current planning techniques used by energy distributors, but it aims at being a decision support tool in evaluating new strategies for the future energy panorama.

Acknowledgments The work is supported by the EU FP7 Project GreenerBuildings, contract no. 258888 and by the Dutch National Research Council, contract no. 647.000.004. Pagani is supported by University of Groningen with the Ubbo Emmius Fellowship 2009.

References

1. Albert, R., Albert, I., Nakarado, G.L.: Structural vulnerability of the North American power grid. Phys. Rev. E **69**(2), 025103 (2004)
2. Barabási, A.L.: Linked: the new science of networks. Am. J. Phys. **71**(4), 409–410 (2004)
3. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509 (1999)
4. Chassin, D.P., Posse, C.: Evaluating North American electric grid reliability using the Barabási Albert network model. Phys. A Stat. Mech. Appl. **355**, 667–677 (2005)
5. Crucitti, P., Latora, V., Marchiori, M.: A topological analysis of the Italian electric power grid. Phys. A Stat. Mech. Appl. **338**(1–2), 92–97 (2004)
6. Dorogovtsev, S.N., Mendes, J.F.F.: Evolution of Networks: From Biological Nets to the Internet and WWW. Oxford University Press, New York (2003)
7. Erdős, P., Rényi, A.: On random graphs I. Publ. Math. Debrecen **6**, 290–297 (1959)
8. Garver, L.: Transmission network estimation using linear programming. IEEE Trans. Power Apparatus Syst. **PAS-89**(7), 1688–1697 (1970)
9. Grigsby, L.L. (ed.): The Electric Power Engineering Handbook. CRC Press, Boca Raton (2007)
10. Holmgren, A.J.: Using graph models to analyze the vulnerability of electric networks. Risk Anal. **26**(4), 955–969 (2006)
11. Joskow, P.L.: Lessons learned from electricity market liberalization. Energy J. **29**(Special I), 9–42 (2008)
12. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. J. Am. Soc. Inf. Sci. Technol. **58**(7), 1019–1031 (2007)
13. Lovins, A.B., Datta, E.K., Feiler, T., Rabago, K.R., Swisher, J.N., Lehmann, A., Wicker, K.: Small is Profitable: The Hidden Economic Benefits of Making Electrical Resources the Right Size. Rocky Mountain Institute, Snowmass (2002)

14. National Grid: Undergrounding high voltage electricity transmission—the technical issues. Technical Report, National Grid (2009)
15. Pagani, G.A., Aiello, M.: The power grid as a complex network: a survey. Technical Report, JBI, University of Groningen. arXiv:1105.3338 (2011)
16. Pagani, G.A., Aiello, M.: Towards decentralization: a topological investigation of the medium and low voltage grids. IEEE Trans. Smart Grid **2**(3), 538–547 (2011)
17. Pagani, G.A., Aiello, M.: Power grid network evolutions for local energy trading. Technical Report, JBI, University of Groningen. arXiv:1201.0962 (2012)
18. Rosato, V., Bologna, S., Tiriticco, F.: Topological properties of high-voltage electrical transmission networks. Electr. Power Syst. Res. **77**(2), 99–105 (2007)
19. Vázquez, A., Pastor-Satorras, R., Vespignani, A.: Large-scale topological and dynamical properties of the Internet. Phys. Rev. E **65**(6), 1–12 (2002)
20. Wang, Z., Scaglione, A., Thomas, R.J.: Generating statistically correct random topologies for testing smart grid communication and networks. IEEE Trans. Smart Grid **1**(1), 28–39 (2010)
21. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393**(6684), 440–442 (1998)

Utility-Based Time and Power Allocation on an Energy Harvesting Downlink: The Optimal Solution

Neyre Tekbiyik, Elif Uysal-Biyikoglu, Tolga Girici and Kemal Leblebicioglu

Abstract In this paper, we consider the allocation of power level and time slots in a frame to multiple users, on an energy harvesting broadcast system. We focus on the offline problem where the transmitter is aware of the energy arrival statistics of a frame before the frame starts. The goal is to optimize throughput in a proportionally fair way, taking into account the inherent differences of channel quality among users. Analysis of structural characteristics of the problem reveals the biconvex nature of the problem. Due to biconvexity, a Block Coordinate Descent (BCD) based optimization algorithm that converges to one of the multiple optima is proposed. Simulation results show that the resulting allocation achieves a good balance between total throughput and fairness.

Keywords Broadcast channel · Energy harvesting · Offline algorithms · Proportional fairness · Time sharing

This work was supported by TUBITAK Grant 110E252.

N. Tekbiyik (✉) · E. Uysal-Biyikoglu · K. Leblebicioglu
Department of Electrical and Electronics Engineering, Middle East Technical University, 06800 Ankara, Turkey
e-mail: ntekbiyik@eee.metu.edu.tr

E. Uysal-Biyikoglu
e-mail: elif@eee.metu.edu.tr

K. Leblebicioglu
e-mail: lebleb@eee.metu.edu.tr

T. Girici
Department of Electrical and Electronics Engineering, TOBB Economics and Technology University, 06560 Ankara, Turkey
e-mail: tgirici@etu.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_9,

1 Introduction

Recent advances in the areas of solar, piezoelectric and thermal energy harvesting, enable systems that are capable of energy harvesting. Communication devices may be powered by rechargeable batteries which may harvest energy through solar cells, vibration absorption devices, thermoelectric generators, wind power, etc. Such energy harvesting abilities can allow sustainable and environmentally friendly deployment of wireless communication networks. However, this renewable energy supply feature also calls for specific design principles to efficiently utilize the dynamic levels of instantaneously available energy. Hence, energy harvesting shifts the nature of energy-aware solutions developed for transmitters from minimizing energy expenditure to optimizing it over time.

Recently, many researchers have focused on optimizing data transmission with an energy harvesting transmitter. A single-user communication system operating with an energy harvesting transmitter is considered in [1], where a packet scheduling scheme that minimizes the time by which all of the packets are delivered to the receiver is obtained. A multi-user extension of [1] has also been considered in [2, 3] and the same time minimization problem is solved for a two user broadcast channel. These approaches are extended by Tutuncuoglu and Yener [4], to the case of a transmitter with a finite capacity battery, and by Ozel et al. [5] to the case of a transmitter operating in a fading channel. Ho and Zhang [6] also consider a time varying channel and energy source for point-to-point wireless communications and use convex optimization techniques to obtain the throughput-optimal energy allocation.

Unlike the studies summarized above, we are interested in proportional fairness among users. Thus, we consider allocating among users the transmission power and the proportion of the time between energy harvests, to maximize the throughput in a proportionally fair way, taking into account the inherent differences of channel quality among users. Specifically, we investigate the proportional fairness based utility maximization problem in a multi-user time-sharing additive white Gaussian noise (AWGN) broadcast channel, where the transmitter is capable of energy harvesting. We focus on the offline problem in which the energy arrival profile (the energy harvesting times and amounts) of a time window, called frame, is known to the transmitter at the beginning of that frame. This problem has two challenging aspects; discovering a fair allocation of the energy interarrival times for all users and, at the same time, finding a power allocation that does not violate the set of energy causality constraints (energy may not be used before it is harvested) to maximize the utility. With the analysis performed to reveal the characteristics of the utility function, the problem presented in this paper is shown to be a biconvex problem [7] with multiple optima. The presented optimization algorithm, BCD, surely converges to a partial optimum (see Sect. 3), and thus, the partial optimal utility of the problem.

We start by describing the problem formulation and structural properties in the next section. BCD algorithm is described in Sect. 3. In Sect. 4, we present our numerical and simulation results. We conclude in Sect. 5 with an outline of future directions.

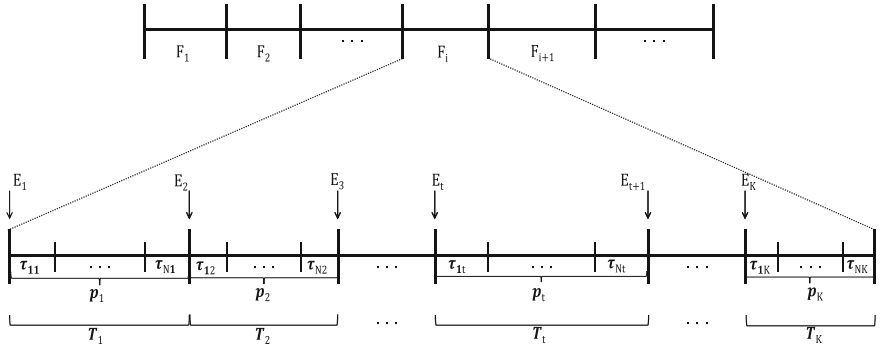


Fig. 1 Problem illustration: there are K energy arrivals in a frame, and, the time between consecutive arrivals are allocated to N users

2 Problem Formulation and Structural Properties

Consider an energy harvesting system in which a single transmitter transmits to N users by time sharing, i.e., each frame, of length F_i, is divided into K slots. Channel remains stable during F_i (g_n, the gain of user n, is constant throughout the frame). The transmitter is equipped with a rechargeable battery such that $E_{ti} > 0$ amount of ambient energy is harvested at the beginning of each time slot t of frame i. The tth slot of frame i is of length T_{ti}. In this paper, we are interested in a specific frame, therefore, we drop the frame indicator i and define the harvested energy in slot t as E_t, and, the length of slot as T_t, as illustrated in Fig. 1. Please note that, the slot lengths do not necessarily need to be equal as the energy arrivals may occur in different moments in time.

In this offline problem, for a given frame, the transmitter needs to choose a power level p_t and a time allocation vector $\tau_t = (\tau_{1t}, \ldots, \tau_{Nt})$, for each time slot t of the frame, where $p_{nt} = p_t$ is the selected transmission power for user n during slot t and, τ_{nt} is the time allocated for transmission to user n during slot t. Our main objective is to maximize a proportionally fair utility function [8], the log-sum of the user rates $\sum_{n=1}^{N} log_2(R_n)$ where $R_n = \sum_{t=1}^{K} \tau_{nt} W \log_2\left(1 + \frac{p_t g_n}{N_o W}\right)$. Hence, we define the following constrained optimization problem which aims to maximize the utility function with respect to the desired constraints.

Problem 1

$$\text{Maximize:} U(\overline{\tau}, \overline{p}) = \sum_{n=1}^{N} \log_2\left(\sum_{t=1}^{K} \tau_{nt} W \log_2\left(1 + \frac{g_n p_t}{N_o W}\right)\right)$$

$$\text{subject to: } \tau_{nt} \geq 0, \;\; p_t \geq 0 \tag{1}$$

$$\sum_{n=1}^{N} \tau_{nt} = T_t \tag{2}$$

$$\sum_{t=1}^{K} \tau_{nt} \geq \epsilon \tag{3}$$

$$\sum_{i=1}^{t} p_i T_i \leq \sum_{i=1}^{t} E_i \tag{4}$$

where $t = 1, \ldots, K$ and $n = 1, \ldots, N$ and, W and N_o are the bandwidth for a single link channel and the power spectral density of the background noise, respectively. (1) represents the nonnegativity constraints. The set of equations in (2) ensure that the total time allocated to users does not exceed the slot length. The ones in (3), on the other hand, are technical constraints that ensure some time ($\geq \epsilon$ where ϵ is an infinitely small number) during the frame, for every user. Finally, the energy causality constraints defined in (4) ensure no energy is transmitted before becoming available. Note that, Problem 1 is a nonlinear non-convex problem with potentially multiple local optima. Therefore, we can only expect that by proper choice of the initial value, our algorithm converges to a stationary point that is nearby the true optimum. In [7], we prove that $-U(\overline{\tau}, \overline{p})$ is a biconvex function, and that, Problem 1 is a biconvex optimization problem. We use this information to develop such an algorithm.

3 Solution Method

As known, $-U(\overline{\tau}, \overline{p})$ is a biconvex function. While not convex, such functions admit efficient coordinate descent algorithms that solve a convex program at each step. Therefore, we present a block coordinate descent based algorithm (BCD) for solving Problem 1. In a BCD, sequentially one block of variables is minimized under corresponding constraints and the remaining blocks are fixed. As we have only two block variables $\overline{\tau}$ and $\overline{p}$, the algorithm alternates between minimization with respect to $\overline{\tau}$ and minimization with respect to $\overline{p}$. BCD operates as follows:

1. Start from any valid time allocation. Assuming that all of the energy E_t is used up until the end of period t, determine the power allocation. This power allocation should satisfy Eq. (4).
2. Keep τ_{nt} fixed for all n and t. Optimize $U(\overline{\tau}, \overline{p})$ with respect to p_t, $t = 1, \ldots, K$ and constraint in Eq. (4).
3. Repeat the following for all $t = 1, \ldots, K$.: Keep τ_{ni} fixed for all $n = 1, \ldots, N$ and $i \neq t$. Also keep p_t fixed for all t. Maximize $U(\overline{\tau}, \overline{p})$ with respect to τ_{nt}, $n = 1, \ldots, N$ and constraint in Eq. (2).
4. If the variables converged, stop. Otherwise, go to Step 2.

We use the well-known Lagrange multiplier method for the optimization of the time variables. For the optimization of the power variables, however, the Sequential Unconstrained Minimization Technique (SUMT) [9] is used. SUMT method converts a constrained optimization problem into an unconstrained one by adding the

constraints to the objective function as a "penalty". It then uses an interior algorithm such as Newton's optimization algorithm [10] to solve the problem with the penalty-added objective function.

Regarding the issue of convergence, Problem 1 is a biconvex optimization problem and thus, there exist many local optima. Hence, convergence to the global optimum is not always guaranteed. However, it is possible to converge to a stationary point by using a block coordinate descent method, provided that the sub-problems have unique solutions [11]. Unfortunately, in our case, the time allocation problem is not strictly convex (only convex) and thus, it has multiple optima. Fortunately, in [7] we prove that, BCD converges to a stationary point, where each stationary point is a partial optimum of the problem, and, all partial optima yield the same utility value. The following definition explains the concept of partial optimum.

Definition 1 *Let* $f : B \rightarrow \Re$ *be a given function and let* $(x^*, y^*) \in B$. *Then,* (x^*, y^*) *is called a partial optimum of* f *on* B, *if*

$$f(x^*, y^*) \leq f(x, y^*) \ \forall x \in B_y^* \quad \textit{and} \quad f(x^*, y^*) \leq f(x^*, y) \ \forall y \in B_x^* \tag{5}$$

4 Numerical and Simulation Results

In this section, we present the numerical and simulation results related to BCD algorithm. Throughout our simulations we use the following setup: $W = 1\,\text{kHz}$, $N_o = 10^{-6}\,\text{W/Hz}$. There happens to be 10 energy arrivals in 100 s. The arrivals are $\overline{E} = [20, 100, 1, 1, 1, 70, 100, 1, 10, 40]$ J in the [1st, 2nd, ..., 10th] slots respectively. First, we assume that there are 5 users in the system. The first user is the strongest one, and, other users are ordered in a such way that the preceding user is twice as strong as the next one, i.e., path losses of the users are; 25, 28, 31, 34, 37 dB respectively. The starting point of the BCD algorithm is the Spend What You Get (SG) policy [12] combined with TDMA time allocation. We performed simulations both for unequal and equal slot lengths. When the slot lengths are not equal ($T_j = T_i$ iff $j = i$), we used [10, 12, 5, 7, 4, 15, 20, 2, 10, 15] sequence and for periodic energy arrivals, we used the equalized version of that sequence, i.e., ($T_j = 10$ for $j = 1, \ldots, 10$). The optimal schedules (power and time), optimal utility and thus, the utility improvement (when compared to SG + TDMA) obtained by BCD, for the these sequences are presented in Table 1. In the table, power and time units are Watts and seconds, respectively.

In recent energy harvesting systems, transmitters have supercapacitors that store the harvested energy and supply in every predetermined time window, allowing the case of periodic energy arrivals. In such a case, if no energy is harvested within a slot, the amount of harvested energy is set to 0 for that slot. As observed from Table 1, periodic energy arrivals assumption does not degrade the system performance, yet increases the utility improvement. In [13] we have shown that, by using the periodic energy arrivals assumption we can analytically derive the power related

Table 1 The results of BCD algorithm for two different slot length sequences

		Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8	Slot 9	Slot 10	Utility	Utility Imp.
Time Allocation	Users vs. Slot Lengths	10	12	5	7	4	15	20	2	10	15		
	1	10	12	0	0	0	0	3.1288	0	0	0		
	2	0	0	0	0	0	5.7638	16.8712	0	0	0		
	3	0	0	0	7	4	9.2362	0	0	0.1987	0		
	4	0	0	0	0	0	0	0	0	9.8013	6.9208	75.7273	% 8.5449
	5	0	0	5	0	0	0	0	2	0	8.0792		
Power Allocation		2	2.0910	5.9724	3.4337	3.4337	3.1027	2.5535	5.9723	4.4268	5.1636		
Time Allocation	Users vs. Slot Lengths	10	10	10	10	10	10	10	10	10	10		
	1	10	10	6.2337	0	0	0	0	0	0	0		
	2	0	0	3.7663	10	10	0	0	0	0	0		
	3	0	0	0	0	0	10	10	0	0	0		
	4	0	0	0	0	0	0	0	10	6.3094	0	75.7325	% 9.6133
	5	0	0	0	0	0	0	0	0	3.9606	10		
Power Allocation		2	2.0182	2.2189	2.5923	2.5923	3.4327	3.4327	4.6482	5.1876	6.2772		

characteristics of the optimal solution of Problem 1 and, develop a simple heuristic that closely tracks the performance of BCD algorithm. Hence, from now on, we present results only for the case of periodic energy arrivals.

In order to test the effect of number of users, on the performance of the BCD algorithm, keeping the number of harvests and harvest values same, we perform series of simulations with different number of users. We aim to observe the changes in utility improvement and fairness. In order to be able to analyze all scenarios, in the next three figures, we use the following setups: (a) The strongest user in the system has 13 dB path loss, and, every new user that joins the system deviates 3 dB from the previous one (has 3 dB more path loss than the one who joined before him/her). (b) The strongest user has 19 dB path loss, and, every new user deviates 3 dB. Figure 2 shows how utility and throughput improvement change with number of users. Clearly, at all instances, BCD outperforms SG + TDMA schedule, i.e., even with a few users, utility and throughput can be improved. The results show that as path losses of the users increase, the utility improvement increases, i.e, BCD's utility improvement performance improves as the channel quality becomes degraded. For example, when case (b) is valid, a utility improvement of approximately 15 % is possible with BCD. Note that, this improvement is considered to be high, since the utility to be maximized is in the form of $log(log(.))$.

In order to measure the fairness of the BCD algorithm, the well-known Jain's fairness index (FI) [14] is used.

$$FI = \frac{(\sum_{i=1}^{N} x_i)^2}{N \cdot \sum_{i=1}^{N} x_i^2} \tag{6}$$

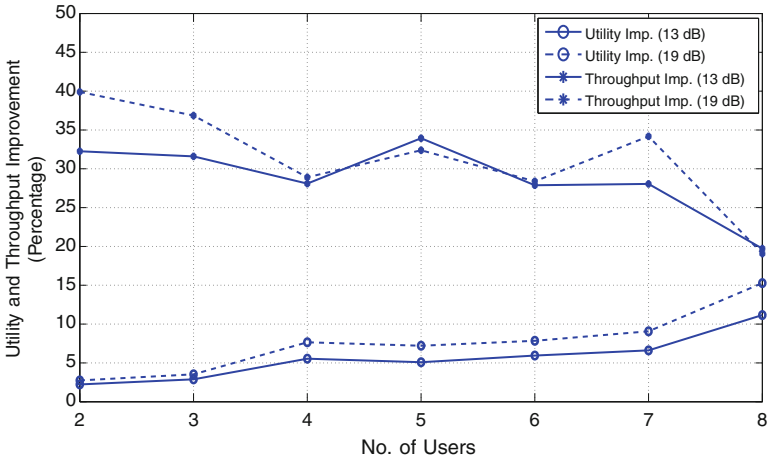


Fig. 2 Utility and total throughput improvement versus number of users: the improvement of BCD over SG + TDMA, for increasing number of users, are compared. The effect of path loss on utility improvement is shown

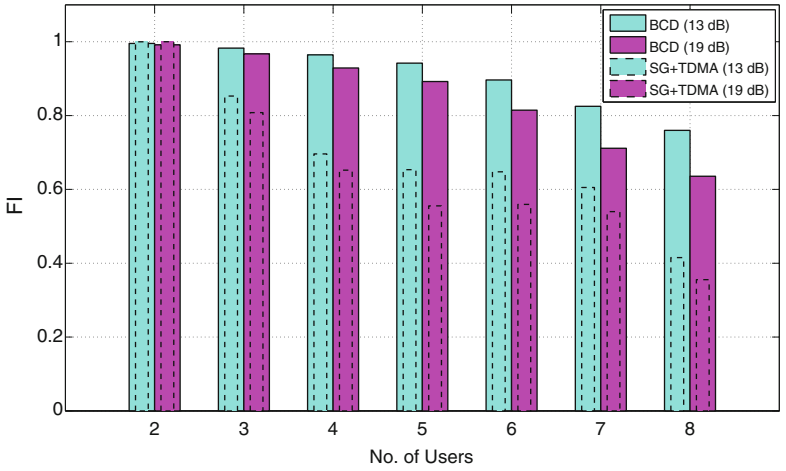


Fig. 3 Fairness index (SG + TDMA, BCD) versus number of users: the fairness of SG + TDMA and BCD, for increasing number of users, are compared through FI. The effect of path loss on fairness is also shown

where x_i represents the average throughput of user i and is defined as $x_i = 2^{U_i}$ for $i = 1, \ldots, N$ where $U_i = \log_2 \left(\sum_{t=1}^{K} \tau_{it} R_{it} \right)$. Note that FI takes the value of 1 when there is a complete fair allocation. As illustrated in Fig. 3 SG + TDMA is worse than BCD in terms of fairness. Especially for eight users, $FI_{SG+TDMA} = 0.41$ whereas $FI_{BCD} = 0.76$. Although low path losses embrace lower utility improvement, they mainly allow BCD algorithm to be very efficient in terms of fairness, e.g. above

0.75. However as the path loss difference between users increase, completely fair allocations may not be the optimal ones. In such a case, the algorithm should favor the strongest users more than it favors the weak users (as shown in Fig. 3), to maximize the utility function and form a proportionally (instead of purely) fair allocation.

5 Conclusion

In this paper, we have investigated the proportional fair power and time allocation problem on an energy harvesting downlink. In order to determine the optimal *off-line* schedule, we designed the problem as a proportional fairness based utility maximization problem. Then, by using the structural characteristics of the problem, i.e., biconvexity and multiple optima, we have shown an algorithm based on block coordinate descent (BCD) that surely converges to a partial optimal solution of the problem. Simulation results show that by allocating among users the transmission power and the proportion of the time between energy harvests BCD achieves a good balance between throughput and fairness. BCD algorithm converges to the partial optimal utility, and can improve the utility approximately 15 % when compared to SG + TDMA schedule.

In [13] we will investigate the power related characteristics of an optimal solution of the proposed problem, and, develop a low-complexity heuristic algorithm that will closely track the performance of the BCD algorithm.

References

1. Yang, J., Ulukus, S.: Transmission completion time minimization in an energy harvesting system. In: 44th Annual Conference on Information Sciences and Systems (CISS), pp. 1–6 (2010)
2. Antepli, M.A., Uysal-Biyikoglu, E., Erkal, H.: Optimal packet scheduling on an energy harvesting broadcast link. IEEE J. Sel. Areas Commun. **29**(8), 1712–1731 (2011)
3. Yang, J., Ozel, O., Ulukus, S.: Broadcasting with an energy harvesting rechargeable transmitter. IEEE Trans. Wirel. Commun. **11**(2), 571–583 (2012)
4. Tutuncuoglu, K., Yener, A.: Optimum transmission policies for battery limited energy harvesting systems. IEEE Trans. Wirel. Commun. (2011) (submitted)
5. Ozel, O., Tutuncuoglu, K., Yang, J., Ulukus, S., Yener, A.: Resource management for fading wireless channels with energy harvesting nodes. In: IEEE INFOCOM, pp. 456–460 (2011)
6. Ho, C., Zhang, R.: Optimal energy allocation for wireless communications powered by energy harvesters. In: IEEE ISIT, pp. 2368–2372 (2010)
7. Tekbiyik, N., Girici, T., Uysal-Biyikoglu, E., Leblebicioglu, K.: Proportional fair resource allocation on an energy harvesting downlink—Part I: structure, arXiv:1205.5147v1 [cs.NI], April 2012 (Submitted)
8. Mao, Z., Koksal, C.E., Shroff, N.B.: Resource allocation in sensor networks with renewable energy. In: IEEE ICCCN, pp. 1–6 (2010)
9. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming Theory and Algorithms. Wiley, New Jersey (2006)

10. Luenberger, D.G., Ye, Y.: Linear and Nonlinear Programming. Springer, New York (2008)
11. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Belmon (1999)
12. Gorlatova, M., Berstein, A., Zussman, G.: Performance evaluation of resource allocation policies for energy harvesting devices. In: IEEE Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), pp. 189–196 (2011)
13. Tekbiyik, N., Uysal-Biyikoglu, E., Girici, T., Leblebicioglu, K.: A practical algorithm for proportional-fair downlink scheduling in the presence of energy harvesting. (Accepted, ISCIS 2012)
14. Mahmoodi, T., Friderikos, V., Holland, O., Aghvami, H.: Balancing sum rate and TCP throughput in OFDMA based wireless networks. In: IEEE ICC, pp. 1–6 (2010)

An Algorithm for Proportional-Fair Downlink Scheduling in the Presence of Energy Harvesting

Neyre Tekbiyik, Elif Uysal-Biyikoglu, Tolga Girici and Kemal Leblebicioglu

Abstract This paper considers the allocation of time slots in a frame, as well as power and rate to multiple receivers on an energy harvesting downlink. Energy arrival times that will occur within the frame are known at the beginning of the frame. The goal is to solve an optimization problem designed to maximize a throughput-based utility function that provides proportional fairness among users. An optimal solution of the problem was obtained by using a Block Coordinate Descent based algorithm, (BCD), in earlier work. However, that solution has high complexity and is therefore not scalable to a large number of users or slots. This paper first establishes some structural characteristics of the optimal solution. Then, building on those, develops a simple and scalable, yet efficient heuristic, named ProNTO. Numerical and simulation results suggest that ProNTO can closely track the performance of BCD.

Keywords Broadcast channel · Energy harvesting · Offline algorithms · Proportional fairness · Time sharing

This work was supported by TUBITAK Grant 110E252.

N. Tekbiyik (✉) · E. Uysal-Biyikoglu · K. Leblebicioglu
Department of Electrical and Electronics Engineering, Middle East Technical University, 06800 Ankara, Turkey
e-mail: ntekbiyik@eee.metu.edu.tr

E. Uysal-Biyikoglu
e-mail: elif@eee.metu.edu.tr

K. Leblebicioglu
e-mail: lebleb@eee.metu.edu.tr

T. Girici
Department of Electrical and Electronics Engineering, TOBB Economics and Technology University, 06560 Ankara, Turkey
e-mail: tgirici@etu.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_10,

1 Introduction

With increasing awareness of the potential harmful effects to the environment caused by CO_2 emissions and the depletion of non-renewable energy sources, many researchers have focused on decreasing the carbon footprint of wireless communications and using energy-wise self-sufficient nodes that can harvest ambient energy. Although harvesting ambient energy allows sustainable and environmentally friendly deployment of wireless networks, harvested power is typically irregular and can at times fall short of typical power consumption levels in wireless nodes. Therefore, the harvested energy may need to be accumulated in storage devices (e.g., supercapacitors) to a sufficient level to operate the nodes. As the energy harvested from the environment is time-varying, utilizing harvested energy efficiently is an important criteria to match the performances of energy harvesting networks with their battery or grid-powered counterparts.

Recently, several studies have focused on optimizing data transmission with an energy harvesting transmitter [1–7]. A single-user communication system operating with an energy harvesting transmitter is considered in [1], where a packet scheduling scheme that minimizes the time by which all of the packets are delivered to the receiver is obtained. A multi-user extension of [1] has also been considered in [2, 3] and the same time minimization problem is solved for a two user broadcast channel. These approaches are extended in [4] and [5] to the case of a transmitter with a finite capacity battery.

In recent work [8], we proposed a proportional fairness based utility maximization problem in a time-sharing multi-user additive white Gaussian noise (AWGN) broadcast channel, where the transmitter is capable of energy harvesting. The goal is to achieve the optimum *off-line* schedule, by assuming that the energy arrival profile at the transmitter is deterministic and known ahead of time in an off-line manner for a time window, called frame, i.e., the energy harvesting times and the corresponding harvested energy amounts are known at the beginning of each frame. The treatment in [8] considered the general case in which the interarrival times between consecutive harvests do not have to be equal. Here, we focus on the case where energy interarrival times are equal. Not all generality is lost, because harvest amounts are arbitrary and the absence of a harvest in a certain slot can be expressed with a harvest of amount zero for the respective slot. Periodic sampling of harvests is also consistent with practice as in many energy harvesting systems, transmitters have supercapacitors that can store the harvested energy and supply in every predetermined time window, allowing the case of periodic energy arrivals.

In [8], we presented BCD algorithm that converges to a partial optimal solution of the problem presented in [8]. Although BCD is guaranteed to converge to a partial optimal solution, it is computationally expensive and when there are tens of users and energy arrivals, forming invertible hessian matrices (needed for the optimization of the power variables) may not be computationally possible. Hence in this paper, we first present the derived characteristics of the optimal solution [10] of the same

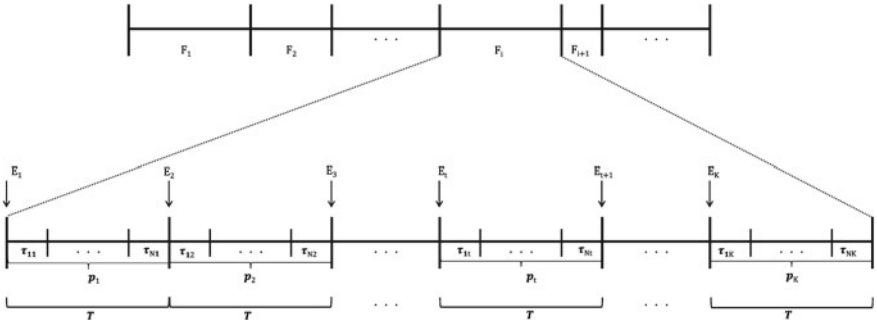


Fig. 1 Problem illustration: there are K energy arrivals in a frame, and, the time between consecutive arrivals are allocated to N users

problem and then, use these characteristics to develop a simple heuristic, ProNTO, which is resistent to increasing number of users and harvests, and able to closely track the performance of the BCD algorithm.

We start by describing the problem formulation in the next section. Next, we discuss the structure and properties of the optimal solution in Sect. 3. Depending on these properties, the ProNTO heuristic is proposed in Sect. 4. In Sect. 5, we present our numerical and simulation results. We conclude in Sect. 6 with an outline of future directions.

2 Problem Formulation

Consider a time-slotted system where each frame, of length F_i, is divided into K slots. There is a single transmitter that transmits to N users by time sharing. Channel conditions remain constant throughout the frame (g_n, the gain of user n, is chosen to be constant during F_i). Note that, we use the same system model as in [8]. Thus, we are interested in a specific frame and, we assume that some energy, E_t, is harvested from the environment at the beginning of each time slot t of the given frame. However, unlike [8], in this paper we assume periodic energy arrivals and hence equal slot lengths, as shown in Fig. 1.

For a given frame, the transmitter needs to choose a time allocation vector $\tau_t = (\tau_{1t}, \ldots, \tau_{Nt})$ and, a power level p_t for each time slot t of the frame, where τ_{nt} and $p_{nt} = p_t$ are the time allocated for transmission to user n and, the transmission power for user n, respectively, during slot t. While doing this, the transmitter needs to maximize a utility function, the log-sum of the user rates [8], which is known to achieve proportional fairness [9]. Thus, we define Problem 1:

Problem 1

$$\text{Maximize: } U(\overline{\tau}, \overline{p}) = \sum_{n=1}^{N} \log_2 \left(\sum_{t=1}^{K} \tau_{nt} W \log_2 \left(1 + \frac{g_n p_t}{N_o W} \right) \right)$$

$$\text{subject to: } \tau_{nt} \geq 0, \;\; p_t \geq 0 \tag{1}$$

$$\sum_{n=1}^{N} \tau_{nt} = T, \quad \sum_{t=1}^{K} \tau_{nt} \geq \epsilon \tag{2}$$

$$\sum_{i=1}^{t} p_i T_i \leq \sum_{i=1}^{t} E_i \tag{3}$$

where $t = 1, \ldots, K$ and $n = 1, \ldots, N$. W and N_o are the bandwidth for a single link channel and, the power spectral density of the background noise respectively. $\frac{g_n p_t}{N_o W}$ is the SNR of user n in slot t. (1) represents the nonnegativity constraints. (2) ensures that the total time allocated to users does not exceed T, and, every user gets some time ($\geq \epsilon$ where ϵ is an infinitely small number) during the frame. (3) ensures that no energy is transmitted before becoming available.

Previous analysis on structural characteristics [8] of Problem 1 revealed that it can be formulated as a biconvex optimization problem, and that it has multiple partial optima. However, the characteristics of the optimal solution have not been analyzed in depth. Therefore, the next section is dedicated to this analysis.

3 Structure and Properties of the Optimal Solution

In this section, we analyze the structural properties of an optimal power-time allocation. Note that, the utility function of Problem 1 can be rewritten as

$$U = \sum_{n=1}^{N} log_2(\overline{\tau_n}^T \overline{R_n}) = U_1 + U_2 + \cdots + U_N \tag{4}$$

where U_n, the utility of user n, is

$$U_n = log_2(\overline{\tau_n}^T \overline{R_n}) \tag{5}$$

where $\overline{R_n} = [R_{n1} \ldots R_{nK}]^T$, $\overline{\tau_n} = [\tau_{n1} \ldots \tau_{nK}]^T$, and $R_{nt} = W log_2 \left(1 + \frac{g_n p_t}{N_o W}\right)$.

3.1 Structure of an Optimal Power Allocation Policy

In this section, we assume that the time allocation is determined, and try to characterize the structure of the optimal power allocation policy. Clearly, when time variables are known constants, Problem 1 reduces to:

Problem 2

$$\text{Maximize: } U(\overline{p}) = \sum_{n=1}^{N} U_n(\overline{p})$$

$$\text{subject to: } p_t \geq 0 \tag{6}$$

$$\sum_{i=1}^{t} p_i T_i \leq \sum_{i=1}^{t} E_i \tag{7}$$

where $t = 1, \ldots, K$ and, U_n is a function of the power variables [as defined in (5)]. As Problem 2 is strictly convex [10], for every given time allocation there exists a unique optimal power allocation. The main problem, Problem 1, is known to be a biconvex optimization problem that has many partial optima [8]. In Theorem 1, we claim that one of these optima contain a nondecreasing power schedule.

Theorem 1 *In case of periodic energy arrivals, ($T_j = T$, for $\forall j \in \{1, \ldots, K\}$), there exists an optimal schedule $(\overline{\tau}^*, \overline{p}^*)$ such that $\overline{p}^*$ is nondecreasing, (e.g., $\overline{p}^* = (p_1, \ldots, p_K)$ where $p_1 \leq p_2 \leq \cdots \leq p_K$).*

Proof Due to space limitations, we refer the interested reader to [10], for the details of this proof.

In Lemma 1 of [10], we have proven that any feasible permutation[1] of the optimal schedule $(\overline{\tau}^*, \overline{p}^*)$, described in Theorem 1, is also optimal. We use this fact and the result of Theorem 1, to develop a close-to-optimal heuristic, ProNTO.

4 ProNTO Heuristic

In this section, we present ProNTO (Powers Nondecreasing—Time Ordered) which is a fast and simple heuristic developed based on the characteristics discovered in Sect. 3 and the simulation results obtained by running BCD algorithm for periodic energy arrivals. ProNTO operates as follows:

1. **For Power Allocation**: Assign nondecreasing powers through the slots by using the energy harvest statistics, as follows:

[1] A feasible permutation is any permutation of a given schedule that does not violate the constraints described in Eqs. (1)–(3).

(a) From a slot, say i, to the next one $i+1$: If harvested energy decreases, defer a Δ amount of energy from slot i to slot $i+1$ to equalize the power levels. Do this until all powers are nondecreasing, and, form a virtual nondecreasing harvest order.
(b) By using the virtual harvest order, assign nondecreasing powers through the slots, i.e., in each slot, spend what you virtually harvested at the beginning of that slot.

2. **For Time Allocation**: Order the users, $u_1, \ldots, u_N$, according to their channel quality and form a user priority vector, $\overline{u^{\downarrow}} = [u_1^{\downarrow}, \ldots, u_N^{\downarrow}]$ where $u_1^{\downarrow}$ represents the user with the best channel. As $K > N$, Allocate every user $\beta = \frac{K - mod(K,N)}{N}$ slots as follows: The first β slots are allocated to $u_1^{\downarrow}$, the next β slots are allocated to $u_2^{\downarrow}$, etc. Add the remaining $mod(K, N)$ slots to the most powerful $mod(K, N)$ users' slots. For example; Let $K = 10$ and $N = 3$, and the path losses of the users to be 11, 15, 8 dB respectively. Then, the first 4 slots are allocated to user 3, the next 3 slots are allocated to user 1, and the last 3 slots are allocated to user 2.

The time allocation method is based on the following observation; when a partial optimal solution obtained by BCD algorithm is modified to form the nondecreasing optimal schedule (the permutation that includes nondecreasing power allocation), the time allocation becomes ordered as described above [10].

5 Numerical and Simulation Results

In this section, we present the numerical and simulation results related to ProNTO heuristic. Throughout our simulations we use the following setup: $W = 1$ kHz, $N_o = 10^{-6}$ W/Hz. We assume that some amount of energy ($\epsilon < E < \infty$ where ϵ is an infinitely small value) is harvested every 10 s ($T = 10$), within a frame (period of known harvests). We first assume that there are four users in the system and the frame length is 100 s. The energy arrivals are $\overline{E} = [20, 100, 1, 1, 1, 70, 100, 1, 10, 40]$ J in the [1st, 2nd, ..., 10th] slots respectively. The first user is the strongest one, and, other users are ordered in a such way that the preceding user is twice as strong as the next one. Thus, the path losses of the users are chosen to be; 19, 22, 25, 28 dB respectively. The starting point of the BCD algorithm is the Spend What You Get (SG) policy [11] combined with TDMA time allocation. The results obtained both for BCD and ProNTO are presented in Table 1. In the table, power and time units are Watts and seconds, respectively. The first result-row of the table represents the original (optimal) results obtained by the BCD algorithm. The results in the second row however, are obtained by modifying the original results to obtain the nondecreasing optimal schedule mentioned in Theorem 1. The third result-row represents the results obtained by using ProNTO heuristic. As observed, ProNTO's utility improvement performance is very close to that of BCD's.

Table 1 The power and time allocation policies, and corresponding utility improvements found by BCD and ProNTO

Method	Time and Power Allocation											Utility	Utility Improvement
	Users/Slots	1	2	3	4	5	6	7	8	9	10		
BCD	1	10	8.5778	0	0	10	0	0	0	0	0	66.3608	7.6522 %
	2	0	1.4222	10	10	0	0	0	0	4.7599	0		
	3	0	0	0	0	0	10	0	0	5.2401	8.3789		
	4	0	0	0	0	0	0	10	10	0	1.6211		
		2	2.3810	2.8029	2.8029	2.3133	4.0502	5.0742	5.0742	3.6943	4.2070		
BCD (Ordered)	1	10	10	8.5777	0	0	0	0	0	0	0	66.3608	7.6522 %
	2	0	0	1.4223	10	10	4.7598	0	0	0	0		
	3	0	0	0	0	0	5.2401	10	8.3789	0	0		
	4	0	0	0	0	0	0	0	1.6211	10	10		
		2	2.3132	2.3810	2.8028	2.8028	3.6343	4.0501	4.2070	5.0742	5.0742		
ProNTO	1	10	10	10	0	0	0	0	0	0	0	66.3005	7.5544 %
	2	0	0	0	10	10	10	0	0	0	0		
	3	0	0	0	0	0	0	10	10	0	0		
	4	0	0	0	0	0	0	0	0	10	10		
		2	2.5750	2.5750	2.5750	2.5750	4.4200	4.4200	4.4200	4.4200	4.4200		

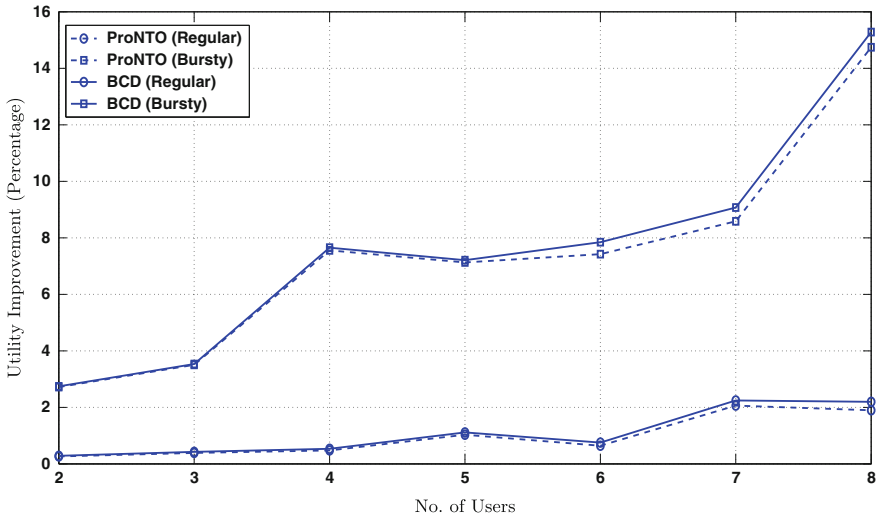


Fig. 2 The effect of different energy arrival series and increasing number of users on the utility improvement

Next, we analyze the effect of the nature of energy arrivals and increasing number of users, on the performance of the ProNTO heuristic. In order to do this, we need to define another energy arrival vector. We assume the new frame length is 120 s and the harvested energies are [73 65 9 19 40 37 22 84 39 67 81 100] J. We call this vector as *Regular* case and the previous one, [20, 100, 1, 1, 1, 70, 100, 1, 10, 40], *Bursty* case. The results are illustrated in Fig. 2.

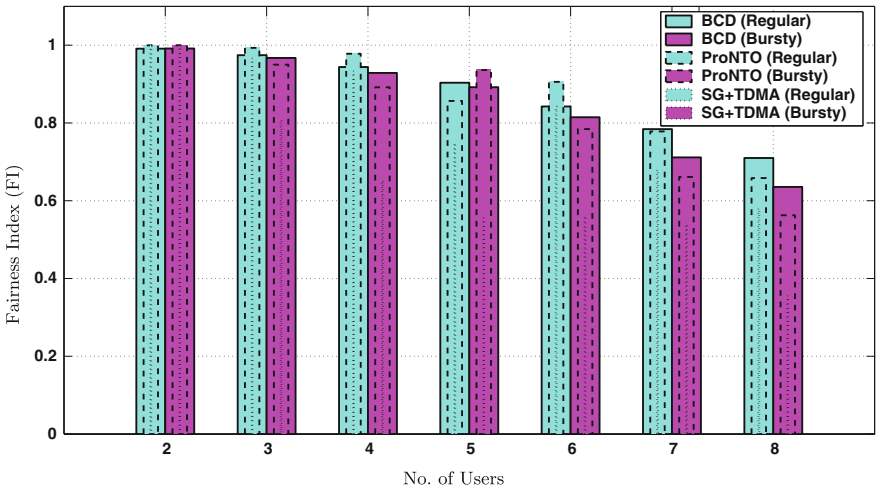


Fig. 3 Fairness index versus energy arrival series and number of users

As seen from the figure, ProNTO closely tracks the performance of the BCD algorithm. i.e., the utility improvement difference between two algorithms is less than 1 % at all instances (no matter how many users exist in the system). It is also observed that the sudden changes in energy arrivals and long term low energy arrivals increase the utility improvement.

We also provide the fairness analysis. In order to measure the fairness of our proposed heuristic, ProNTO, and compare it with the optimal one, BCD, we use the well-known Jain's fairness index [12]. The index FI takes the value of 1 when there is a complete fair allocation. For computing $FI = \frac{(\sum_{i=1}^{N} x_i)^2}{N \cdot \sum_{i=1}^{N} x_i^2}$, we use the no. of bits sent to the users, $x_i = 2^{U_i}$ for $i = 1, \ldots, N$, where U_i is as defined in Eq. (5). From Fig. 3, it is clear that when the number of users in the system increase, all schemes tend to be less fair. This is normal since the path loss differences between users increase. As illustrated, SG+TDMA is the worst choice, and ProNTO and BCD are competitive in terms of fairness.

6 Conclusion

In this paper, we have extended our previous work [8]. By deriving the optimal power allocation policy related characteristics of the problem proposed in [8], we developed ProNTO, i.e., a simple-but-efficient heuristic that can closely track the performance of the optimal BCD algorithm. ProNTO can operate in couple of seconds, whereas BCD operates in order of minutes. Simulation results show that the utility improvement difference between BCD and ProNTO is less than 1 %.

Interesting future directions include analyzing the time allocation related characteristics of the proposed problem, and combining these results with the power related characteristics obtained in this paper, to improve the ProNTO heuristic. Another direction may be the development of an online algorithm that will bypass the need for offline knowledge about the energy arrival profile.

References

1. Yang, J., Ulukus, S.: Transmission completion time minimization in an energy harvesting system. In: Conference on Information Sciences and Systems (CISS), pp. 1–6. (2010)
2. Yang, J., Ozel, O., Ulukus, S.: Broadcasting with an energy harvesting rechargeable transmitter. IEEE Trans. Wire. Commun. **11**(2), 571–583 (2012)
3. Antepli, M.A., Uysal-Biyikoglu, E., Erkal, H.: Optimal packet scheduling on an energy harvesting broadcast link. IEEE J. Sel. Areas Commun. **29**(8), 1712–1731 (2011)
4. Ozel, O., Yang, J., Ulukus, S.: Broadcasting with a battery limited energy harvesting rechargeable transmitter. In: IEEE WiOpt, pp. 205–212. (2011)
5. Tutuncuoglu, K., Yener, A.: Optimum transmission policies for battery limited energy harvesting nodes. IEEE Trans. Wire. Commun. **11**(3), 1180–1189 (2012)
6. Ozel, O., Tutuncuoglu, K., Yang, J., Ulukus, S., Yener, A.: Resource management for fading wireless channels with energy harvesting nodes. In: IEEE INFOCOM, pp. 456–460. (2011)
7. Ho, C., Zhang, R.: Optimal energy allocation for wireless communications powered by energy harvesters. In: IEEE ISIT, pp. 2368–2372. (2010)
8. Tekbiyik, N., Uysal-Biyikoglu, E., Girici, T., Leblebicioglu, K.: Utility-based time and power allocation on an energy harvesting downlink: the optimal solution. ISCIS (2012) (accepted)
9. Mao, Z., Koksal, C.E., Shroff, N.B.: Resource allocation in sensor networks with renewable energy. In: IEEE ICCCN, pp. 1–6. (2010)
10. Tekbiyik, N., Girici, T., Uysal-Biyikoglu, E., Leblebicioglu, K.: Proportional fair resource allocation on an energy harvesting downlink—part II: algorithms, arXiv:1205.5153v1 [cs.NI], April 2012. (submitted)
11. Gorlatova, M., Berstein, A., Zussman, G.: Performance evaluation of resource allocation policies for energy harvesting devices. In: IEEE Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), pp. 189–196. (2011)
12. Mahmoodi, T., Friderikos, V., Holland, O., Aghvami, H.: Balancing sum rate and TCP throughput in OFDMA based wireless networks. In: IEEE ICC, pp. 1–6. (2010)

Part III
Performance Modelling and Evaluation

Compositional Verification of Untimed Properties for a Class of Stochastic Automata Networks

Nihal Pekergin and Minh-Anh Tran

Abstract We consider Stochastic Automata Networks whose transition rates depend on the whole system state but are not synchronised and are restricted to satisfy a property called *inner proportional*.We prove that this class of SANs has both product form steady-state distribution and product form probability over untimed paths. This product form result is then applied to check formulae that are equivalent to some special structure that we call *path-product* of sets of untimed paths. In particular, we show that product form solutions can be used to check unbounded Until formulae of the Continuous Stochastic Logic.

1 Introduction

Probabilistic model checking is an extension of the formal verification methods for systems exhibiting stochastic behaviour. The system model is usually specified as a state transition system, with probabilities attached to transitions, for example, Markov chains. A wide range of quantitative performance, reliability, and dependability measures can be specified using temporal logics such as Continuous Stochastic Logic (CSL) defined over Continuous Time Markov Chains (CTMC) [1] and Probabilistic Computational Tree Logic (PCTL) defined over Discrete Time Markov Chains (DTMC) [8]. To perform model checking by numerical analysis we need to compute transient-state or steady-state distribution of the underlying CTMC. The numerical model checking has been studied extensively and numerous algorithms [2] have

N. Pekergin (✉) · M.-A. Tran
LACL, University of Paris-Est Créteil Val de Marne, 61 avenue Général de Gaulle, 94010 Créteil, France
e-mail: nihal.pekergin@u-pec.fr

M.-A. Tran
e-mail: minh-anh.tran@u-pec.fr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_11,

been devised and implemented in different model checkers. Despite the considerable works in the domain, the numerical Markovian analysis still remains a problem.

Different approaches have been applied to overcome the state space explosion problem. Data structures which lead to compact representations of large models such as Binary Decision Diagrams (BDD) and Multi Terminal Binary Decision Diagrams (MTBDD) with efficient manipulation algorithms have been applied to consider large models. This approach is called symbolic model checking [3]. Another approach is based on the state space reduction techniques. The idea here is to have a representation of the underlying model in a reduced-size model, which is called abstraction in model checking [10]. The large models can also be analysed by decomposition which means that sub-systems are analysed in isolation and then the global behaviour is deduced from these solutions. This is called as compositional model checking [4, 6].

The goal of this paper is to present a model checking approach which is able to take advantage of product form solutions. It is worth pointing out that product form solutions play an important role in calculating stationary distributions of Markov chains in performance evaluation [1], but, on the contrary, are believed to have no significant use in model checking. In this paper, we study a subclass of Stochastic Automata Networks (SANs) without synchronisations, which have product form steady-state distributions [7]. In the subclass, there is no synchronisation, all transition rates are functional and restricted to satisfy a property that we call *inner proportional*. This class remains large enough, for example, to generalise competing Markov chains [5]. We profit from product form solutions of this class to perform the CSL model checking for the untimed Until path and the steady-state formulae.

The rest of the paper is organised as follows: Sect. 2 gives a brief introduction for CSL and then introduces the class of SANs for which the compositional model checking is performed. Section 3 proves the product form solution for the steady-state and Sect. 4 provides the product form over untimed paths.

2 Framework and Model

CSL Model checking: In this paper, we consider the steady-state and untimed Until formulae of CSL model checking. We briefly give the syntax and semantic for these operators and we refer to [1, 2] for further information. Model $\mathcal{M}$ is a time-homogeneous CTMC with infinitesimal generator Q taking values in a set of states S. AP denotes a finite set of atomic propositions, and $L : S \to 2^{AP}$ is the labelling function which assigns to each state $s \in S$ the set $L(s)$ of atomic propositions $a \in AP$ those are valid in s. A path through $\mathcal{M}$ can be finite or infinite. A finite path σ of length n is a sequence of states: $\sigma = s^0, s^1, \ldots, s^n$ with transition rates $Q(s^i, s^{i+1}) > 0$. We denote by $paths_s$ the set of all paths starting from state s. Let p be a probability threshold and $\triangleleft$ be an arbitrary operator in the set $\{\leq, \geq, <, >\}$.The syntax of CSL is defined by :

$$\phi ::= true \mid a \mid \phi \wedge \phi \mid \neg\phi \mid \mathcal{P}_{\triangleleft p}(\phi \, \mathcal{U} \, \phi) \mid \mathcal{S}_{\triangleleft p}(\phi)$$

The expression $\mathcal{P}_{\triangleleft p}(\phi_1 \mathcal{U} \phi_2)$ asserts that the probability measure of paths satisfying $\phi_1 \mathcal{U} \phi_2$ meets the bound given by $\triangleleft p$. The path formula $\phi_1 \mathcal{U} \phi_2$ asserts that ϕ_2 will be satisfied at some time $t \in [0, \infty)$ and that at all preceding time ϕ_1 holds. $\mathcal{S}_{\triangleleft p}(\phi)$ asserts that the steady-state probability for ϕ-states meets the bound $\triangleleft p$. We present briefly the semantics of these formula where $\models$ is the satisfaction operator:

$s \models true$	for all $s \in S$
$s \models a$	iff $a \in L(s)$
$s \models \neg\phi$	iff $s \not\models \phi$
$s \models \mathcal{P}_{\triangleleft p}(\phi_1 \mathcal{U} \phi_2)$	iff $\mathbb{P}^{\mathcal{M}}(s, \phi_1 \mathcal{U} \phi_2) \triangleleft p$
$s \models \mathcal{S}_{\triangleleft p}(\phi)$	iff $\sum_{s \mid s \models \phi} \pi^{\mathcal{M}}(s) \triangleleft p$

where $\mathbb{P}^{\mathcal{M}}(s, \phi_1 \mathcal{U} \phi_2)$ denotes the probability measure of all paths σ starting from s ($\sigma \in paths_s$) satisfying $\phi_1 \mathcal{U} \phi_2$, i.e., $\mathbb{P}^{\mathcal{M}}(s, \phi_1 \mathcal{U} \phi_2) = \mathbb{P}\{\sigma \in paths_s \mid \sigma \models \phi_1 \mathcal{U} \phi_2\}$; $\pi^{\mathcal{M}}(s)$ denotes the steady-state probability of state s of the chain $\mathcal{M}$. In the case $\mathcal{M}$ is ergodic, the steady-state distribution is independent of the initial state, then the steady-state formula is satisfied or not whatever the initial state.

SANs with local, functional and inner proportional transitions: Consider a network of N interacting stochastic automata $A_1, A_2, \ldots A_N$ where

- Transitions are *local*, i.e., not synchronised: it is forbidden to have two events occurring at the same time in two different automata.
- Transition rates of each automaton depend on the state of the whole system. Such transitions are also called *functional* transitions.
- For any automaton, transition rates are restricted to be *inner proportional*: they may depend on the state of all other automata; however, the proportion between two arbitrary transition rates of this automaton remains independent from the state of other automata.

In this work, we note

- $s = (s_1, s_2, \ldots, s_N)$ the state vector, $s_{-k} = (s_1, \ldots, s_{k-1}, s_{k+1}, \ldots, s_N)$ the state vector without component s_k.
- $\mathcal{S}_k$ the set of states of automaton A_k, $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_N$ the set of all system states, and $\mathcal{S}_{-k} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_{k-1} \times \mathcal{S}_{k+1} \times \cdots \times \mathcal{S}_N$ the set of states of all automata other than A_k.
- $Q_k^s : \mathcal{S}_k \times \mathcal{S}_k \to \mathbb{R}$ the infinitesimal generator of A_k when the system is in state s. More precisely, transition rates of A_k are functions of the state vector s_{-k}.

In state s, the total outgoing rate from automaton A_k is $-Q_k^s(s_k, s_k) = \sum_{s_k' \neq s_k} Q_k^s(s_k, s_k')$.

Property 1 ***(Characterisation of inner proportional transitions)*** *The transition rates of A_k are inner proportional if and only if there exists a state-dependent factor $\alpha_k : \mathcal{S}_{-k} \to \mathbb{R}$ and an infinitesimal generator $Q_k : \mathcal{S}_k \times \mathcal{S}_k \to \mathbb{R}$ which does not depend on the vector s_{-k} such that*

$$Q_k^s(s_k, s_k') = \alpha_k(s_{-k}) Q_k(s_k, s_k') \qquad \forall s_k, s_{k'} \in \mathcal{S}_k, \forall s_{-k} \in \mathcal{S}_{-k}. \tag{1}$$

Matrix Q_k is the infinitesimal generator of the *representative automaton* of A_k (or A_k in isolation). In isolation, the total outgoing rate from state s_k is given by $-Q_k(s_k, s_k) = \sum_{s'_k \neq s_k} Q_k(s_k, s'_k)$.

3 Product Form Solution for the Steady-State

First of all, SANs with local, functional and inner proportional transitions form a subclass of SANs without synchronisations considered in [7]. In this work, Fourneau et al. considered SANs where transitions are local and functional, but not necessarily inner proportional. They denote by F_k the set of infinitesimal generators of A_k. This notation F_k denotes the set $\{Q_k^s : s \in \mathcal{S}\}$ in our model. Theorem 6 of [7] states that a SAN with local and functional transitions has a product form steady-state distribution if for any automaton A_k there exists a probability distribution π_k that verifies the following equation

$$\pi_k Q = 0 \qquad \forall Q \in F_k. \tag{2}$$

In the view of Property 1, for inner proportional transitions, F_k is given by

$$F_k = \{\alpha_k(s_{-k}) Q_k : s_{-k} \in \mathcal{S}_{-k}\},$$

where α_k is a real-valued function of s_{-k} and Q_k is the representative infinitesimal generator of automaton A_k. Thus, a distribution π_k satisfies Eq. (2) for all infinitesimal generators of A_k if it satisfies the following Eq. (3) for only Q_k.

Theorem 1 *If for each automaton A_k in isolation there exists a probability distribution π_k such that*

$$\pi_k Q_k = 0, \tag{3}$$

then the steady-state distribution of the system has the following product form

$$\pi(s) = C \prod_{k=1}^{N} \pi_k(s_k). \tag{4}$$

This product-form solution can be used to check the steady-state formula $\mathcal{S}_{\lhd p}(\phi)$ to see if the sum of steady-state probabilities of states satisfying ϕ meets the bound p or not. In the following, two applications of this product form result will be illustrated.

Example 1 Generalised competing Markov chains. We extend the system of competition between concurrent processes over a number of shared resources [4, 5]. The extension is that common resources are no longer limited to be mutually exclusive and strong blocking but may be used by different components at the same time. In other words, transition rates may not be switched off to zero but are only reduced

by some factor when common resources are shared. Transition rates are local, functional and inner proportional: when a component shares some common resources with others, its transition rates are reduced by some factor α, which might be a function of the states of all other resources. More precisely, the transition rate matrix of component k is of the form $Q_k^s = \alpha_k(s_{-k})Q_k$. Thus, Theorem 1 applies to this system of generalised competing Markov chains.

Example 2 Multiclass queue with proportional state-dependent rates In this example, Theorem 1 is applied to a multiclass queue with state-dependent arrival rates and service rates. Consider a queue of N classes of customers where customers of each class arrive according to a variable-rate Poisson process. Let x_k be the number of class-k customers, $x = (x_1 \dots x_N)$ be the system state. For class k, suppose that service requirements follow an exponential distribution of parameter μ_k and the arrival rate $\lambda_k(x_{-k})$ is a general function of the vector x_{-k} composed of numbers of customers of other classes. Besides, suppose that the service effort $\Phi_k(x_{-k})$ allocated to class k is also a function of x_{-k}. Thus, class k is characterised by state-dependent arrival rate $\lambda_k(x_{-k})$ and departure rate $\mu_k\Phi_k(x_{-k})$.

SAN representation. Let us describe each class by an automaton. First, transition rates of each automaton are arrival rate and departure rate of the corresponding class. Therefore, these transition rates are functional. Second, if two events of two different classes are not allowed to occur at the same time, transition rates are local. Finally, if the ratio between arrival rate $\lambda_k(x_{-k})$ and departure rate $\mu_k\Phi_k(x_{-k})$ is equal to a constant λ_k/μ_k for any class k, transition rates are inner proportional. The system is a SAN with local, functional and inner proportional transitions. Thus, Theorem 1 applies and gives us an example of state-dependent multiclass queue with product form solutions w.r.t. classes.

4 Product Form Solution for Untimed Paths

In this section, we refer to a transition as a *k-move* if it corresponds to an event of automaton A_k, and we consider an arbitrary starting state $s = (s_1 \dots s_k \dots s_N)$. Product form solution for untimed paths is based on the following key result which is a direct consequence of the inner proportional characterisation (Property 1).

Property 2 *Conditioning on the event E_k^j that the first k-move happens at jth transition, the probability of the event $Obs(s_k, s_k')$ of observing the move (s_k, s_k') at this first k-move depends neither on the index j nor on the state of other automata:*

$$\mathbb{P}\left(Obs(s_k, s_k') \mid E_k^j\right) = \frac{Q_k(s_k, s_k')}{-Q_k(s_k, s_k)} \qquad \forall k, j. \tag{5}$$

Assumption In the rest of the work, we suppose that automaton A_k will make a move in the future with probability one if the total outgoing rate of its representative

automaton is strictly positive. This assumption states that the system will make a *k-move* and the first *k-move* will happen at jth transition with some finite index j.

Property 3 *Conditioning that the total outgoing rate of the representative automaton of A_k is strictly positive, automaton A_k will make a move in the future and the probability of observing (s_k, s'_k) at the first k-move does not depend on the state of other automata and is given by:*

$$\mathbb{P}\left(Obs(s_k, s'_k) \mid -Q_k(s_k, s_k) > 0\right) = \frac{Q_k(s_k, s'_k)}{-Q_k(s_k, s_k)} \quad \forall k. \tag{6}$$

In this part, we are interested in checking if state s satisfies an untimed formula ϕ. If the formula corresponds to a set of untimed paths, the work consists in calculating probabilities conditioned on the set of untimed paths starting with s. Let us denote this set by $U^{(s)}$. Besides, for the representative automaton of A_k, let $U_k^{(s_k)}$ be the set of untimed paths starting with s_k.

Definition 1 For any untimed path $\sigma = (s^0, s^1, s^2, \ldots)$, its k-projection is

$$proj_k(\sigma) = (s_k^{i_0}, s_k^{i_1}, s_k^{i_2}, \ldots), \quad 0 = i_0 < i_1 < i_2 < \cdots$$

such that any two consecutive system states s^j, s^{j+1} whose kth components are the same, i.e., $s_k^j = s_k^{j+1}$, are projected into a unique state s_k^j.

Thus, a k-projection of a path is defined such that repeated states are deleted.Consider an arbitrary starting state $s = (s_1, s_2, \ldots, s_N)$ and a finite untimed path $\sigma_k = (s_k, s_k^1, \ldots, s_k^l)$ of A_k in isolation. We say that the k-projection of an untimed path σ starts with σ_k if

$$proj_k(\sigma) = (s_k, s_k^1, \ldots, s_k^l, \ldots).$$

In the rest of the paper, the notation $proj_k(\sigma) = \sigma_k$ indicates that the k-projection of σ starts with σ_k. For example, consider $\sigma_k = (s_k, s'_k)$ of length 1. Property 3 gives the probability that the k-projection of σ starts with (s_k, s'_k), i.e., automaton A_k will make a move and the first *k-move* corresponds to (s_k, s'_k).

Theorem 2 *Conditioned on starting states s and s_k respectively, the probability of observing an untimed path σ whose k-projection starts with σ_k is equal to the probability of observing σ_k in the representative automaton of A_k*:

$$\mathbb{P}\left(\sigma : proj_k(\sigma) = \sigma_k \mid U^{(s)}\right) = \mathbb{P}\left(\sigma_k \mid U_k^{(s_k)}\right). \tag{7}$$

In the following we shall consider sets of untimed paths. We first introduce the notion of *path-product* over these sets.

Definition 2 For all $k = 1 \ldots N$, let U_k be a set of untimed paths σ_k in $\mathcal{S}_k$. The path-product of the sets $U_1 \ldots U_N$ is defined by the set of untimed paths σ in $\mathcal{S}$,

$$U \equiv \{\sigma : proj_k(\sigma) = \sigma_k, \sigma_k \in U_k, k = 1 \dots N\}, \text{ we note } U = \odot U_k.$$

For example, the set of untimed paths starting with state s is the path-product of the sets of untimed paths starting with state s_k of automaton A_k for all $k = 1 \dots N$, that is, $U^{(s)} = \odot U_k^{(s_k)}$.

Theorem 3 *Let s be an arbitrary starting state, U_k be a set of finite untimed paths starting with s_k in $\mathcal{S}_k$ for any automaton A_k, and U be the path-product $\odot U_k$. We have the following product form*

$$\mathbb{P}\left(\sigma \in U \mid U^{(s)}\right) = \prod_{k=1}^{N} \mathbb{P}\left(\sigma_k \in U_k \mid U_k^{(s_k)}\right). \tag{8}$$

Theorem 3 is important as it gives us a compositional method to check any formulae that is equivalent to a path-product of sets of single component untimed paths. In particular, we shall consider global unbounded Until formulae in the sequel.

Single component unbounded Until formulae: One consequence of Theorem 2 is the following result which provides a compositional method to check any single component untimed formula ω_k.

Theorem 4 *For any system state $s = (s_1, \dots, s_k, \dots, s_N)$ and for any single component untimed formula ω_k, the satisfaction of ω_k by the whole system is equivalent to its satisfaction by component k:* $s \models \omega_k \Longleftrightarrow s_k \models \omega_k$.

Thus, one may simply check if state s_k verifies formula ω_k for automaton A_k in isolation instead of working with global state s. For example, one may remove all functional interactions and only needs to pay attention to the corresponding isolated chain (or isolated class) in the model of generalised competing Markov chains (or multiclass queue with proportional state-dependent rates respectively).

Global unbounded Until formulae: Let $U^{(\phi \mathcal{U} \psi)}$ be the set of all untimed paths σ that satisfy the Until formula $(\phi \,\mathcal{U}\, \psi)$. The probability that s satisfies $(\phi \,\mathcal{U}\, \psi)$ is the following probability:

$$\mathbb{P}\left(\sigma \in U^{(\phi \mathcal{U} \psi)} \mid U^{(s)}\right) = \mathbb{P}\left(\sigma \in U^{(s)} \cap U^{(\phi \mathcal{U} \psi)} \mid U^{(s)}\right).$$

Theorem 5 *If $U^{(s)} \cap U^{(\phi \mathcal{U} \psi)}$ is a path-product of the form*

$$U^{(s)} \cap U^{(\phi \mathcal{U} \psi)} = \odot U_k \tag{9}$$

where U_k is some set of finite untimed paths σ_k of automaton A_k in isolation for all k, the probability that s satisfies the formula $\phi \,\mathcal{U}\, \psi$ has the following product form

$$\mathbb{P}\left(s \models \phi\, \mathcal{U}\, \psi\right) = \prod_{k=1}^{N} \mathbb{P}\left(\sigma_k \in U_k | U_k^{(s_k)}\right). \tag{10}$$

This is a direct consequence of Theorem 3 applied to the set $U = U^{(s)} \cap U^{(\phi \mathcal{U} \psi)}$. The idea of this result is to decompose the Until formula probability into separated components. However condition (9) seems to be sophisticated. Let us illustrate it by considering a concrete Until formula in the following.

Application of the compositional approach: Consider the multiclass queue described in Example 2 with batch Poisson arrivals [9]: For each class, arrivals of batches follow a Poisson process, where the batch size is a positive integer random variable. The Poisson parameter and the random variable for the batch size are functional, i.e., may depend on the state of all other classes.Suppose that arrival rates, batch sizes, departure rates depend on the system state such that inner proportional property holds : $Q_k^s = \alpha_k(s_{-k}) Q_k \quad \forall k$. With this extension, the SAN remains local, functional and inner proportional.

In a multiclass queue, we are often interested in the number of customers of each class. The logic formulae are to compare this number of customers to a threshold or a composition of these formulae. For each class k, let M_k be a threshold. We have a failure (overload) for class k if its number of customers reaches M_k. Whenever this happens the class stays at state M_k forever by convention, that is, $Q_k(M_k, s_k) = 0 \ \forall s_k \neq M_k$. On the contrary, the system functions properly if there exists a class k such as its number of customers does not exceed a threshold m_k. We are interested in verifying the Until formula $(\phi\, \mathcal{U}\, \psi)$ where

$$\begin{cases} \phi = \phi_1 \vee \cdots \vee \phi_N, \ \phi_k = \{x_k \leq m_k\} \\ \psi = \psi_1 \wedge \cdots \wedge \psi_N, \ \psi_k = \{x_k \geq M_k\}. \end{cases} \tag{11}$$

Condition ψ means failure of all classes, on the contrary, condition ϕ means that the system functions with at least one class. Let us remark that this Until formula is different from the steady-state probability of being in ψ-states, we consider indeed the probability to reach ψ-states passing through ϕ-states.

Consider the probability $\mathbb{P}(s \models \phi\, \mathcal{U}\, \psi)$ for an arbitrary state s. In order to use the above compositional approach, we shall determine the corresponding sets $U^{(s)}$, $U^{(\phi \mathcal{U} \psi)}$ and their intersection. First of all, $U^{(s)}$ is composed of untimed paths that begin with s. This set is simply the path-product of sets $U_k^{(s_k)}$ of untimed paths that begin with s_k for each component k, i.e., $U^{(s)} = \odot U_k^{(s_k)}$. Second, $U^{(\phi \mathcal{U} \psi)}$ is composed of finite untimed paths that satisfy the Until formula $(\phi\, \mathcal{U}\, \psi)$. Lastly, the intersection of the two sets is given by the following set of finite untimed paths $U = \{\sigma : \sigma \ starts \ with \ s, \sigma \models \phi\, \mathcal{U}\, \psi\}$. Replacing $(\phi\, \mathcal{U}\, \psi)$ by its definition described by Eq. (11), we obtain $U = \{\sigma : proj_k(\sigma) = \sigma_k, \sigma_k \ starts \ with \ s_k, \ \sigma_k \models \phi_k\, \mathcal{U}\, \psi_k, k = 1 \ldots N\} = \odot U_k$, where $U_k = \{\sigma_k : \sigma_k \ starts \ with \ s_k, \sigma_k \models \phi_k\, \mathcal{U}\, \psi_k\}$. As a result, Theorem 5 can be applied and the probability that s satisfies $(\phi\, \mathcal{U}\, \psi)$ is given by

$$\prod_{k=1}^{N} \mathbb{P}\left(\sigma_k \in U_k \mid U_k^{(s_k)}\right) = \prod_{k=1}^{N} \mathbb{P}\left(s_k \models \phi_k \,\mathcal{U}\, \psi_k\right).$$

In this example, the global Until formula can be decomposed into single component Until formulae. Instead of calculating the probability that some starting state satisfies a global Until formula, one only needs to calculate the product of corresponding single component probabilities.

5 Conclusion

In this paper we prove the product form solutions for the steady-state distribution of a class of SANs which generalises competing Markov chains. We perform the verification of the untimed Until and the steady-state formulae for this class of models through the product form solutions. In the last years, the common points for the performance evaluation and the quantitative model checking have been emphasised by many authors. Product form solutions have been largely used in performance evaluation and we think that it would be interesting to look for classes of models that can be efficiently model checked by means of product form solutions.

Acknowledgments The authors thank to Jean-Michel Fourneau for the fruitful discussions on product form solutions of SANs.

References

1. Aziz, A., Sanwal, K., Singhal, V., Brayton, R.: Model-checking continuous time Markov chains. ACM Trans. Comput. Logic **1**(1), 162–170 (2000)
2. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.-P.: Model-checking algorithms for continuous-time Markov chains. IEEE Trans. Softw. Eng. **29**(6), 524–541 (2003)
3. Baier, C, Katoen, J.-P., Hermanns, H.: Approximate symbolic model checking of continuous-time markov chains. In: CONCUR 99, LNCS 1664, pp. 146–161 (1999)
4. Ballarini, P., Horváth, A.: Compositional model checking of product-form CTMCs. Electron. Notes Theor. Comput. Sci. **250**, 21–37 (2009)
5. Boucherie, R.J.: A characterization of independence for competing Markov chains with applications to stochastic Petri nets. IEEE Trans. Softw. Eng. **20**, 536–544 (1994)
6. Buchholz, P., Katoen, J.-P., Kemper, P., Tepper, C.: Model-checking large structured Markov chains. J. Log. Algebraic Progr. **56**(1–2), 69–97 (2003)
7. Fourneau, J.M., Plateau, B., Stewart, W.J.: An algebraic condition for product form in stochastic automata networks without synchronizations. Perform. Eval. **65**, 854–868 (2008)
8. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects Comput. **6**(5), 512–535 (1994)
9. Kleinrock, L.:. Queueing Systems, volume I: Theory. Wiley Interscience, New York (1975)

10. Mamoun, M.B., Pekergin, N., Younès, S.: Model checking of continuous-time markov chains by closed-form bounding distributions. In: Third International Conference on the Quantitative Evaluation of Systems, pp. 189–198 (2006)

Computing Entry-Wise Bounds of the Steady-State Distribution of a Set of Markov Chains

F. Ait Salaht, J. M. Fourneau and N. Pekergin

Abstract We present two algorithms to find the component-wise upper and lower bounds of the steady-state distribution of an ergodic Markov chain. whose transition matrix **M** is entry-wise larger than matrix **L**. The algorithms are faster than Muntz's approach. They are based on the polyhedral theory developed by Courtois and Semal and on a new iterative algorithm which gives bounds of the steady-state distribution at each iteration.

1 Introduction

The basic theory of finite Markov chains and the algorithms are now a well-known technique for the study of dynamical systems. However finding the exact transition probability to describe a Markov chain is still a difficult problem in many engineering problems. Quite often, we only know that the transition probabilities belong to an interval. This is equivalent to state that the chain **M** is inside a set of chains described by an entry-wise lower bounding matrix **L** and an entry-wise upper bounding matrix **U**. We assume that the matrix is irreducible and aperiodic such that the steady-state distribution exists. A natural question is to find upper and lower bounds for the steady-state distribution for all the matrices in the set. More precisely we define $\mathcal{S}$ the set of irreducible and aperiodic stochastic matrices **M** such that $\mathbf{L} \preceq \mathbf{M}$ where $\preceq$ denotes the element-wise comparison of matrices and vectors. And we want to compute two non negative vectors l and u such that for all **P** in $\mathcal{S}$, we have $l \preceq \pi_P \preceq u$, where

F. A. Salaht · J. M. Fourneau (✉)
PRiSM, Université de Versailles-Saint-Quentin, CNRS UMR,
8144 Versailles, France
e-mail: jmf@prism.uvsq.fr

N. Pekergin
LACL, Université Paris Est, Créteil, France

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_12,

π_P is the steady-state distribution of **P**. The first approach to compute bounds on the steady-state distribution is based on the polyhedral theory developed by Courtois and Semal [4] and applied by Muntz [7]. However this approach is based on the resolution of n Markov chains where n is the size of the state space and we must perform the numerical computation with an accurate algorithm. If we use GTH [8] for the resolution of each chain, we obtain a total complexity of n^4 as GTH is a variant of the Gaussian elimination and it has a cubic complexity. Buchholz has proposed in [1, 2] two other approaches using both matrices **L** and **U** to define the set of matrices. Both algorithms require much more computation steps than the algorithm we propose here. As they used both matrices **L** and **U**, the bounds are more precise. But the first algorithm is reported by the author in [2] to be numerically unstable and the analysis of chains with more than 50 nodes is not feasible. Here we propose a new method. We still use the polyhedral theory and the same arguments. But we use a new numerical technique [3] to solve the steady-state distribution of each matrix considered in the polyhedral approach. This algorithm provides at each iteration upper and lower bounds of the steady-state distribution. Therefore we obtain bounds at the first iteration and at each iteration the bounds are improved. Thus our algorithm is much faster and if we iterate until convergence we obtain the same bounds as the ones obtained by Muntz. It is also numerically stable as it is only based on the product of non negative vectors. In the following of the paper, we describe in the next section the $I\nabla L$ and $I\nabla U$ Algorithms as they have been presented in [3]. Then, we present how we can combine results by Courtois and Semal for the polyhedral approach with the former algorithms to derive bounds for the steady-state distribution for a set of Markov chains. We also present the complexity of the approach and we illustrate the approach with some numerical results.

2 Algorithms Based on Monotone Sequences

We use the following notations. All vectors are row vectors, **e** is a vector with all component equal to 1, $\mathbf{e}_i$ is a row vector with 1 in position i and 0 elsewhere and the vector with all entries equal to 0 is denoted by **0**. We will denote by $\preceq$ the element-wise comparison of two vectors (or matrices). Finally, x^t denotes the transposed vector x, and $||x||$ is the sum of the elements of vector x. Let **P** be a finite stochastic matrix. We assume that **P** is ergodic. We first introduce some quantities easily computed from **P**.

Definition 1 Set $\nabla_P[j] = \mathbf{min}_i \mathbf{P}[i, j]$ and $\Delta_P[j] = \mathbf{max}_i \mathbf{P}[i, j]$. Remark that ∇_P may equal to vector **0** but Δ_P is positive as the chain is irreducible.

Bušić and Fourneau [3] proposed two iterative algorithms based on simple (**max**, +) (resp. (**min**,+)) properties, called $I\nabla L$ (resp. $I\nabla U$) which provide at each iteration a new lower (resp. upper) bound $x^{(k)}$ (resp. $y^{(k)}$) of the steady-state distribution of **P**.

Algorithm 1 Algorithm Iterate on ∇ Lower Bound (I∇L)

Require: $a \preceq \pi$, $b \preceq \nabla_{\mathbf{P}}$ and $b \neq \mathbf{0}$.
Ensure: Successive values of $x^{(k)}$.

1: $x^{(0)} = a$.
2: **repeat**
3: $\quad x^{(k+1)} = \mathbf{max}\left\{x^{(k)}, x^{(k)}\mathbf{P} + b(1 - ||x^{(k)}||)\right\}$.
4: **until** $1 - ||x^{(k)}|| < \epsilon$.

One can check that the conditions on the initialisation part of the algorithms require that $||\nabla_{\mathbf{P}}|| > 0$. We can use $a = \mathbf{0}$, $b = \nabla_{\mathbf{P}}$ and $c = \Delta_{\mathbf{P}}$ in the initialization steps of the algorithms. These algorithms have the following very important properties, which are all proved in [3]:

Algorithm 2 Algorithm Iterate on ∇ Upper Bound (I∇U)

Require: $c \succeq \pi$, $b \preceq \nabla_{\mathbf{P}}$ and $b \neq \mathbf{0}$.
Ensure: Successive values of $y^{(k)}$.

1: $y^{(0)} = c$.
2: **repeat**
3: $\quad y^{(k+1)} = \mathbf{min}\left\{y^{(k)}, y^{(k)}\mathbf{P} + b(1 - ||y^{(k)}||)\right\}$.
4: **until** $||y^{(k)}|| - 1 < \epsilon$.

Theorem 1 *Let* $\mathbf{P}$ *be an irreducible and aperiodic stochastic matrix with steady-state probability distribution* π. *If* $\nabla_P \neq \mathbf{0}$, *Algorithm I*∇L *provides at each iteration lower bounds for all components of* π *and converges to* π *for any value of the parameters a and b such that* $a \preceq \pi$, $b \preceq \nabla_P$ *and* $b \neq \mathbf{0}$. *Similarly, Algorithm I*∇U *provides at each iteration upper bounds for all components of* π *and converges to* π *for any value of the parameters c and b such that* $\pi \preceq c$, $b \preceq \nabla_P$ *and* $b \neq \mathbf{0}$.

Note that combining both results we obtain a proved envelope for all the components of vector π. The norm of the envelope [3] converges to zero faster than a geometric with rate $(1 - ||b||)$. The algorithms have been extended in [6] to deal with infinite matrix. The algorithms have been implemented in a tool called XBorne [5].

3 Polyhedral Bounds Using Matrix L

The theoretical background is based on Courtois and Semal polyhedral results on steady-state distribution [4]. Let n be the number of states and m the number of non zero entries in $\mathbf{L}$.

Theorem 2 *[7] Given a lower bound* $\mathbf{L}$ *of the transition probability matrix of a given DTMC we can compute bounds for its steady-state probability vector* π. *In a*

first step we compute the steady-state solution of n DTMCs. Transition probability matrix $\mathbf{L}^i$ *associated with the* i*th DTMC is obtained from sub-stochastic matrix* $\mathbf{L}$ *by increasing the elements of column* i *to make* $\mathbf{L}^i$ *stochastic. Let* π^i *be the steady-state probability vector solution of the* i*th DTMC. The lower (resp. upper) bound on the steady-state probability of state* j *is computed as* $\mathbf{min}_i \pi^i[j]$ *(resp.* $\mathbf{max}_i \pi^i[j]$*).*

$$\mathbf{min}_i \pi^i[j] \leq \pi[j] \leq \mathbf{max}_i \pi^i[j] \tag{1}$$

We now show how one can combine Theorem 2 and $I\nabla L$ and $I\nabla U$ algorithms to prove new methods which provide at each iteration a component-wise bound on the steady-state distribution.

Definition 2 Let $\mathbf{L}$ be a sub-stochastic matrix on $\mathcal{F}$, we define for all $i \in \mathcal{F}$: $\alpha^t[i] = 1 - \sum_j \mathbf{L}[i,\ j]$. We assume that $\forall i \in \mathcal{F}\ ,\ \alpha\ [i] > 0$.

The assumption of positivity of every entry of vector α is mandatory for our approach but if the assumption does not hold it is sufficient to consider a new matrix smaller than L such that the assumption on vector α is satisfied. Remember that for all $i \in \mathcal{F}$, $\mathbf{L}^i$ is the matrix built from $\mathbf{L}$ by increasing column i until the matrix becomes stochastic. Thus $\mathbf{L}^i = \mathbf{L} + \alpha^t\, e_i$. Let us denote by π^i the dominant eigenvector of matrix $\mathbf{L}^i$.

Property 1 For all $i \in \mathcal{F}$, we have $\|\nabla_{\mathbf{L}^i}\| > 0$.

Proof The assumption on α implies that $\nabla_{\mathbf{L}^i}[i] > 0$, thus $\nabla_{\mathbf{L}^i} \neq \mathbf{0}$.

Algorithm for an Upper Bounding Distribution: Let $Y^{(k),i}$ be an upper bound at iteration k provide by iterative algorithm $I\nabla U$ for matrix $\mathbf{L}^i$. Due to the results in [3] and Theorem 2, we have:

Lemma 1 $\forall i \in \mathcal{F}$, *we have* $\pi^i \preceq Y^{(k),i}$, *thus* $\boldsymbol{max}_i\{\pi^i\} \preceq \boldsymbol{max}_i\{Y^{(k),i}\}$.

Lemma 2 *For all* $k \geq 0$, $\pi \preceq \boldsymbol{max}_i\{\pi^i\} \preceq \boldsymbol{max}_i\{Y^{(k),i}\}$.

Based on $I\nabla U$ algorithm and the previous results, we derive an iterative algorithm to compute at each iteration (index k in the algorithm) an upper bound on the steady-state distribution of $\mathbf{L}$. The main idea behind this algorithm is to compute first, for all $i \in \mathcal{F}$ an upper bound $Y^{(k),i}$ associated with matrix $\mathbf{L}^i$ using $I\nabla U$ algorithm. Then, we apply Muntz's result to deduce an upper bound on steady-state distribution of π. This process is iterated until the stopping criterion is reached. It is proved in [3] that each sequence $Y^{(k),i}$ converges faster than a geometric with rate $(1 - ||\nabla_{\mathbf{L}^i}||)$. Thus the stopping criterion is the sum of the residual for all sequences $Y^{(k),i}$. Once all these sequences have converged, the **min** operator between the distributions does not change either. Note that in the case where all the sequences $Y^{(k),i}$ have converged, the bounds we give are equal to the bounds provided by Muntz.

Algorithm 3 Algorithm Iterate on ∇ Upper Bound on a Set ($I\nabla US$)

Require: $\forall\, i \in \mathcal{F},\ \alpha[i] > 0$;
Ensure: Successive values of $Y^{(k)}$.

1: $\forall i \in \mathcal{F}, \mathbf{L}^i = \mathbf{L} + \alpha^t e_i, c^i = \Delta_{\mathbf{L}^i}, b^i = \nabla_{\mathbf{L}^i}, Y^{(0),i} = c^i$.
2: **repeat**
3: $\quad \forall\, \mathrm{i} \in \mathcal{F}, Y^{(k+1),\,i} = \mathbf{min}\left\{Y^{(k),\,i}, Y^{(k),\,i}\mathbf{L}^i + b^i(1 - ||Y^{(k),\,i}||)\right\}$.
4: $\quad Y^{(k+1)} = \mathbf{max}_i\left\{Y^{(k+1),i}\right\}$.
5: **until** $\sum_i (||Y^{(k),i}|| - 1) < \epsilon$.

Theorem 3 *Let* $\mathbf{L}$ *be an irreducible sub-stochastic matrix, Algorithm 3 provides at each iteration k an element-wise upper bound on the steady-state distribution of any ergodic matrix entry-wise larger than* $\mathbf{L}$.

Proof This is a straightforward application of Lemma 2.

The algorithm is based on a multiplication of a vector by a matrix, an addition of two vectors and an entry-wise comparison of n vectors. Thus we observe that the cost of the algorithm is dominated by the product of $Y^{(k),\,i}$ by $\mathbf{L}^i$. Muntz's approach has a much larger complexity: we need to consider also n matrices and compute their steady-state distribution using an accurate technique such as GTH [8]. The complexity is $O(n^4)$ using a full matrix representation for $\mathbf{L}$.

Property 2 The complexity of each iteration of $I\nabla US$ algorithm is $O(n.m)$ if we use a sparse matrix representation and $m > n$.

Proof Assume that matrix $\mathbf{L}$ has a sparse representation with n rows and m non-zero elements, the augmented matrix $\mathbf{L}^i,\ i \in \mathcal{F}$ has at most $m + n$ non-zero elements. Therefore, the complexity of the product vector-matrix is $O(n.m)$ if $m > n$. Furthermore, the complexity for searching the maximum (resp. minimum) in an unsorted vector of length n is $O(n)$.

Example 1 Consider matrix $\mathbf{L} = \begin{pmatrix} 0.5 & 0.4 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.8 & 0.0 \\ 0.1 & 0.0 & 0.0 & 0.6 \\ 0.3 & 0.0 & 0.0 & 0.6 \end{pmatrix}$. Algorithm $I\nabla US$ provides the following upper bounds:

k	1	2	3	4	$\|\|Y^{(k)}\|\| - 1$
1	0.6000	0.5000	0.6600	0.9000	1.6600
3	0.5888	0.4110	0.3624	0.7880	1.1502
11	0.4769	0.3245	0.2966	0.5853	0.6833
21	0.4570	0.2810	0.2646	0.5139	0.5165
41	0.4546	0.2606	0.2496	0.4803	0.4450
61	0.4545	0.2581	0.2478	0.4762	0.4366
79	0.4545	0.2578	0.2476	0.4757	0.4356

Algorithm for Lower Bounding Distribution: Let $X^{(k),i}$ be a lower bound of the steady-state distribution of L^i at iteration k provided by iterative algorithm $I \nabla L$.

Lemma 3 *For all $k \geq 0$, we have: $\mathbf{0} \preceq \boldsymbol{min}_i\{X^{(k),i}\} \preceq \boldsymbol{min}_i\{\pi^i\} \preceq \pi$.*

Vector $\mathbf{min}_i\{X^{(k),i}\}$ is non negative. We now search the smallest index k which provides a vector with at least one positive entry and the smallest index such that all entries are positive. Let us introduce some notions of graph theory. We associate with matrix $\mathbf{M}$ a directed graph (digraph in the following) G with vertex set $V = \{1, 2, \ldots, n\}$ and directed edges set E associated with the non zero entries of $\mathbf{M}$. We assume that $\mathbf{M}$ is irreducible. Thus G is strongly connected.

Distance The *distance* from vertex x to vertex y (denoted as $\mathbf{d}^{\mathbf{M}}[x, y]$) is the number of directed edges of a shortest path from x to y in G.
Eccentricity For a given vertex x, the *eccentricity* denoted $\mathbf{Ecc}^{\mathbf{M}}[x]$ is defined to be the distance from x to the farthest vertex:
$\mathbf{Ecc}^{\mathbf{M}}[x] = \mathbf{max}_{y \in V}\{\mathbf{d}^{\mathbf{M}[x, y]}\}$.
Diameter The *diameter* of G, denoted $\mathbf{Diam}^{\mathbf{M}}$, is the value of the largest eccentricity:
$\mathbf{Diam}^{\mathbf{M}} = \mathbf{max}_{x \in V}\mathbf{max}_{y \in V}\{\mathbf{d}^{\mathbf{M}}[x, y]\} = \mathbf{max}_{x \in V}\{\mathbf{Ecc}^{\mathbf{M}}[x]\}$.

Lemma 4 *For all $i, j \in \mathcal{F}$: $\boldsymbol{d}^{\mathbf{L}^i}[i, j] = \boldsymbol{d}^{\mathbf{L}}[i, j]$ and $\boldsymbol{Ecc}^{\mathbf{L}^i}[i] = \boldsymbol{Ecc}^{\mathbf{L}}[i]$ (i.e. adding column i to $\mathbf{L}$ does not change neither the distance from i to an arbitrary node j, nor the eccentricity of i). Furthermore,*
$\boldsymbol{max}_i\{\boldsymbol{Ecc}^{\mathbf{L}^i}[i]\} = \boldsymbol{max}_i\{\boldsymbol{Ecc}^{\mathbf{L}}[i]\} = \boldsymbol{Diam}^{\mathbf{L}}$.

Proof Adding column i to matrix $\mathbf{L}$ implies that we add directed edges entering node i in the digraph. Such edges are not used to find the shortest path from node i to other nodes of the digraph. Thus the distance out of i and the eccentricity of i are kept unchanged.

Property 3 If $k \geq \mathbf{d}^{\mathbf{L}^i}[i, j]$ then $X^{(k),i}[j] \neq 0$. Furthermore, if $k \geq \mathbf{Ecc}^{\mathbf{L}^i}[i]$ then $X^{(k),i}[j] \neq 0$ for all $j \in \mathcal{F}$.

Theorem 4 *Let $\mathbf{L}$ be an irreducible sub-stochastic matrix, Algorithm 4 provides at each iteration k a lower bound on steady-state distribution $X^{(k)}$ and for all $k \geq \boldsymbol{Diam}^{\mathbf{L}}$ all the entries of $X^{(k)}$ are positive.*

Proof The first statement is a direct consequence of Lemma 3 and Theorem 2. Furthermore, Property 3 states that for all $k \geq \mathbf{Ecc}^{\mathbf{L}^i}[i]$ we have $X^{(k),i}[j] \neq 0$ for all $j \in \mathcal{F}$. Due to Lemma 4, we can simplify $\mathbf{Ecc}^{\mathbf{L}^i}[i] = \mathbf{Ecc}^{\mathbf{L}}[i]$. Therefore, we have, after considering all the indices i and taking the maximum of the eccentricity: $\forall k \geq \mathbf{max}_i(\mathbf{Ecc}^{\mathbf{L}}[i]), \quad X^{(k)} \neq 0, \forall j \in \mathcal{F}$. Remembering that the maximum of the eccentricity is the diameter completes the proof. □

Similarly, it comes immediately from Property 3 that if $k > \mathbf{max}_i(\mathbf{d}^{\mathbf{L}}[i, j])$, then $X^{(k)}[j] > 0$. Thus if $k > \mathbf{min}_j\ \mathbf{max}_i(d^{\mathbf{L}}[i, j])$, then $X^{(k)} \neq \mathbf{0}$.

Algorithm 4 Algorithm Iterate on ∇ Lower Bound on a Set ($I\nabla LS$)

Require: $\forall\, i \in \mathcal{F},\ \alpha[i] > 0$;
Ensure: Successive values of $X^{(k)}$.

1: $\forall i \in \mathcal{F}, \mathbf{L}^i = \mathbf{L} + \alpha^t e_i, b^i = \nabla_{\mathbf{L}^i}, X^{(0),i} = b^i$.
2: **repeat**
3: $\quad \forall\, \mathrm{i} \in \mathcal{F},\ X^{(k+1),\,i} = \mathbf{max}\left\{X^{(k),\,i}, X^{(k),\,i}\mathbf{L}^i + b^i(1 - ||X^{(k),\,i}||)\right\}$.
4: $\quad X^{(k+1)} = \mathbf{min}_i\left\{X^{(k+1),i}\right\}$.
5: **until** $\sum_i (1 - ||X^{(k),i}||) < \epsilon$.

The complexity of $I\nabla LS$ is roughly the same to the $I\nabla US$ algorithm at each iteration and equal to $O(m.n.\mathbf{Diam}^{\mathbf{L}})$ to reach the first non trivial lower bound on each component of the steady-state distribution.

Example 2 Consider again matrix $\mathbf{L}$ introduced in Example 1. We compute the vector of the eccentricity for the states in $\mathcal{F}$, $\mathbf{Ecc}^{\mathbf{L}} = [3,\ 2,\ 2,\ 3]$. Therefore $\mathbf{Diam}^{\mathbf{L}} = 3$. Algorithm $I\nabla LS$ provides the following lower bounds:

k	1	2	3	4	$1 - \|\|X^{(k)}\|\|$
1	0.0000	0.0000	0.0000	0.0000	1.0000
3	0.0272	0.0140	0.0096	0.0384	0.9108
11	0.1427	0.0692	0.0609	0.1887	0.5386
21	0.1975	0.0951	0.0848	0.2150	0.4077
41	0.2232	0.1072	0.0960	0.2181	0.3554
61	0.2264	0.1087	0.0974	0.2182	0.3494
79	0.2267	0.1089	0.0975	0.2182	0.3487

4 Comparison and Final Remarks

The main advantage of our approach is that it is much faster than the technique introduced by Muntz. Both approaches are based on the polyhedral theory. The algorithms proposed by Muntz and his colleagues need to compute the steady-state distribution of n Markov chains. The most accurate solution is to use GTH algorithm. Indeed we do not have a proved convergence test for iterative algorithm such has Gauss Seidel or SOR (see Stewart's book for all the details [9]). Our approach is based on a new algorithm for solving the steady-state, which has several interesting features. First it is an iterative algorithm which provides upper and lower bounds of the steady-state of matrix $\mathbf{L}^i$ at each iteration. The cost of one iteration is $O(n\ m)$. For the lower bound, one must perform at least a number of iteration larger than the diameter to obtain a non trivial bound. Another desirable feature of our technique is that we can obtain bound on a subset of the elements. Indeed, we can compute only some probabilities and not all of them to reduce the number of operations (Table 1).

Table 1 Execution times for the algorithms for random examples

n size of matrix **L**	Execution time (s)		Norm of the lower bound	
	Muntz	$I\nabla LS$, $K = 100$	Muntz	$I\nabla LS$, $K = 100$
50	0.19	0.53	0.76	0.72
100	2.48	1.98	0.73	0.68
200	36.45	8.35	0.70	0.65
500	1.37 10^3	103.88	0.67	0.61

We report the execution times in seconds and the norm of the bounds for all algorithms. We apply them on random Markov chains with size between 50 and 500 and a number of transitions out of each state equal to 4. The implementation is based on Matlab and the experiences were performed on a 2.8 GHz quad-core Xeon processor. We perform 50 experiments for each size.

Acknowledgments This work was partially supported by a grant from CNRS GdR RO 2011.

References

1. Buchholz, P.: An improved method for bounding stationary measures of finite Markov processes. Perform. Eval. **62**, 349–365 (2005)
2. Buchholz, P.: Bounding reward measures of markov models using the Markov decision processes. Numer. Linear Algebra Appl. **18**(6), 919–930 (2011)
3. Busic, A., Fourneau, J.-M.: Iterative component-wise bounds for the steady-state distribution of a Markov chain. Numer. Linear Algebra Appl. **18**(6), 1031–1049 (2011)
4. Courtois, P.-J., Semal, P.: Bounds for the positive eigenvectors of nonnegative matrices and for their approximations by decomposition. J. ACM **31**(4), 804–825 (1984)
5. Fourneau, J.-M., Le Coz, M., Pekergin, N., Quessette, F., An open tool to compute stochastic bounds on steady-state distributions and rewards. In: 11th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2003). IEEE Computer Society, Orlando, FL (2003)
6. Fourneau, J.-M., Quessette, F.: Some improvements for the computation of the steady-state distribution of a Markov chain by monotone sequences of vectors. In: ASMTA, volume 7314 of Lecture Notes in Computer Science, pp. 178–192. Springer (2012)
7. Franceschinis, G., Muntz, R.R.: Bounds for quasi-lumpable Markov chains. In: Performance '93, pp. 223–243. Elsevier Science Publishers B. V. (1994)
8. Grassman, W., Taksar, M., Heyman, D.: Regenerative analysis and steady state distributions for Markov chains. Oper. Res. **33**(5), 1107–1116 (1985)
9. Stewart, W.: Introduction to the Numerical Solution of Markov Chains. Princeton University Press, New Jersey (1995)

Interoperating Infrastructures in Emergencies

Antoine Desmet and Erol Gelenbe

Abstract The great challenge in handling the security and resilience in emergency situations is that threats will typically affect more than one infrastructure. A fire is not only a direct hazard for people but it will also short the electrical system, cutting off the lights and possibly the communications and sensor infrastructure and even create more fires. The Tsunami in Japan in 2011 flooded the nuclear reactors but also cut of the pumps that were designed to respond to any flooding situations. This paper is part of a project that addresses these cascaded failures and studies them via simulation. To provide some quantitative estimates of the effect of such cascaded threats, we use the Distributed Building Evacuation Simulator (DBES) to represent the effect of a hazard (in this case a fire) which destroys the sensor system which is used to compute the best advice given to people that are evacuated during the fire. Our simulations compare the situation when the sensor system is intact, and also when it is compromised. As expected, some results highlight the poor overall system performance when the underlying infrastructures are damaged. However, in some scenarios, the degraded system appears to perform as well as the intact one. An analysis into the fault-tolerance of the system leads to some design guidelines which can be applied to design fault-tolerant systems.

Keywords Cyber-physical systems · Emergency navigation · Interacting critical systems · Wireless sensor networks

A. Desmet (✉) · E. Gelenbe
Intelligent Systems and Networks Group, Department of Electrical and Electronic Engineering, Imperial College, London SW7 2BT, UK
e-mail: a.desmet10@imperial.ac.uk

Erol Gelenbe
e-mail: e.gelenbe@imperial.ac.uk

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_13,

1 Introduction

Whenever an undesirable incident occurs such as a fire, a terrorist act, or an accident that requires the intervention of emergency services, critical infrastructures such as water and electricity, even if just within a single large building, must interact with sensor networks and distributed computation for decision making, and with each other in order to offer overall security for both human beings and goods and services. This paper addresses this precise issue of fusing inter-operable "networks" within the hypothetical situation of a need for emergency evacuation for illustration purposes. Thus emergency management serves as a vignette or test-case for an overall cyber-technical environment that exploits wireless technologies, micro-sensing and distributed decision making when incidents occur that implicate or damage several of the underlying infrastructures. For instance, if the water sprinkling system in a building is actuated by the electrical system, while the sensor system also depends on electrical energy, a fire that is initially detected by the sensors will eventually cause short circuits and further fires in the electrical systems, which in turn will bring down both the sprinkling and the sensor network and the computerized distributed decision making system.

Networking and wireless sensing enable applications in environmental sensing [26], health monitoring [18], surveillance [22], intelligent transportation [28], smart tourism [4], and emergency response [10, 23]. During a fire, temperature and gas sensors are responsible for monitoring the spreading of hazards, cameras can track the spread of the fire and the movement of civilians, ultrasound detects obstacles in the environment, and monitors dynamic changes in built structures through destruction and debris. Intelligent evacuation scheduling can be carried out by cooperation between people with mobile devices, decision nodes, sensors, and civilians with mobile devices [8, 9] and in case of major breakdowns or catastrophes opportunistic communication can also be used [17]. Civilians with mobile devices will follow the best known paths, directions and distributed decisions will help select those paths, and signposting and mobile devices can be used for sharing the best advice with the evacuees. In this area there is an abundant literature; early work [21] assumes that there is only one exit, while *Artificial potential fields* that have long been used in mission planning [14, 19] can be used to compute evacuation paths in a distributed manner where the exit point creates an attractive force that pulls the evacuees (and their mobile devices) towards the exit, while each obstacle and hazard generates a repulsive force pushing them away from obstacles and hazards. On the other hand, shortest paths to exits may lead the evacuees along paths that are close to hazards [21]. The work in [27] uses ideas from *multipath routing* in mobile ad hoc networks to guide people as far away from hazardous regions as possible, and in [24] this approach is extended to a 3D guide for people (downstairs to exits, or to rooftops when there are no obvious safe evacuation paths). Human congestion is considered in [5, 6], where a distributed protocol is proposed to balance the number of evacuees among multiple navigation paths to different exits; each sensor is location-aware and capable of detecting the number of evacuees within its sensing coverage. Geometric

approaches exploits the unique properties of geometric graphs to plan evacuation paths as far as possible from the hazards as in [3] where *Delaunay triangulations* [25] are used to partition a WSN into several triangular areas for planning area-to-area navigation paths in a distributed manner. Since location information regarding sensors and users may not always be available, in [20] a road map is used in each user device to compute navigation paths. Based on distances of sensors to the hazardous areas, the backbone of the road map [2], is created and a shortest path tree rooted at the exit is constructed so that each evacuee can avoid the hazardous areas. In [1], a system is presented which estimates how long a hazard takes to reach a given sensor, to compute evacuation paths which offer the longest safest time for evacuees to make their way out of the building. Concepts such as "hazard time" and the evacuation delay are also considered in [7].

In this paper an emergency evacuation assistance system is presented. Through simulation, we aim to study the variations in overall performance caused by faults in underlying systems. The emergency evacuation system proposed relies on two underlying infrastructures: (a) a user localization sub-system, which sets the origin of the personalized "recommended evacuation path" issued by the system to each user; and (b) the fire monitoring sub-system, which determines the areas that ought to be avoided and defines the safest path for each evacuee. In particular, this paper will explore the impact a degraded fire detection network has on the evacuation assistance system's overall performance. The following section will introduce the model used for this simulation. Section 3 presents and analyzes the experimental results. The final section proposes methods to improve the fault-tolerance of the overall system.

2 Simulation Model

The Distributed Building Evacuation Simulator (DBES) platform is used to simulate emergency scenarios. DBES is an agent-based platform dedicated to building evacuation simulation, comprising of generic and expandable models for buildings, occupants, hazards, evacuation policies, path congestion and more. A distinctive feature of DBES is that it entirely run by autonomous agents. This organic architecture allows simulations to run in a mostly decentralized structure, and supports parallel processing with distribution over a pool of networked computers.

The building simulated is a 3-floor rectangular building of 60×24 m inspired from a building on the Imperial College campus, featuring office space and a ground floor lobby area. Figure 1 shows the floor plans of the two first levels of the this building. The building comprises of two evacuation exits, located at the Southern edge of the ground floor. The two grey arrows on Fig. 1b mark the main flow of users evacuating from the upper floors. The number of building users is set to $\{10, 20, 30, 40\}$ users per floor. For reference, values of 30 and 40 users per floor correspond to the standard occupancy of the real-life building. The building users' normal behavior is to spend most of their time in their *dedicated area*—such as their office—and to make occasional short trips to other areas. As soon as the fire alarm is triggered, the evacuee

Fig. 1 Two floor plans of the three-storey building simulated. The graphics show the sensor nodes (*black dots*) and the network layout (each zone marked with a different line style), and the fire outbreak locations in each scenario (*flame icon*). **a** Floor 2; **b** floor 1

receives a recommended evacuation path on a handheld communication and tracking device, and immediately proceeds to evacuate using this path. For greater consistency, each simulation is iterated 10 times with a different set of initial user locations and designated working areas. The building's fire monitoring system consists of a dense deployment of 40–60 inexpensive wired fire sensors per floor. The sensors are divided into *zones* (5–8 per floor) which stem from the floor's central fire panel. For cost and scalability reasons, the backbone of a zone is formed of a data and power "bus" cable, to which fire sensors connected. Owing to this bus wiring, damage or short-circuiting of any wire or sensor generates a fault condition which affects the entire zone. Figure 1a, b shows the topology of the first and second floor's fire monitoring sensor network. Different *line styles* are applied to distinguish each zone. A tracking subsystem is implemented to determine the location of every user. At this stage, the model represents a simplified RF localization component. A more realistic simulation

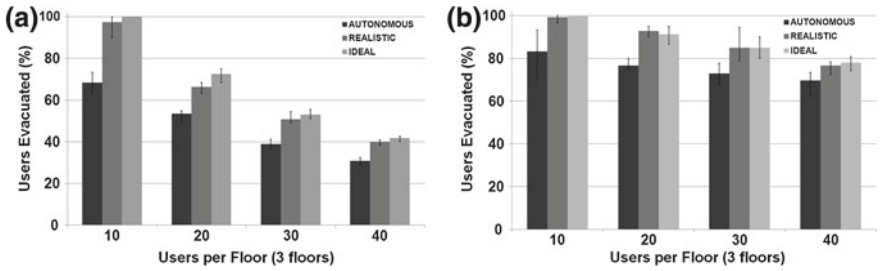


Fig. 2 Evacuation ratios for each experiment. **a** Simulation results for fire starting on floor 1. **b** Simulation results for fire starting on floor 2

of this sub-system, including fire damage, will be undertaken in the next step of this research project.

The evacuation assistance system is hosted on a central server, located in a safe area of the building. Its role is to send individualized evacuation path recommendations to each user, where the path starts from the individual's measured location. The route decision is made using the Dijkstra algorithm, with vertices of the building graph weighed according to their length. The weight of vertices affected by fire is multiplied by a large coefficient, itself proportional to the fire intensity. As a result, the decision system tends to favor paths which are free from any fire hazard, over shorter and perhaps riskier paths. The evacuation assistance system does not implement a mechanism to detect "stale" information: once a fire sensor becomes inoperative, the last value recorded is preserved and used to make all future calculations.

3 Experimental Results

The simulation comprises of two scenarios, featuring different initial fire outbreak locations which are marked by a flame icon on Fig. 1a, b. The results presented on Fig. 2a, b indicate the percentage of building users which managed to evacuate the building, for various densities of users per floor. The results shown are the average of ten simulations, the error bars indicate the single lowest and highest values obtained. Each bar graph has three series which correspond to different models:

- The dark-gray bars serve as a control test, where occupants evacuate the building upon hearing the alarm—without any further form of assistance. This is referred to as the "autonomous" run.
- Bars in gray relate to simulations where fire-induced sensor failure is modeled. Simulation parameters are set so that sensors are able to record moderate fire intensities before eventually defaulting as the fire intensity grows. This model is referred to as the "realistic" model.
- Bars in the light-gray shade represent the outcome of a simulation run where destruction of the fire sensing network does not occur. It is referred to as the

"ideal" model, and shows the "best case" results that a fire-proof system could achieve.

Figure 2a shows somewhat expectable results. User density increases path congestion, which reduces the chances of users evacuating the building in time—regardless of any assistance received. Unaware of the fire outbreak location, autonomous users evacuate the building by following the shortest path they know of. Often, their path meets the fire and forces them to backtrack or follow lengthy bypassing routes, hence the larger casualty count for this population. On the other hand, users in the ideal model benefit from timely and accurate information to avoid the fire. However, casualties do occur in the ideal configuration, as the system generally favors longer, safer paths and does not try to address path congestion. In the realistic scenario, the information provided is only accurate up to the time where the system starts using out-of-date measurements, as sensors become inoperative. A difference in casualty ratio of 5–10 points can be observed between the ideal and realistic configurations, which highlights the importance of modeling coupled sub-components when simulating such complex systems.

The overall evacuation ratios observed in Fig. 2b are better than those in Fig. 2a. The main reason for this is that, generally, the higher the floor on which the fire breaks out, the lesser it affects the downwards flow of users. Beyond this observation, the order of magnitude in evacuation ratios between assisted and autonomous users remains comparable across the two scenarios. However, the most striking difference is that the ideal and realistic models display near-identical results in Fig. 2b. This suggests that the information which is not captured by the defective fire sensor network is of little value, since the degraded system performs as well as its indestructible counterpart. To explain this, let us consider the fire outbreak locations from an evacuation viewpoint. The location on floor 1 corresponds to a low-traffic area which is not on any main evacuation route. The initial outbreak of fire in this dead-end location does not pose any immediate threat to users, and as such, the system merely recommends each user to evacuate following the shortest path to an exit. As the fire grows, a fault eventually develops in the local zone and shuts it down. This condition allows the fire to spread un-monitored, and inevitably reach "evacuation-critical" areas without any user re-routing occurring. On the other hand, in the second floor outbreak scenario, the presence of fire in an evacuation-critical area of the building is reported *before* the sensor network develops a fault. The high-level system uses this information to re-route all user paths which run through this otherwise busy evacuation path—effectively providing users with life-saving advice.

4 Comments and Conclusions

The variations in success rate observed between the two scenarios is owed to the difference in "information value" acquired *before* the sensor network eventually defaulted. The *value* of fire information is location-dependant: the more evacuation

paths tend to run through a location, the higher the value of fire information pertaining to this area. To create a dependable and fault-tolerant system, the underlying sensor network must be designed in a way that maximizes the amount of high-value information acquired before defaulting.

Thus the first step we have addressed in designing a robust underlying infrastructure is to the identify essential components, i.e. sensors covering paths which are evacuation-critical. A "criticality metric" can be derived from the amount of use each path segment gets during a set of evacuation drills. Before considering hardware redundancy or hardened components, resiliency can be improved by re-designing the zones. Assuming that every zone will report a few fire readings before defaulting, overall, a zone is only guaranteed to provide information as valuable as its *least-critical* senor can provide. Based on this observation, creating zones with homogeneous sensors (in term of information value) has the potential to improve results. In particular, neighboring high-value sensors should be wired into a dedicated zone. Resiliency can also be reinforced by adaptive on-line behavior of the agents involved in an emergency as has been done for quality of service driven routing in networks [12].

Having highlighted the flaws of a simplistic sensor network layout, and proposed a design method to increase the fault-tolerance of such networks, the next step we have addressed is to validate its effectiveness against other designs. Another aspect that needs to be investigate are mechanism to detect and handle faulty zones. While the system can be set to assume that the entire area covered by a faulty zone is hazardous, this conservative reasoning may be detrimental to users, especially for large zones where a confined hazard would cause users to be re-routed around the whole area, needlessly slowing down the evacuation process.

While in this paper we have focused on simulations to study of inter-network dependencies and fault-tolerance in emergency management, in future work we may need to call upon probability modeling and distributed control that originates in computer systems, networks and distributed systems [11, 15, 16] and on models of uncertainty of information [13] provide an assessment of the accuracy of the information that is being collected, and offer computationally fast predictions of overall performance prior to simulations.

References

1. Barnes, M., Leather, H., Arvind, D.K.: Emergency evacuation using wireless sensor networks. In: IEEE Conference on Local, Computer Networks, pp. 851–857 (2007)
2. Bruck, J., Gao, J., Jiang, A.A.: MAP: medial axis based geometric routing in sensor networks. In: ACM International Conference on Mobile Computing and Networking, pp. 88–102 (2005)
3. Chen, P.Y., Chen, W.T., Shen, Y.T.: A distributed area-based guiding navigation protocol for wireless sensor networks. In: IEEE International Conference on Parallel and Distributed Systems, pp. 647–654 (2008)
4. Chen, P.Y., Chen, W.T., Tseng, Y.C., Huang, C.F.: Providing group tour guide by rfids and wireless sensor networks. IEEE Trans. Wirel. Commun. **8**(2), 1276–1536 (2009)

5. Chen, P.Y., Kao, Z.F., Chen, W.T., Lin, C.H.: A distributed flow-based guiding navigation protocol in wireless sensor networks. In: International Conference on Parallel Processing (2011) (to appear)
6. Chen, W.T., Chen, P.Y., Wu, C.H., Huang, C.F.: A load-balanced guiding navigation protocol in wireless sensor networks. In: IEEE Global Telecommunications Conference, pp. 1–6 (2008)
7. Cherniak, A., Zadorozhny, V.: Towards adaptive sensor data management for distributed fire evacuation infrastructure. In: International Conference on Mobile Data, Management, pp. 151–156 (2010)
8. Filippoupolitis, A., Gelenbe, E.: A decision support system for disaster management in buildings. In: Summer Computer Simulation Conference, pp. 141–147 (2009)
9. Filippoupolitis, A., Gelenbe, E.: A distributed decision support system for building evacuation. In: IEEE International Conference on Human System, Interactions, pp. 320–327 (2009)
10. Fischer, C., Gellersen, H.: Location and navigation support for emergency responders: a survey. IEEE Pervasive Comput. **9**(1), 38–47 (2009)
11. Gelenbe, E.: A unified approach to the evaluation of a class of replacement algorithms. IEEE Trans. Comput. **C-22**(6), 611–618 (1973)
12. Gelenbe, E.: Steps towards self-aware networks. Commun. ACM **52**, 66–75 (2009)
13. Gelenbe, E., Hébrail, G.: A probability model of uncertainty in data bases. In: ICDE, pp. 328–333. IEEE Computer Society (1986)
14. Gelenbe, E., Hussain, K., Kaptan, V.: Simulating autonomous agents in augmented reality. J. Syst. Softw. **74**(3), 255–268 (2005)
15. Gelenbe, E., Sevcik, K.C.: Analysis of update synchronisation algorithms for multiple copy data bases. IEEE Trans. Comput. **C-28**(10), 737–747 (1979)
16. Gelenbe, E., Stafylopatis, A.: Global behavior of homogeneous random neural systems. Appl. Math. Model. **15**, 534-541 (1991)
17. Gorbil, G., Gelenbe, E.: Opportunistic communications for emergency support systems. In: International Conference on Ambient Systems, Networks and Technologies, pp. 1–9 (2011)
18. Hu, F., Xiao, Y., Hao, Q.: Congestion-aware, loss-resilient bio-monitoring sensor networking for mobile health applications. IEEE J. Sel. Areas Commun. **27**(4), 450–465 (2009)
19. Kaptan, V., Gelenbe, E.: Fusing terrain and goals: agent control in urban environments. In: Dasarathy, B.V. (ed.) Multisource Information Fusion: Architectures, Algorithms, and Applications, vol. 6242, pp. 71–79. SPIE (2006)
20. Li, M., Liu, Y., Wang, J., Yang, Z.: Sensor network navigation without locations. In: IEEE INFOCOM, pp. 2419–2427 (2009)
21. Li, Q., Rosa, M.D., Rus, D.: Distributed algorithms for guiding navigation across a sensor network. In: ACM International Conference on Mobile Computing and Networking, pp. 313–325 (2003)
22. Liu, H., Wan, P., Jia, X.: Maximal lifetime scheduling for sensor surveillance systems with k sensors to one target. IEEE Trans. Parallel Distrib. Syst. **17**(12), 1526–1536 (2006)
23. Malan, D.J., Fulford-Jones, T.R., Nawoj, A., Clavel, A., Shnayder, V., Mainland, G., Welsh, M., Moulton, S.: Sensor networks for emergency response: challenges and opportunities. IEEE Pervasive Comput. **3**(4), 16–23 (2004)
24. Pan, M.S., Tsai, C.H., Tseng, Y.C.: Emergency guiding and monitoring applications in indoor 3D environments by wireless sensor networks. Int. J. Sens. Netw. **1**(1/2), 2–10 (2006)
25. Preparata, F.P., Shamos, M.I.: Computational Geometry: An Introduction. Springer, New York (1985)
26. Terrestrial ecology observing systems, center for embedded networked sensing. http://research.cens.ucla.edu/
27. Tseng, Y.C., Pan, M.S., Tsai, Y.Y.: Wireless sensor networks for emergency navigation. IEEE Comput. **39**(7), 55–62 (2006)
28. Tubaishat, M., Zhuang, P., Qi, Q., Shang, Y.: Wireless sensor networks in intelligent transportation systems. Wirel. Commun. Mob. Comput. **9**(3), 287–302 (2009)

Cooperating Stochastic Automata: Approximate Lumping an Reversed Process

S. Balsamo, G. Dei Rossi and A. Marin

Abstract The paper aims at defining a novel procedure for approximating the steady-state distribution of cooperating stochastic models using a component-wise lumping. Differently from previous approaches, we consider also the possibility of lumping the reversed processes of the cooperating components and show the benefits of this approach in a case study.

1 Introduction

The steady-state analysis of cooperating Markov chains plays a central role in performance engineering but its application is somehow limited by the large number of states that the models can quickly reach when consisting of many components. Following the line of [5], we propose to replace the cooperating models by simpler ones (with less states) that approximate the behaviour of the original. With respect to the *strong equivalence* introduced in [5], our work exploits the results given in [1] in which it is shown that, under a set of assumptions, the lumping can be applied to the reversed Markov chains underlying the various components. With respect to other work on exact or approximate lumping, we remark that we apply the lumping to the component processes and not to the joint process. On the one hand, this is computationally efficient but on the other we have to consider a stricter notion of lumpability that takes into account how the components cooperate. The main contribution of this paper is introducing a framework in which clustering algorithms can be applied

S. Balsamo · G. Dei Rossi (✉) · A. Marin
DAIS, Università Ca' Foscari di Venezia, Venezia, Italy
e-mail: deirossi@dais.unive.it

S. Balsamo
e-mail: balsamo@dais.unive.it

A. Marin
e-mail: marin@dais.unive.it

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_14,

to obtain approximate lumping processes of a set of cooperating components. We introduce an error measure of an approximate lumping, and we show that this allows us to control the quality of the obtained approximation.

2 Theoretical Background

We briefly recall some notions given in [1] in order to keep the paper self-contained. We consider pairwise cooperation of stochastic automata (as, e.g., in [3]). Bold letters denote matrices and row vectors. $\mathbf{e}_n$ is the n–dimension vector whose components are all 1, $\mathbf{I}_n$ is the identity matrix of size $n \times n$. Sizes are omitted when they can be implicitly assumed. Let us consider a pair of components M_1 and M_2 which synchronise on a set of transition types $\mathcal{T} = \{1, 2, \ldots, T\}$. The rate of a transition type is a positive real number $\lambda_t, t \in \mathcal{T}$. For each label $t \in \mathcal{T}$, we define two matrices $\mathbf{E}_{1t}$ and $\mathbf{E}_{2t}$ that describe the behaviour of component M_1 and M_2, respectively, with respect to synchronisation t and whose dimensions are $N_k \times N_k$ for $\mathbf{E}_{kt}$, with $k = 1, 2$ and N_k representing the number of states of component M_k. Matrix element $\mathbf{E}_{kt}(s, s')$ denotes the probability that automaton M_k moves from state s to state s' joint with a transition with the same type t performed by the other automaton; hence $1 \leq s, s' \leq N_k$, and $0 \leq \mathbf{E}_{kt}(s, s') \leq 1$. Moreover, the sum $R_{kt}(s)$ of any row s of matrix $\mathbf{E}_{kt}$ is in the interval [0, 1] and can be interpreted as the probability that component M_k accepts to synchronise on t given that its actual state is s. Under exponential assumptions, the infinitesimal generator $\mathbf{Q}$ of the CTMC underlying the synchronisation of the two automata is defined as in [2, 3].

The main restriction we consider in this work concerns the class of synchronisations that we admit in our model. We say that type $t \in \mathcal{T}$ is *non-blocking* if for at least one of the cooperating automata M_1 and M_2 it holds that $R_{kt}(s) = 1$ for all $s = 1, \ldots, N_k$. In this case we say that t is active in M_h, with $h \neq k$, and passive in $M_k, k, h \in \{1, 2\}$. The model defined by the cooperation of M_1 and M_2 on transition types $\mathcal{T}$ is *feed-forward* if it is possible to identify a model M_k and M_h, $h \neq k$, $h, k \in \{1, 2\}$ such that for all $t \in \mathcal{T}$ one of the following holds:

1. t is active in M_k and passive in M_h,
2. t is active in M_h and passive in M_k and $\mathbf{E}_{kt} = \mathbf{I}$.

We call M_k and M_h the active and passive model, respectively. Without loss of generality, we henceforth consider M_1 active and M_2 passive. Moreover, we order the transition types such that: for $t = 1$, we have $\mathbf{E}_{21} = \mathbf{I}$, for $t = 2$ we have $\mathbf{E}_{12} = \mathbf{I}$ and for $2 < t \leq T$ we have that t is passive in M_2. Moreover, we write $q_2^2(s_2, s_2') = \lambda_2 \mathbf{E}_{22}(s_2, s_2')$, with $1 \leq s_2, s_2' \leq N_2$ and $q_1^t(s_1, s_1') = \lambda_t \mathbf{E}_{1t}(s_1, s_1')$ for $1 \leq t \leq T, t \neq 2$ and $1 \leq s_1, s_1' \leq N_1$.

Roughly speaking, we aim at replacing the active component M_1 by a smaller one denoted by $\tilde{M}_1$ such that the marginal distribution of M_2 in the cooperation $\tilde{M}_1 \otimes M_2$ is identical to that of M_2 in the cooperation $M_1 \otimes M_2$ as proposed in [4, 5], where $\otimes$

is defined as in [2]. However, we both consider the possibility of lumping the forward or the reversed automaton as in [1].

Definition 1 (**Exact lumped automata**). *Given active automaton M_1, a set of transition types $\mathcal{T}$, and a partition of the states of M_1 into $\tilde{N}_1$ clusters $\mathcal{S} = \{\tilde{1}, \tilde{2}, \ldots, \tilde{N}_1\}$, we say that $\mathcal{S}$ is an exact lumping for M_1 if:*

1. $\forall \tilde{s}_1, \tilde{s}_1' \in \mathcal{S}, \tilde{s}_1' \neq \tilde{s}_1, \forall s_1 \in \tilde{s}_1 \;\; \varphi_1^1(s_1, \tilde{s}_1') = \tilde{q}_1^1(\tilde{s}_1, \tilde{s}_1')$, *for some* $\tilde{q}_1^1(\tilde{s}_1, \tilde{s}_1') > 0$
2. $\forall t > 2, \forall \tilde{s}_1, \tilde{s}_1' \in \mathcal{S}, \forall s_1 \in \tilde{s}_1 \;\; \varphi_1^t(s_1, \tilde{s}_1') = \tilde{q}_1^t(\tilde{s}_1, \tilde{s}_1')$, *for some* $\tilde{q}_1^t(\tilde{s}_1, \tilde{s}_1') > 0$

where $\varphi_1^t(s_1, \tilde{s}_1') = \sum_{s_1' \in \tilde{s}_1'} q_1^t(s_1, s_1')$. *If M_1 is lumpable with respect to $\mathcal{S}$, we define the automaton $\tilde{M}_1$ with $\tilde{N}_1$ states as follows:*

$$\tilde{\mathbf{E}}_{11}(\tilde{s}_1, \tilde{s}_1') = \begin{cases} \tilde{q}_1^1(\tilde{s}_1, \tilde{s}_1')\tilde{\lambda}_1^{-1} & \text{if } \tilde{s}_1 \neq \tilde{s}_1' \\ 0 & \text{otherwise} \end{cases}, \tilde{\mathbf{E}}_{12} = \mathbf{I}, \tilde{\mathbf{E}}_{1t}(\tilde{s}_1, \tilde{s}_1') = \tilde{q}_1^t(\tilde{s}_1, \tilde{s}_1')\tilde{\lambda}_t^{-1} \; t > 2$$

where: $\tilde{\lambda}_t = \max_{\tilde{s}_1 = 1, \ldots, \tilde{N}_1} \left(\sum_{\tilde{s}_1' = \tilde{1}}^{\tilde{N}_1} \tilde{q}_1^t(\tilde{s}_1, \tilde{s}_1') \right)$ *for* $t \neq 2$, $\tilde{\lambda}_2 = \lambda_2$ *are the rates associated with the transition types in the cooperation between $\tilde{M}_1$ and M_2.*

As one may expect, if $\tilde{M}_1$ is an exact lumped automaton of M_1, then the CTMC underlying $\tilde{M}_1$ is an exact lumping of that of M_1 in the standard sense of [8]. In what follows we assume that M_1, M_2 and their cooperation to have ergodic underlying CTMCs.

Theorem 1 *Given the model $M_1 \otimes M_2$, let $\tilde{M}_1$ be an exact lumping of M_1 whose clusters are $\mathcal{S} = \{\tilde{1}, \ldots, \tilde{N}_1\}$. Then, the marginal steady-state distribution π_2 of M_2 is given by:*

$$\forall s_2 = 1, \ldots, N_2, \quad \pi_2(s_2) = \sum_{s_1=1}^{N_1} \pi(s_1, s_2) = \sum_{\tilde{s}_1=1}^{\tilde{N}_1} \pi^*(\tilde{s}_1, s_2), \tag{1}$$

where π is the steady-state distribution of the cooperation between M_1 and M_2 and π^* that of the cooperation between $\tilde{M}_1$ and M_2.

Timed-reversed automata Theorem 2 relies on the theory of reversed Markov processes as studied in [7]. From the transition rates of a reversed CTMC we can efficiently compute the unnormalised steady-state distribution and vice versa. Using those results, we give the following definition:

Definition 2 (**Timed-reversed automata**). *Given the active automaton M_1 synchronising on transition type $\mathcal{T}$ with rates $\lambda_1, \ldots, \lambda_T$, we define the timed-reversed automaton M_1^R as follows:*

$$\mathbf{E}_{1t}^R(s_1, s_1') = \frac{\pi_1(s_1')}{\pi_1(s_1)} q_1^t(s_1', s_1) \frac{1}{\lambda_t^R} \quad t \neq 2 \qquad\qquad \mathbf{E}_{12}^R = \mathbf{I}$$

where : $\lambda_t^R = \max_{s_1=1,\ldots,N_1} \left(\sum_{s_1'=1}^{N_1} q_1^{tR}(s_1, s_1') \right)$
and $q_1^{tR}(s_1, s_1') = (\pi_1(s_1')/\pi_1(s_1)) q_1^t(s_1', s_1)$, *for all* $1 \le s_1, s_1' \le N_1$.

Theorem 2 *Given the model* $M_1 \otimes M_2$, *let* M_1^R *be the reversed automaton of* M_1 *and let* $\tilde{M}_1^R$ *be an exact lumping of* M_1^R *whose clusters are* $\mathcal{S} = \{\tilde{1}, \ldots, \tilde{N}_1\}$. *Then, the marginal steady-state distribution* π_2 *of* M_2 *is given by:*

$$\forall s_2 = 1, \ldots, N_2, \pi_2(s_2) = \sum_{s_1=1}^{N_1} \pi(s_1, s_2) = \sum_{\tilde{s}_1=1}^{\tilde{N}_1} \tilde{\pi}^R(\tilde{s}_1, s_2), \tag{2}$$

where π *is the steady-state distribution of the cooperation between* M_1 *and* M_2 *and* $\tilde{\pi}^R$ *that of the cooperation between* $\tilde{M}_1^R$ *and* M_2.

3 Approximate Computation of Marginal Distribution

3.1 Evaluating the Quality of an Approximate Lumping

First, we address the problem of measuring how close an arbitrary state partition is to an exact lumping as given in Definition 1. Let M_1 be the active model and $\mathcal{T}$ the transition type set with the convention that $t = 1$ ($t = 2$) denotes the type of the transitions that M_1 (M_2) can carry out independently of M_2 (M_1), and $t > 2$ denotes the type for the synchronised transitions in which M_1 is active. Given an arbitrary partition $\mathcal{W}$ (note that we reserve $\mathcal{S}$ for denoting partitions that are also lumpings), we measure the coefficient of variation of the outgoing fluxes $\phi_1^t(s_1)$ of the states in $\tilde{s}_1$. In the following definitions we assume the empty cluster (that would lead to a $0/0$ in the definition) to have error 0.

Definition 3 (ϵ-**error**). *Given model* M_1 *and a partition of states* $\mathcal{W} = \{\tilde{1}, \ldots, \tilde{N}_1\}$, *for all* $\tilde{s}_1 \in \mathcal{W}$ *and* $t > 2$, *we define:*

$$\overline{\phi}_1^t(\tilde{s}_1) = \left(\sum_{s_1 \in \tilde{s}_1} \pi_1(s_1) \phi_1^t(s_1) \right) \frac{1}{\sum_{s_1 \in \tilde{s}_1} \pi_1(s_1)}$$

$$\epsilon^t(\tilde{s}_1) = 1 - \exp\left(- \sqrt{ \sum_{s_1 \in \tilde{s}_1} \frac{\pi_1(s_1)(\phi_1^t(s_1) - \overline{\phi}_1^t(\tilde{s}_1))^2}{\sum_{s \in \tilde{s}_1} \pi_1(s_1)} } \right).$$

Observe that $0 \le \epsilon^t(\tilde{s}_1) < 1$ and if automaton $\tilde{M}_1$ is an exact lumping of M_1, then $\forall t > 2, \forall \tilde{s}_1 = 1, \ldots, \tilde{N}_1$ we have that $\epsilon^t(\tilde{s}_1) = 0$. We use the minimisation of error ϵ to perform a first rough clustering of the states of M_1 as described in Sect. 3.2. The following definition gives a more accurate measure of the error of a partition.

Definition 4 (δ-**error**) *Given model M_1 and a partition of states $\mathcal{W} = \{\tilde{1}, \ldots, \tilde{N}_1\}$, for all $\tilde{s}_1, \tilde{s}'_1 \in \mathcal{W}$, we define:*

$$\overline{\varphi}_1^t(\tilde{s}_1, \tilde{s}'_1) = \begin{cases} 0 & \tilde{s}_1 = \tilde{s}'_1 \wedge t = 1 \\ \frac{\left(\sum_{s_1 \in \tilde{s}_1} \pi_1(s_1)\varphi_1^t(s_1, \tilde{s}'_1)\right)}{\sum_{s_1 \in \tilde{s}_1} \pi_1(s_1)} & \textit{otherwise} \end{cases}$$

$$\left(\sigma^t(\tilde{s}_1, \tilde{s}'_1)\right)^2 = \sum_{s_1 \in \tilde{s}_1} \frac{\pi_1(s_1)(\varphi_1^t(s_1, \tilde{s}'_1) - \overline{\varphi}_1^t(\tilde{s}_1, \tilde{s}'_1))^2}{\sum_{s \in \tilde{s}_1} \pi_1(s)} \tag{3}$$

$$\delta^t(\tilde{s}_1, \tilde{s}'_1) = 1 - e^{-\sigma(\tilde{s}_1, \tilde{s}'_1)}$$

where function φ_1^t has been defined in Definition 1.

Similarly to ϵ^t, also for error δ^t we have that $0 \leq \delta^t(\tilde{s}_1, \tilde{s}_2) < 1$.

3.2 Algorithm Definition

In this section we propose an algorithm to approximate the marginal distributions of cooperating stochastic automata. Informally, the algorithm exploits the heuristics defined for solving clustering problems in order to obtain an automaton $M_1^{\simeq}$, which is close to an exact lumping of M_1, where we use the δ- and ϵ-errors to measure the goodness of the approximation of $M_1^{\simeq}$. In what follows we focus our attention on clustering M_1, but the same discussion is obviously valid for M_1^R. An *ideal* algorithm based on the theory we developed should work as illustrated in Table 1. The idea is that the analyst specifies the models and a pair of tolerance constants, $\epsilon \geq 0$, and $\delta \geq 0$ and the algorithm uses the best approximated lumping (i.e., that with the smallest number of clusters) for which each synchronising transition type satisfies the conditions stated in Steps 1 and 3 to compute the marginal steady-state distribution of the components. The algorithm surely terminates because when the number of clusters of the partition is equal to the number of states of M_1, we have an exact lumping. Obviously, if one desires to specify the number of clusters of $M_1^{\simeq}$ rather than the tolerance criteria, the algorithm can be simplified. The following definition specifies how we derive an approximate lumped automaton given a partition of its states $\mathcal{W}$.

Definition 5 (**Approx. lumped automata**). *Given active automaton M_1, a set of transition types $\mathcal{T}$, and a partition of the states of M_1 into $\tilde{N}_1$ clusters $\mathcal{W} = \{\tilde{1}, \tilde{2}, \ldots, \tilde{N}_1\}$, then we define the automaton $M_1^{\simeq}$ as follows:*

$$\tilde{\mathbf{E}}_{11}(\tilde{s}_1, \tilde{s}'_1) = \begin{cases} \overline{\varphi}_1^1(\tilde{s}_1, \tilde{s}'_1)\tilde{\lambda}_1^{-1} & \text{if } \tilde{s}_1 \neq \tilde{s}_2 \\ 0 & \text{otherwise} \end{cases} \quad \tilde{\mathbf{E}}_{12} = \mathbf{I}, \tilde{\mathbf{E}}_{1t}(\tilde{s}_1, \tilde{s}_1) = \overline{\varphi}_1^t(\tilde{s}_1, \tilde{s}'_1)\lambda_t^{-1} \quad t > 2$$

Table 1 Ideal algorithm for computing the approximated marginal distributions of cooperating automata

• Input: automata M_1, M_2, $\mathcal{T}$, tolerances $\epsilon \geq 0$, $\delta \geq 0$
• Output: marginal distribution π_1 of M_1; approximated marginal distribution of M_2
1. Find the minimum $\tilde{N}_1'$ such that there exists a partition $\mathcal{W} = \{\tilde{1}, \ldots, \tilde{N}_1'\}$ of the states of M_1 such that $\forall t \in \mathcal{T}, t > 2$ and $\forall \tilde{s}_1 \in \mathcal{W}\ \epsilon(\tilde{s}_1) \leq \epsilon$
2. Let $\mathcal{W}' \leftarrow \mathcal{W}$
3. Check if partition $\mathcal{W}'$ is such that $\forall t \in \mathcal{T}$, $\forall \tilde{s}_1, \tilde{s}_2 \in \mathcal{W}$, $\tilde{s}_1 \neq \tilde{s}_2$, $\delta^t(\tilde{s}_1, \tilde{s}_1') \leq \delta$. If this is true then return the marginal distribution of M_1 and the approximated of M_2 by computing the marginal distribution of $\tilde{M}_1 \otimes M_2$ and terminate.
4. Otherwise, refine partition $\mathcal{W}$ to obtain $\mathcal{W}^{new}$ such that the number of clusters of $\mathcal{W}^{new}$ is greater than the number of clusters in $\mathcal{W}'$. $\mathcal{W}' \leftarrow \mathcal{W}^{new}$. Repeat from Step 3

where $\tilde{\lambda}_t = \max_{\tilde{s}_1=1,\ldots,\tilde{N}_1} \left(\sum_{\tilde{s}_1'=1}^{\tilde{N}_1} \overline{\varphi}_1^t(\tilde{s}_1, \tilde{s}_1') \right)$ *are the rates associated with the transition types in the cooperation between* $\widetilde{M_1}$ *and* M_2.

The algorithm is called *ideal* because the problem of performing an optimal clustering is known to be NP-hard and hence a sub-optimal solution is usually computed using some heuristics [12]. We now consider the main steps of the algorithm of Table 1 and discuss how they can be implemented in practice. Note that the choice of an algorithm often depends on the characteristics of the dataset that one is studying. Since we consider general automata, we decided to propose at least two different solutions for each of the problems proposed by the algorithm of Table 1, i.e., implementing Step 1 and Step 4.

The initial clustering based on ϵ-error The initial clustering can be implemented with various algorithms. The similarity measure between two states s_1 and s_1' can be the Euclidean distance between the vectors $(\phi_1^3(s_1), \ldots, \phi_1^T(s_1))$ and $(\phi_1^3(s_1'), \ldots, \phi_1^T(s'))$. In case of hierarchical clustering a divisive (top-down) approach can be adopted and the algorithm stops when the condition of ϵ-error stated in Step 1 is satisfied. Another approach, which is used in the example we propose in Sect. 4, is to use K-means. K-means is a fast algorithm for clustering whose drawback consists in the need of specifying a-priori the desired number of clusters K. Therefore, this is usually the good choice if some intuition can drive the analyst in choosing an initial value for K. If the constraints on ϵ cannot be satisfied by the choice of K, then K-means must be run again specifying a number of clusters $K' > K$.

Refining the clustering obtained in Step 1 We consider the problem of refining the clustering obtained in Step 1 in order to satisfy the conditions required by the tolerance constant δ. Note that often the partitions obtained from the first step are sufficient to obtain good approximations in real models such as queues (as shown by the Example of Sect. 4). Let $\mathcal{W} = \{\tilde{s}_1, \ldots, \tilde{s}_n\}$ be the clustering obtained by the first phase of the algorithm and we want to obtain $\mathcal{W}' = \{\tilde{p}_1, \ldots, \tilde{p}_m\}$ with $m > n$ such that $\mathcal{W}'$ is a refinement of $\mathcal{W}$. Note that K-means cannot be straightforwardly

applied to cluster the states because the distance measures among the states depend on the cluster themselves.

Spectral analysis Spectral analysis is applied to stochastic matrices $\mathbf{P}$, therefore we must discretise the CTMC underlying automaton M_1. Let $\mathbf{Q}_1$ be the infinitesimal generator of M_1, then, following [11], let $\Delta t = \max(|\mathbf{Q}(i,i)|)$, where $\mathbf{Q}(i,i)$ are the diagonal elements of $\mathbf{Q}$. Then, we have $\mathbf{P} = \mathbf{Q}\Delta t + \mathbf{I}$. Now observe that an exact lumping for $\mathbf{P}$ is also an exact lumping for $\mathbf{Q}$, and a good approximation for $\mathbf{P}$ is also a good approximation for $\mathbf{Q}$. However, recall that an exact lumping of CTMC corresponding to $\mathbf{Q}$ is a necessary condition for satisfying Definition 1 but not sufficient. This method has been introduced by Jacobi in [6] for identifying approximated lumpings in Markov chains. The algoritm relies on the property that a Markov chain admits an exact lumping with K states if and only if there are exactly K right eigenvectors of $\mathbf{P}$ with elements that are constant over the aggregates (see [6] and the references therein), thus it defines $\mathbf{R} = \mathbf{P}^\top\mathbf{P} - \mathbf{P} - \mathbf{P}^\top + \mathbf{I}$ and compute its eigenvalues. Then, it obtains the eigenvectors $\mathbf{u}_i$ corresponding to the K smallest eigenvalues. Finally, the algorithm associate a vector $\mathbf{v}_j$ with each state j of the process that is defined as $\mathbf{v}_j = (u_j(1), \ldots, u_j(K))$. The last step consists in running K-Means to obtain K clusters. Note that the original algorithm uses the Euclidean distance between vectors to perform the clustering, however we need to obtain a partition $\mathcal{W}'$ that is a refinement of $\mathcal{W}$. Therefore, we define the distance function d as:

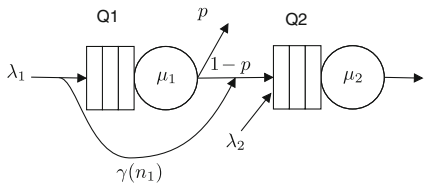
$$d(\mathbf{v}_i, \mathbf{v}_j) = \begin{cases} ||\mathbf{v}_i - \mathbf{v}_j||_2 & \text{if } i, j \in \tilde{s}_1 \text{ for some } s_1 \in \mathcal{W} \\ \infty & \text{otherwise} \end{cases}.$$

Iterative algorithm This algorithm is an adaptation to our framework of the class of clustering algorithms described in [9]. We briefly describe the original version as illustrated in [6]. Let $\mathbf{Z}$ be a $N_1 \times K$ matrix whose element are in $\{0, 1\}$ and each row has exactly one non-zero element and $K \leq N_1$. Let $\tilde{\mathbf{P}}$ be a $K \times K$ matrix. Then, if we can find a matrix $\mathbf{Z}$ such that $\mathbf{PZ} = \mathbf{Z}\tilde{\mathbf{P}}$, $\tilde{\mathbf{P}}$ denotes the transition matrix of a lumped process. Elements $\mathbf{Z}(s_1, \tilde{s}_1)$ of $\mathbf{Z}$ are 1 if $s_1 \in \tilde{s}_1$, 0 otherwise, with $1 \leq s_1 \leq N_1$ and $1 \leq \tilde{s}_1 \leq K$. The approximate lumping is obtained by iteratively assigning a random state s_1 to cluster $\tilde{s}'_1$, where $\tilde{s}'_1$ is chosen as the cluster which minimises the local error measure $||(\mathbf{PZ})(s_1, *) - \mathbf{P}^{\simeq}(\tilde{s}_1, *)||_2$, where $\mathbf{P}^{\simeq}$ is a candidate partition and $\mathbf{P}^{\simeq}(s, *)$ denotes row s of the matrix. Obviously, the algorithm can converge to a local minimum. For a comparison between this approach and the spectral decomposition see [6, 9].

4 Example

In this section we consider an example consisting of a model in which the active automaton is not exactly lumpable, and, by applying the approach described above, we show how state aggregation of an automaton can both drastically reduce the state

Fig. 1 Example model



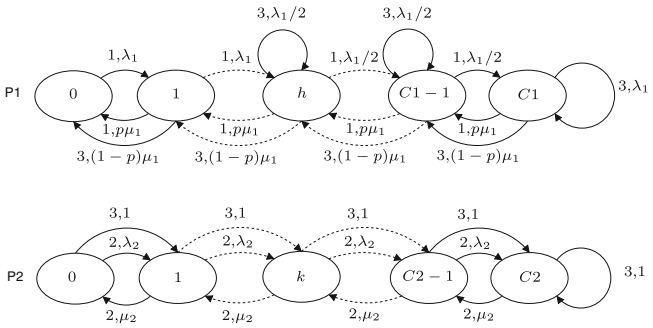


Fig. 2 Automata of the example model

space cardinality and provide a good approximation of the marginal steady-state probabilities of the passive automaton. Consider the queueing network represented in Fig. 1 where the stations Q1 and Q2 have a finite capacity $C1$ and $C2$ and customers arrives at Q1 and Q2 according to homogeneous Poisson processes of parameter λ_1 and λ_2, respectively. The behaviour of the customers arriving at the first queue can be described by a function $\gamma : \{0, \ldots, C1\} \to [0, \lambda_1]$ that given n_1, i.e., the number of customers already present in Q1, gives the rate of customers skipping the first queue, while the rate of the customers that regularly enqueue in the first station is given by $\lambda_1 - \gamma(n_1)$, where function γ is defined as follows:

$$\gamma(n_1) = \begin{cases} 0 & \text{if } n_1 \leq \left\lfloor \dfrac{C1}{2} \right\rfloor \\ \dfrac{\lambda_1}{2} & \text{if } \left\lfloor \dfrac{C1}{2} \right\rfloor < n_1 < C1 \\ \lambda_1 & \text{if } n_1 = C1 \end{cases}$$

When Q2 is full, customers are lost. Figure 2 shows the component-wise stochastic processes underlying the described system, where $P1$ and $P2$ are the stochastic automata for Q1 and Q2, respectively and state h is the first one where γ assumes the value $\lambda_1/2$. We shall apply the techniques and algorithms proposed in Sect. 3 to this example and we present a comparison of our results with those obtained by other methods. The parameters are $C1 = 20$, $C2 = 20$, $\lambda_1 = 6$, $\lambda_2 = 1$, $\mu_1 = 4$, $\mu_2 = 4$, $p = 0.7$.

Using an implementation of the ideal algorithm of Table 1 on the process $P1$ and enforcing the use of K clusters, one should get a partition $\mathcal{L} = \{L_1, \ldots, L_K\}$, where each L_i is a subset of the states of $P1$. Using our implementation with $\epsilon = 10^{-13}$ and $\delta = 0.95$ as tolerances, as we were interested in evaluating a coarse-grained approximated lumping, we obtained 4 clusters, i.e., $L_1 = \{0\}$, $L_2 = \{1, \ldots, 10\}$, $L_3 = \{11, \ldots, 19\}$ and $L_4 = \{20\}$. Notice that the resulting clustering is not an exact lumping, thus Theorem 1 does not hold. We can however use this clustering to compute an approximation of the marginal steady-state probabilities of the process $P2$, reducing drastically the state space of the joint process. Using the same technique, we partitioned the reversed process $P1^R$ of $P1$, obtaining $\overline{L_1} = \{0, \ldots, 10\}$, $\overline{L_2} = \{11, 12\}$, $\overline{L_3} = \{13, \ldots, 19\}$ and $\overline{L_4} = \{20\}$, which is not an exact lumping, thus Theorem 2 cannot be applied.

In order to evaluate the results of our technique, we compared them with the solution obtained with other ones, namely we obtained the marginal steady state probabilities of $P2$ using the following methods:

1. The computation of the joint probabilities between the approximate lumping $\mathcal{L}$ of $P1$ and the process $P2$. [FW-Lump]
2. The computation of the joint probabilities between the approximate lumping $\overline{\mathcal{L}}$ of the of the *reversed* process $P1^R$ and the process $P2$. [RV-Lump]
3. The Approximated Product Form of order 4 introduced in [2]. [APF]
4. The Fixed Point Approximation [10]. [FPA]
5. The exact computation of the joint probabilities between $P1$ and $P2$. [Exact]

In order to compare the quality of the distributions obtained by the analysed methods, we used the *Kullback-Leibler divergence* between the exact marginal probability distribution of P2 and these approximations, computed as:

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)},$$

where P is the exact probability distribution and Q is the estimated one. Table 2 reports the Kullback-Leibler divergence, the average number of customers $E[N]$ and its relative error for Q2. While all the methods offer a reasonable approximation on average performance indices, the error can be remarkable on their steady-state distribution. Although we cannot consider these numerical results a complete study of the relative merit of the various methods, it is possible to point out some differences among them. In particular, in this case, the results given by the approximate lumping of the reversed process are more accurate than those obtained using the forward process. This is because grouping states by the sum of outgoing rates of synchronising transition types leads to 4 sets in the forward process, and to 3 sets in the reversed one. This allows the algorithm to refine one more time one of the clusters by an enforced split in 4 components.

Table 2 Comparison between approximation methods

	FW-lump	RV-lump	APF	FPA	Exact
KL div.	0.0065	0.0045	0.0451	0.0112	0
$E[N]$	11.62	11.55	9.990	11.80	11.33
Rel. err.	0.0259	0.0200	0.1178	0.0424	0

5 Final Remarks

In this paper we have proposed a methodology for approximating the marginal distributions in the cooperations of two stochastic processes through approximated lumping using results from Theorems 1 and 2. The main contribution relies in the evaluation of lumping quality through ϵ and δ-error, and in the possibility to choose the better lumping between the one on the forward and on the one on reversed process. The advantages of this approach lie on the fact that it is done on single cooperating automata and that, under some assumptions, it can lead to better approximations with respect to other popular techniques, as shown in Sect. 4. Future research directions include the application of our methodology to real case studies with large state spaces and the investigation on the relations between error metrics and clustering algorithm.

References

1. Balsamo, S., Dei Rossi, G., Marin, A.: Lumping and reversed processes in cooperating automata. In: Proceedings of International Conference on Analytical and Stochastic Modeling Techniques and Applications (ASMTA), Lecture Notes in Computer Science 7314. pp. 212–226. Springer, Grenoble (2012)
2. Buchholz, P.: Product form approximations for communicating Markov processes. Perf. Eval. **67**(9), 797–815 (2010), Special Issue: QEST 2008
3. Fourneau, J.M., Plateau, B., Stewart, W.J.: Product form for stochastic automata networks. In: Proceedings of ValueTools '07. pp. 1–10. ICST, Brussels, (2007)
4. Gilmore, S., Hillston, J., Ribaudo, M.: An efficient algorithm for aggregating PEPA models. IEEE Trans. Software Eng. **27**(5), 449–464 (2001)
5. Hillston, J.: A compositional approach to performance modelling. Ph.D. thesis, Department of Computer Science, University of Edinburgh (1994)
6. Jacobi, M.N.: A robust spectral method for finding lumpings and meta stable states of non-reversible Markov chains. Elect. Trans. Num. An. **37**, 296–306 (2010)
7. Kelly, F.: Reversibility and Stochastic Networks. Wiley, New York (1979)
8. Kemeny, J.G., Snell, J.L.: Finite Markov Chains, chap. II. D. Van Nostrand Company, inc. (1960)
9. Lafon, S., Lee, A.: Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parametrization. IEEE Trans. Pattern Anal. Mach. Intell. **28**, 1393–1403 (2006)

10. Miner, A.S., Ciardo, G., Donatelli, S.: Using the exact state space of a markov model to compute approximate stationary measures. In: Proceedings of ACM SIGMETRICS. pp. 207–216. ACM, New York (2000)
11. Stewart, W.J.: Probability, Markov Chains, Queues, and Simulation. Princeton University Press, UK (2009)
12. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Trans. Neural Networks **16**(3), 645–678 (2005)

Product Form Solution for a Simple Control of the Power Consumption in a Service Center

J. M. Fourneau

Abstract We analyse a queue or a network of queues where the external arrival rate changes according to a modulating Markov chain. We assume that, in response to this rate fluctuation, a control mechanism changes the power consumption and thus the services rates to keep the loads constant. We prove that this simple control mechanism leads to a Markov chain with a product form solution under usual assumptions.

1 Introduction

Power consumption is one of the main research problem we have to consider to design an efficient cloud computing service [1]. One of the main difficulties is the provisioning and the dimensioning of the service capacities in the presence of a highly fluctuating arrival of jobs. An over dimensioning of the servers leads to a poor system utilisation and a low server utilisation. With the best current designs, the power consumed by an idle server is about 65 % of its peak consumption [6]. Hence, the only realistic way to reduce significantly the power consumption of a service center or a data center is to power down some servers whenever that can be justified by the load conditions [7]. Here, we consider a more abstract representation of the system. We assume that the arrival rates vary according to a modulating Markov chain. In response, we assume that we have a simple control which changes the service rates and the routing probabilities. Changing the service rate relies on a clock modification of the processors and this leads to a modification of the energy used by the server.

J. M. Fourneau (✉)
PRiSM, Université de Versailles-Saint-Quentin, CNRS UMR, 8144 Versailles, France
e-mail: jmf@prism.uvsq.fr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_15,

We basically represent a network of queues modulated by an external Markov chain. This is typically denoted as a network of queues in a random environment. One studies the influence of a stochastic process (usually denoted as a phase) on the evolution of a queue or a set of queues. Only a few results have been presented on open networks of n queues modulated by a phase. As the state space is infinite in several dimensions it is not possible to apply the matrix geometric approach. In [9], Zhu studies the steady-state distribution of a modulated Markovian open network of infinite queues and proves a sufficient condition for the existence of a product form steady-state distribution. The steady-state distribution of the number of customers in a queue is geometrically distributed with ratio $\rho_{i,j}$ for queue i when the phase is in state j. The proof is based on the reversed process of the network of queues. The condition is simple: the ratio $\rho_{i,j}$ must be constant when the phase changes.

In [5], Verchère et al. generalized Zhu's result to multidimensional continuous-time Markov chains (CTMC in the following) in which a phase process models the environment. The transition rates of the network of queues can depend on the state of the phase but the transition rates of the phase do not change when the state of the network evolves. The chain modeling the phases is irreducible and finite. Therefore it has a steady-state distribution. Verchère et al. first proved a general theorem on this model and then derived corollaries when the network of queues is a G-network with positive and negative customers. Unlike Zhu's method, the proof in [5] is not based on reversibility.

More recently we have proposed in [3, 4] a more general results based on Stochastic Automata Networks (SAN in the following) with functions and without synchronisation. Clearly, such a model may readily represent a set of queues modulated by a phase. But it is more general because we allow the transition rates of the phase to depend on the state of the queues. We do not require any constraint on the functions. The proof is based on the properties of the generalised tensor algebra proposed by Plateau [2] which are given in the next section. Then, we will define the model and prove the product-form of the steady-state distribution of the Markov chain. Our simple model leads to an analytical solution. This opens the way to more sophisticated analysis or optimisation problems including stochastic bounds [8] for instance to give some times to the controller to react or to model servers which may be switched on and off by blocks [7].

2 A Brief Introduction to Tensor and Product Form

Let us first define the usual tensor product and sum. Recall that with

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix},$$

the tensor product $D = A \otimes B$ is defined as:

$$\left(\begin{array}{ccc|ccc} a_{11}*b_{11} & a_{11}*b_{12} & a_{11}*b_{13} & a_{12}*b_{11} & a_{12}*b_{12} & a_{12}*b_{13} \\ a_{11}*b_{21} & a_{11}*b_{22} & a_{11}*b_{23} & a_{12}*b_{21} & a_{12}*b_{22} & a_{12}*b_{23} \\ a_{11}*b_{31} & a_{11}*b_{32} & a_{11}*b_{33} & a_{12}*b_{31} & a_{12}*b_{32} & a_{12}*b_{33} \\ \hline a_{21}*b_{11} & a_{21}*b_{12} & a_{21}*b_{13} & a_{22}*b_{11} & a_{22}*b_{12} & a_{22}*b_{13} \\ a_{21}*b_{21} & a_{21}*b_{22} & a_{21}*b_{23} & a_{22}*b_{21} & a_{22}*b_{22} & a_{22}*b_{23} \\ a_{21}*b_{31} & a_{21}*b_{32} & a_{21}*b_{33} & a_{22}*b_{31} & a_{22}*b_{32} & a_{22}*b_{33} \end{array}\right),$$

and the tensor sum $C = A \oplus B$ is given by

$$\left(\begin{array}{ccc|ccc} a_{11}+b_{11} & b_{12} & b_{13} & a_{12} & 0 & 0 \\ b_{21} & a_{11}+b_{22} & b_{23} & 0 & a_{12} & 0 \\ b_{31} & b_{32} & a_{11}+b_{33} & 0 & 0 & a_{12} \\ \hline a_{21} & 0 & 0 & a_{22}+b_{11} & b_{12} & b_{13} \\ 0 & a_{21} & 0 & b_{21} & a_{22}+b_{22} & b_{23} \\ 0 & 0 & a_{21} & b_{31} & b_{32} & a_{22}+b_{33} \end{array}\right).$$

$C = A \oplus B$ is the transition rate matrix of two independent chains associated with A and B. We now turn our attention to the interaction between chains. We assume that the rate at which a transition occurs may be a *function* of the state of a set of chains. Such transitions are called *functional* transitions. Transitions that are not functional are said to be *constant*. Functional rates are non negative but they may be zero. Suppose, for example, that the rate of transition from state 2 to state 3 in the second automaton is d_{23} when the first chain is in state 1 and e_{23} when the first chain is in state 2. Suppose in addition that the rate at which the first chain produces transitions from state 1 to state 2 is f_{12}, g_{12}, h_{12} depending on whether the second chain is in state 1, 2 or 3. The global infinitesimal generator is now ($*$ represents the normalization):

$$\left(\begin{array}{ccc|ccc} * & b_{12} & b_{13} & f_{12} & 0 & 0 \\ b_{21} & * & d_{23} & 0 & g_{12} & 0 \\ b_{31} & b_{32} & * & 0 & 0 & h_{12} \\ \hline a_{21} & 0 & 0 & * & b_{12} & b_{13} \\ 0 & a_{21} & 0 & b_{21} & * & e_{23} \\ 0 & 0 & a_{21} & b_{31} & b_{32} & * \end{array}\right).$$

Definition 1 (**Functional stochastic matrix**) *F is a functional stochastic matrix if and only if the entries of F (denoted as f_{ij}) are non negative functions for each index $i \neq j$.*

Due to the functional entries in the stochastic matrices, the chains are not independent. Therefore in general the steady-state distribution does not have product form. To deal with functional matrices, Plateau has introduced a generalization of the tensor products denoted as $\otimes_g$ in the following.

Definition 2 (**Generalized Tensor Product** [2]) *Assume that matrix A is a function of the state of B and matrix B is a function of the state of A. We denote this dependence as* $A(\mathcal{B})$ *and* $B(\mathcal{A})$. *The generalized tensor product* $C = A \otimes_g B$ *is defined by* $c_{\{(i_1,j_1);(i_2,j_2)\}} = a_{i_1 j_1}(i_2) b_{i_2 j_2}(i_1)$, *because before the transition, the state of A is* i_1 *and the state of B is* i_2.

Similarly we define the Generalized Tensor Sum (denoted as $\oplus_g$) as usual:

$$D = A(\mathcal{B}) \oplus_g B(\mathcal{A}) \Leftrightarrow D = A(\mathcal{B}) \otimes_g Id_B + Id_A \otimes_g B(\mathcal{A}),$$

where Id_M is the identity matrix the size of which is equal to the size of matrix M. This generalization of tensor product and tensor sum allows to build the transition rate matrix of continuous-time SANs [2], queueing networks or other components-based models with transition rates or probabilities which may be functional. Note that the generalized tensor product is still well defined and the tensor representation still valid if the matrices are infinite. The generalized tensor product and sums have many algebraic properties such as Associativity and Distributivity (see [2] for a list of properties and some proofs). A model is described by the set of functional matrices. We now build the functional dependency graph. It is a directed graph the nodes of which are the components. A directed edge (x, y) in the functional dependency graph represents that component x uses the state of component y to compute some of its transition probabilities. We assume that the functional dependency graphs do not contain self-loops. The main result we obtain requires that the functional dependency graph is a Directed Acyclic Graph.

We now present some lemmas and properties to show the difference between the usual tensor product ($\otimes$) and the functional tensor product ($\otimes_g$).

Property 1 (**Compatibility with matrix multiplication**) *For all matrices A and B and vectors* π_A *and* π_B *whose sizes are consistent:*

$$(\pi_A \otimes \pi_B) \times (A \otimes B) = (\pi_A \times A) \otimes (\pi_B \times B).$$

where $\times$ *is the ordinary product (see [2]).*

Property 2 *Let* π_A *be an eigenvector of A associated with eigenvalue* λ_A *and* π_B *an eigenvector of B associated with eigenvalue* λ_B, *then* $\pi_A \otimes \pi_B$ *is an eigenvector of* $A \otimes B$ *with eigenvalue* $\lambda_A \lambda_B$.

Unfortunately these properties do not hold in general when we use the generalized tensor product. However it is possible to prove some results with some assumptions on the functional dependency graph.

Property 3 *Let B be a positive matrix, let* $A(\mathcal{B})$ *be a matrix whose elements are functions of the index of B. Assume that w is an eigenvector of B with eigenvalue* λ. *Assume that for all states s of B,* $A(s)$ *has an eigenvector v associated with eigenvalue*

μ. Then we have:

$$(v \otimes w) \times \left(A(\mathcal{B}) \otimes_g B\right) = \lambda\mu\,(v \otimes w)\,.$$

3 The Model and Its Solution

We study a modulated network of queues. We consider a continuous time Markov chain P denoted as the phase. We assume that this chain is finite and ergodic. The network of queues consists in n queues and this number of queues does not change with the phase. The service times are assumed to be exponential and arrival processes assumed to be Poisson. We assume also that the buffers are infinite and that each queue is associated with one server.

The arrival process changes according to a phase process. The main idea is that the network control reacts using service rates and routing of packets to minimise the power consumption. More precisely, we assume that the main objective of the service rates and routing modifications is to keep the load constant in all queues when the traffic rates change with the modulation. We do not claim that the controller is able to find the optimal solution for all traffic patterns. This is an open question that we will address in a sequel of that paper. When the phase does not change, the networks have a product-form solution at equilibrium with a geometric marginal distribution. However the ratios ρ_i of these distributions are not given by the same flow equation. Let us now define the model more precisely.

In phase ϕ, the arrival rate at queue i is λ_i^ϕ, the routing matrix is P^ϕ, the service rate is μ_i^ϕ. Let $M(\phi)$ be the Markov chain of the network of queues during phase ϕ. We assume that all chains $M(\phi)$ are ergodic. Let π^ϕ be the steady-state distribution of CTMC $M(\phi)$. It is clear that the invariant distribution of $M[\phi]$ is a product of geometric distributions with rate ρ_i^ϕ, solutions of the fixed point equations:

$$\rho_i^\phi = \frac{\lambda_i^\phi + \sum_{i=1}^n \mu_j^\phi \rho_j^\phi P^\phi(j,i)}{\mu_i^\phi}.$$

Indeed, matrix $M(\phi)$ represent a Jackson network. Therefore,

$$\pi^\phi(x1, \ldots, x_n) = \oplus_{i=1}^n g_i^\phi(x_i),$$

where $g_i^\phi(x_i) = (1-\rho_i^\phi)(\rho_i^\phi)^{x_i}$. Moreover by construction, we have $\pi^\phi M(\phi) = 0$ for all phase ϕ. Note also that for all phase ϕ, the solution is fully characterised by vector (ρ_i^ϕ) indexed by the queue number i. Let us now build the CTMC of the global model using the generalised tensor representation introduced by Plateau.

Property 4 *The CTMC of the model is defined by $P \bigoplus_g M(\mathcal{P})$ and the functional dependency graph is a DAG.*

Proof First, we have a functional dependency because all the matrices $M(\phi)$ depends on the state of the modulating chain P. Following Plateau [2], this is denoted as $M(\mathcal{P})$. We have a generalised tensor sum. Finally the functional dependency graph has two nodes: one (say p) models the phase and the other one (say n) represents the network of queues. As the transitions rates of the network are function of the phase, we have a directed edge from p to n. But the transitions of the phase are independent of the state of the network. Therefore we do not have a directed edge form n to p. Clearly the graph is a DAG.

Theorem 1 *Let α be the steady-state distribution of matrix P. If the vector (ρ_i^{ϕ}) is the same for all phase ϕ, then the model of the modulated network of queues has a product form steady-state distribution π:*

$$\pi = \alpha \otimes \pi^{\phi 0},$$

where $\phi 0$ is an arbitrary phase (as they all give the same vector π^{ϕ}). It is worthy to remark that a product form solution is the tensor product of smaller vectors. This explains why the algebraic representation leads to the solution.

Proof The proof relies on simple properties of the generalised tensor. We must prove that $(\alpha \otimes \pi^{\phi 0}) \cdot (P \bigoplus_g M(\mathcal{P})) = 0$. Remember that the operation "$\cdot$" is the usual vector-matrix product. Let Id be the identity matrix. We first describe the tensor sum as a tensor product and we use the distributivity:

$$(\alpha \otimes \pi^{\phi 0}) \cdot (P \bigoplus_g M(\mathcal{P})) = (\alpha \otimes \pi^{\phi 0}) \cdot (P \otimes Id) + (\alpha \otimes \pi^{\phi 0}) \cdot (Id \bigotimes_g M(\mathcal{P}))$$

We now show using the compatibility with ordinary product and Property 3 that both terms are equal to 0. First, consider the product: $(\alpha \otimes \pi^{\phi 0}) \cdot (P \otimes Id)$. Due to the compatibility with ordinary product we have:

$$(\alpha \otimes \pi^{\phi 0}) \cdot (P \otimes Id) = (\alpha \cdot P) \otimes (\pi^{\phi 0} \cdot Id) = 0 \otimes \pi^{\phi 0} = 0.$$

Indeed $(\alpha \cdot P) = 0$ as α is the steady-state distribution of the CTMC associated with P.

Now consider $(\alpha \otimes \pi^{\phi 0}) \cdot (Id \bigotimes_g M(\mathcal{P}))$. We can use Property 3 with $\lambda = 1$ associated to matrix Id and $\mu = 0$ for the matrices $M(\phi)$. Indeed, $\pi^{\phi 0} \cdot M(\phi) = 0$ for all phase ϕ because $\pi^{\phi 0} = \pi^{\phi}$. Therefore:

$$(\alpha \otimes \pi^{\phi 0}) \cdot (Id \bigotimes_g M(\mathcal{P})) = (\alpha \cdot Id) \otimes 0 = 0.$$

This concludes the proof. □

Property 5 *If the topology of the network is a DAG, it is sufficient to analyse the queues according to the topological ordering associated with the DAG to find an admissible solution for the service rates and the routing matrix.*

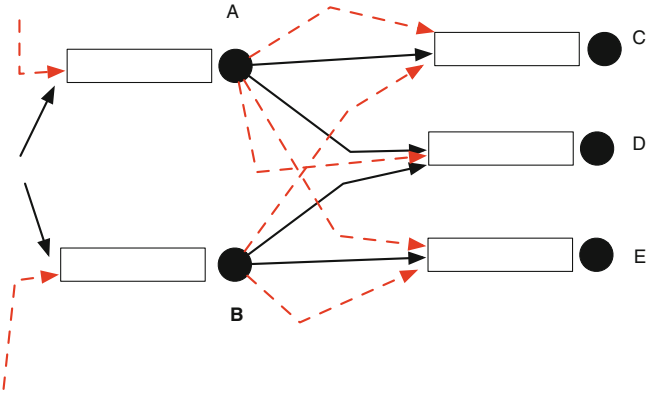


Fig. 1 A simple network of queues modelling a service center

The proof is omitted due to lack of space but the example below shows how the method works in that case. When we have a general topology with directed loops in the network, the problem is still open to find a simple algorithm.

3.1 A Toy Problem

We consider a very simple network of queues to illustrate the approach (see Fig. 1). We assume for the sake of readability that the phase has only two states b and r. The arrival rate of packets $\lambda_i^{(\phi)}$ at queue i depends on phase ϕ and the routing of customers as well. The network consist in 5 stations. The objective of the control is to keep the load in each queue constant. The controller may change the service rate of customers and the routing matrix $P^{(\phi)}$ to reach this objective.

During phase r, we use the routes depicted as red dotted lines in Fig. 1. Similarly the routes used in phase b are drawn as black straight lines in the same figure. We assume that during phase b the parameters are: $\lambda_A^{(b)} = 1$, $\lambda_B^{(b)} = 4$ for the arrival rates and the routing matrix is:

$$P^{(b)} = \begin{pmatrix} 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1/3 & 2/3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

where the queues are ordered in the alphabetical ordering to index the routing matrix. The customers after service completion at queues C, D, and E leave the network. The services rates are fixed by the controller and are equal to $(2, 8, 1, 22/6, 16/3)$ for queues A to E. Thus all the queues have a load equal to 0.5 and we do not have a bottleneck in the network. Note that this is not required by our theorem, we only need that the distributions are kept unchanged when we change the phase. However using

the same load for all the queues helps to simplify the presentation of the numerical results. Now assume that the phase changes to r. The arrival rates are now equal to $\lambda_A^{(r)} = 2$, $\lambda_B^{(r)} = 1$ and the controller has to react changing the services rates and the routing matrix to keep the same loads and the same marginal distributions. First, one has to change the service rate in queue A to have the same load. Thus the service rate μ_A becomes 4. Similarly, $\mu_B = 2$. Let us now turn to the routing matrix.

$$P^{(r)} = \begin{pmatrix} 0 & 0 & 1/10 & 11/20 & 7/20 \\ 0 & 0 & 1/10 & 0 & 9/10 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

We can now easily compute the intensity of the flows entering stations C, D and E. We found that the flow entering station C has intensity $3/10$. With a similar computation, the flow entering D (resp. E) has an intensity of $11/10$ (resp. $8/5$). The controller now gives a smaller service rate for stations C to E. They are respectively $\mu_C = 3/20$, $\mu_D = 11/20$, and $\mu_E = 4/5$. Thus the load is 0.5 for all stations in the network. Despite the traffic modification it is possible to make the controller obtain the same load for every queue. Thus the product form solution applies.

This simple strategy may be used to reduce the energy consumption when the arrival rates decrease and it has a closed-form solution. We hope that it may be possible to solve a more sophisticated optimisation problem using our analytical result as a basic method or as a bound.

References

1. Berl, A., Gelenbe, E., Girolamo, M.D., Giuliani, G., de Meer, H., Quan, D.M., Pentikousis, K.: Energy-efficient cloud computing. Comput. J. **53**(7), 1045–1051 (2010)
2. Fernandes, P., Plateau, B., Stewart, W.J.: Efficient descriptor-vector multiplications in Stochastic Automata Networks. J. Acm **45**(3), 381–414 (1998)
3. Fourneau, J.-M., Plateau, B., Stewart, W.: Product form for stochastic automata networks. In: Glynn, P.W. (ed.) Proceedings of the 2nd International Conference on Performance Evaluation Methodolgies and Tools, Valuetools 2007. Icst, Nantes (2007)
4. Fourneau, J.-M., Plateau, B., Stewart, W.: An algebraic condition for product form in stochastic automata networks without synchronizations. Perform. Eval. **85**, 854–868 (2008)
5. Fourneau, J.-M., Verchère, D.: G-réseaux dans un environnement aléatoire. RAIRO-Oper. Res. **34**, 427–448 (2000)
6. Greenberg, A.G., Hamilton, J.R., Maltz, D.A., Patel, P.: The cost of a cloud: research problems in data center networks. Comput. Commun. Rev. **39**(1), 68–73 (2009)
7. Mitrani, I.: Service center trade-offs between customer impatience and power consumption. Perform. Eval. **68**(11), 1222–1231 (2011)
8. Stoyan, D.: Comparaison Methods for Queues and Other Stochastic Models. Wiley, Berlin (1983)
9. Zhu, Y.: Markovian queueing networks in a random environment. Oper. Res. Lett. **15**, 11–17 (1994)

Part IV
Data Analysis

Statistical Tests Using Hinge/ϵ-Sensitive Loss

Olcay Taner Yıldız and Ethem Alpaydın

Abstract Statistical tests used in the literature to compare algorithms use the misclassification error which is based on the 0/1 loss and square loss for regression. Kernel-based, support vector machine classifiers (regressors) however are trained to minimize the hinge (ϵ-sensitive) loss and hence they should not be assessed or compared in terms of the 0/1 (square loss) but with the loss measure they are trained to minimize. We discuss how the paired t test can use the hinge (ϵ-sensitive) loss and show in our experiments that doing that, we can detect differences that the test on error cannot detect, indicating higher power in distinguishing between the behavior of kernel-based classifiers (regressors). Such tests can be generalized to compare $L > 2$ algorithms.

1 Introduction

Statistical tests in the literature, namely, paired t test, 5×2 cv t test [1], 5×2 cv F test [2], are based on the misclassification error which corresponds to 0/1 loss. Support vector machine classifiers [3] are trained to minimize the *hinge loss* which not only checks whether the decision is on the right side of the boundary but also its position in the margin. Let us say $f(x^t) \in \Re$ is the kernel classifier output for input

O. T. Yıldız (✉)
Department of Computer Engineering, Işık University, TR-34980
Istanbul, Turkey
e-mail: olcaytaner@isikun.edu.tr

E. Alpaydın
Department of Computer Engineering, Boğaziçi University, TR-34342
Istanbul, Turkey
e-mail: alpaydin@boun.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_16,

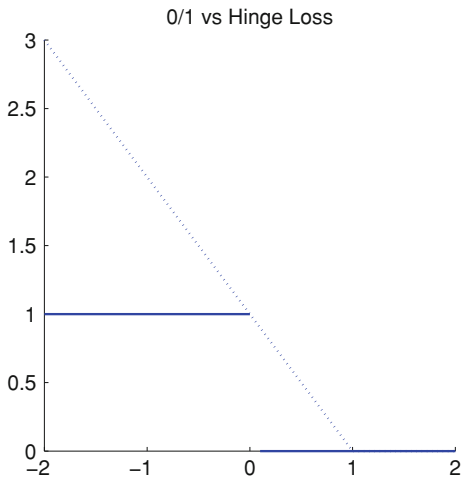


Fig. 1 0/1 versus hinge loss as a function of $f(x^t)$ for $y^t = 1$

x^t and $y^t \in \{-1, +1\}$ is the desired output, 0/1 and hinge loss are defined as (Fig. 1):

$$0/1 \text{ loss} = \begin{cases} 0 & \text{if } f(x^t)y^t \geq 1 \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

$$\text{hinge loss} = \begin{cases} 0 & \text{if } f(x^t)y^t \geq 1 \\ 1 - f(x^t)y^t & \text{otherwise} \end{cases} \tag{2}$$

Misclassification error only checks whether the classifier output is on the correct side of the boundary; hinge loss differs in two respects: (1) It also penalizes slightly those instances that are on the correct side but are in the *margin*, that is, to classified with enough confidence, and (2) the misclassified instances are penalized linearly proportional to how deep they are in the wrong side.

Most of the regression algorithms are trained to minimize the square loss. In support vector regression, we use the ϵ-sensitive loss which tolerate errors up to ϵ and redefines the margin as the ϵ-tube. Let us say $f(x^t) \in \Re$ is the support vector regression output for input x^t and $y^t \in \Re\}$ is the desired output, square and ϵ-loss are defined as (Fig. 2):

$$\text{square loss} = |y^t - f(x^t)|^2 \tag{3}$$

$$\epsilon\text{-sensitive loss} = \begin{cases} 0 & \text{if } |y^t - f(x^t)| \leq \epsilon \\ y^t - f(x^t)| - \epsilon & \text{otherwise} \end{cases} \tag{4}$$

ϵ-sensitive loss differs from square loss in two respects: (1) it does not penalize those instances that are in the ϵ-tune, and, (2) the instances are penalized linearly proportional to how much they are far from the correct output. Therefore, ϵ-sensitive loss is more tolerant to noisy instances and thus more robust than the square loss.

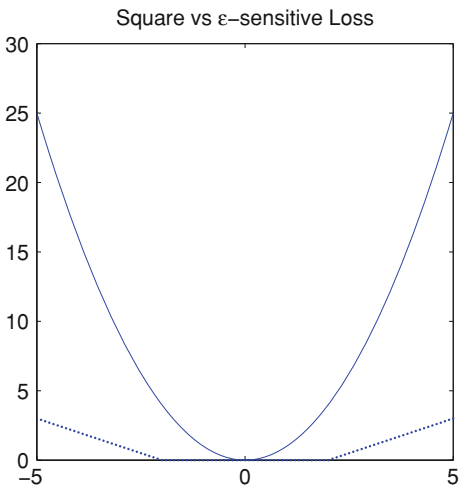


Fig. 2 Square versus ϵ-sensitive loss

Taking the losses used in training into account while comparing the test performance of kernel algorithms would thus enable to better distinguish between their generalization behavior. The two kernel algorithms compared may be using two different kernels or their kernels may be using two different sources of input, etc. and we want to check if there is a significant difference between them, for example, to test whether a new proposed kernel leads to improvement.

In statistical testing, we run both algorithms a number of times on a number of (training, validation) folds and compare the distributions of validation results for statistically significant difference; typically, k-fold cross-validation is used to generate k (training, validation) data set pairs by resampling from a single data set [4].

This paper is organized as follows: In Sect. 2, we discuss paired t test usually used for error comparison and discuss how it can also use hinge or ϵ-sensitive loss. We give our experimental results in Sect. 3 and conclude in Sect. 4.

2 Paired t Test for Comparison

Let us say for all folds, $j = 1, \ldots, k$, we train both algorithms on training fold j and test on validation fold j and obtain the performance value $x_{ij}, i = 1, 2$ where x_{ij} is the total loss on the validation set where loss can be calculated using any of Eqs. (1)–(4). It is important that all compared algorithms use the same training, validation data so that the comparison is *paired*. We want to check if the two sets of x_{1j} and x_{2j} can be said to come from the same population or whether they come from two distinct populations.

In the paired t test, we assume that the populations are normal and check if they have the same mean and for this, we test if their paired differences, $d_j = x_{1j} - x_{2j}$, have a mean of zero:

$$H_0 : \mu_d = 0 \text{ versus } H_1 : \mu_d \neq 0$$

We calculate the average and variance of paired differences

$$m = \sum_{j=1} d_j / k \ , \ s^2 = \sum_j (d_j - m)^2 / (k-1)$$

Under the null hypothesis, the statistic

$$t' = \frac{\sqrt{k} m}{s} \tag{5}$$

is t-distributed with $k-1$ degrees of freedom. We reject the null hypothesis that the two algorithms generalize equally well according to whichever loss we use if $|t'| > t_{\alpha/2,k-1}$ with $(1-\alpha)100\,\%$ confidence.

This test assumes that each x_{ij} is normally distributed. For any of Eqs. (1)–(4), loss on each validation set is independent and identically distributed (but not necessarily normal). From the central limit theorem, we know that when we sum these up, the total loss converges to the normal distribution (unless the data set is small) and hence the normality assumption is tenable. In our experiments, the validation sets are not small, and again using the central limit theorem, we can also claim normality for the hinge values. As we will report later on, for all losses, the samples are indeed found to be normally distributed when tested with a normality test experimentally.

In Fig. 3, we see a comparison on a synthetic two-dimensional classification data where we can see the discriminants and the margins. In this case of comparing the linear and Gaussian kernel, the test on error does not reject but the test on hinge loss finds a significant difference. As we see in Fig. 3a and b, the two kernels lead to discriminants which are not different in terms of the boundary but they have significantly different margins. In Fig. 3c, we see the histogram of paired differences whose expected value is not so far from 0 and hence, the paired t test on error does not reject the equality of means. If we similarly look at Fig. 3d, we see why the paired t test on hinge loss rejects; all paired differences are positive in Fig. 3d.

3 Experimental Results

We report our experiments on 11 data sets (*australian, breast, credit, cylinder, german, pima, mammographic, satellite47, tictactoe, titanic, transfusion*) for classification and 9 datasets (*abalone, add10, boston, california, concrete, puma8fh, puma8fm, puma8nh, puma8nm*) for regression from the UCI Repository. We used four different support vector machines with *linear, quadratic, cubic* and *Gaussian*

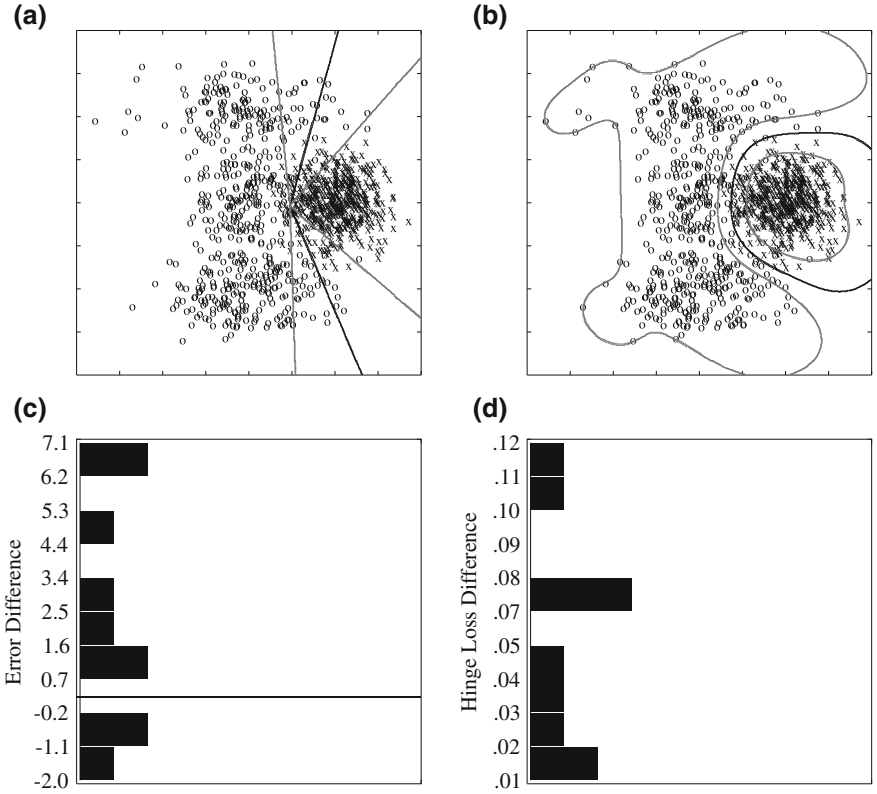


Fig. 3 Comparison of linear and Gaussian kernels on synthetic classification data (the nonlinearity with the linear kernel is due to normalization). **a** Linear kernel. **b** Gaussian kernel. **c** Error difference. **d** Hinge difference

kernels. All kernels are normalized for the discriminants to have the same scale. We use tenfold cv and set $\alpha = 0.05$.

To be able to use the paired t test on hinge or ϵ-sensitive loss, we need to make sure that the assumption of normality holds. For all kernels, we used (the univariate version of the) normality test [5] and counted the percentage of times that the test rejects that the sample comes from a normal population. On each data set, we repeated the tenfold experiment ten times and the values in Table 1 are hence proportions over $11 \times 10 = 110$ runs. As we see there, the percentage of rejects using the hinge loss compare well with the percentage of rejects using error, indicating that paired t test on hinge loss is as applicable as parametric tests on error. Actually it seems as if the kernel type is a more influential factor in the normality of results than the performance criterion.

On all classification data sets for all kernel types, we do pairwise comparisons using both error and hinge loss and compare the test results. For all 11 data sets and

Table 1 Percentage of rejects of normality for 0/1, hinge, square, and ϵ-sensitive losses using different kernels

Kernel	Loss measure			
	0/1	Hinge	Square	ϵ-sens.
Linear	0.136	0.109	0.000	0.000
Quadratic	0.009	0.018	0.078	0.067
Cubic	0.009	0.000	0.033	0.022
Gaussian	0.055	0.045	0.067	0.056

Table 2 Percentage of agreement/disagreement of 0/1 and hinge loss

0/1	Hinge	
	Accept	Reject
Accept	26.4	33.6
Reject	6.7	33.3

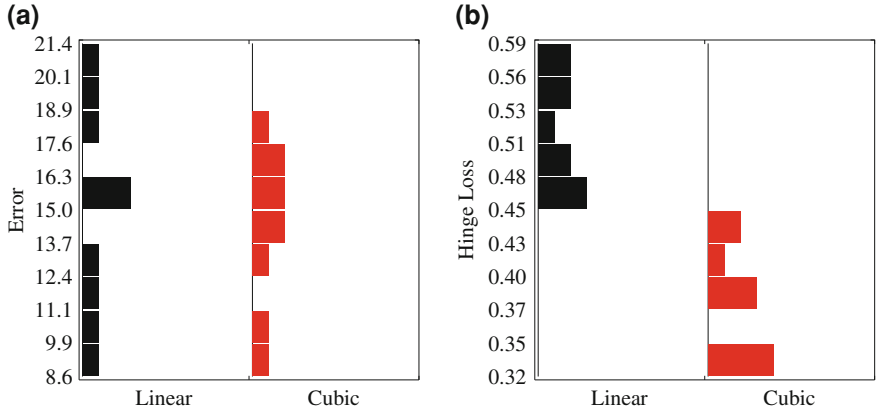


Fig. 4 On *credit* data set, results of comparison of linear and cubic kernels. Paired t test on error does not reject whereas test on hinge loss rejects. **a** 0/1 error. **b** Hinge histograms

for ten independent runs for each, for all $4 \times 3/2 = 6$ pairwise comparison of four kernels, we have a total of 660 comparisons and the values reported are percentages. In Table 2, we see that the paired t tests on error and hinge loss agree in their decisions in $26.4 + 33.3 = 59.7\,\%$ of the cases. When they disagree, in $33.6\,\%$ of the cases, the test on hinge loss finds a significant difference and rejects whereas the test on error considers them comparable; the opposite occurs in $6.7\,\%$ of the cases. This shows that in around one-third of the cases, there is a difference between classifiers in terms of hinge loss and this difference information is lost if 0/1 error is used.

As an example where tests on error and hinge loss disagree, in Fig. 4, we compare linear and cubic kernels on the *credit* data set. We see that though the classifiers are not significantly different in terms of error, they are significantly different in terms of hinge loss. The paired t test does not reject in terms of error because as we see in Fig. 4a, the two error distributions overlap whereas they do not overlap in terms of

Table 3 Percentage of agreement/disagreement of the paired t test on square and ϵ-sensitive loss

Square	ϵ-sensitive	
	Accept	Reject
Accept	11.5	2.4
Reject	3.0	83.1

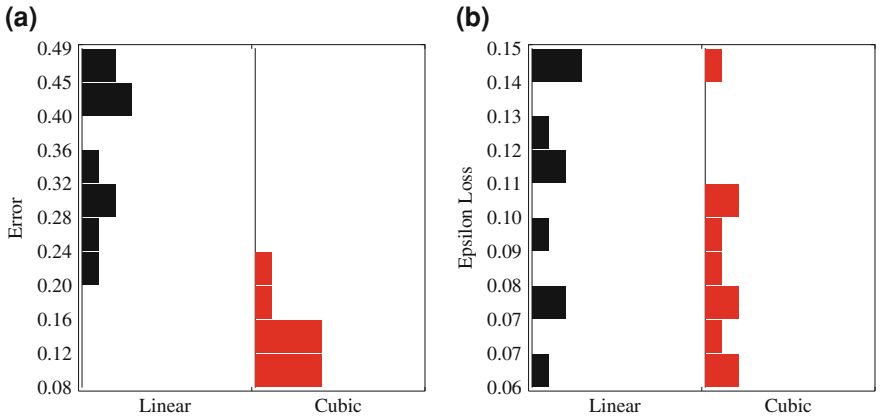


Fig. 5 On *boston* data set, results of comparison of linear and cubic kernels. Paired t test on squared loss rejects whereas test on ϵ-sensitive loss does not reject. **a** Square loss. **b** ϵ-sensitive histograms

hinge loss, as we see in Fig. 4b, and the paired t test on hinge loss rejects the null hypothesis of equality of means.

On all regression data sets for four polynomial kernel types, we did pairwise comparisons using both square and ϵ-sensitive loss and compare the test results. For all 9 data sets and for ten independent runs for each, for all $4 \times 3/2 = 6$ pairwise comparison of three kernels, we have a total of 540 comparisons and the values reported are percentages. In Table 3, we see that the paired t tests on square and ϵ-sensitive loss agree in their decisions in $11.5+83.1 = 94.6\,\%$ of the cases. Contrary to the classification case, they disagree only in $5.4\,\%$ of the cases, indicating that using the paired t test, we can use ϵ-sensitive loss as well as the square loss.

As an example where tests on square and ϵ-sensitive loss disagree, in Fig. 5, we compare linear and cubic kernels on the *boston* data set. We see that though the classifiers are significantly different in terms of square loss, they are not significantly different in terms of ϵ-sensitive loss. The paired t test rejects in terms of square loss because as we see in Fig. 5a, the two error distributions do not overlap whereas they overlap in terms of ϵ-sensitive loss, as we see in Fig. 5b, and the paired t test on ϵ-sensitive loss does not reject the null hypothesis of equality of means. For this case, square loss is sensitive to outliers (due to quadratic increase) and therefore rejects the null hypothesis whereas ϵ-sensitive loss is more robust to outliers and fails to reject the null hypothesis.

4 Conclusions and Future Work

Kernel-based, support vector machine classifiers and regressors are trained to minimize the hinge loss and ϵ-sensitive loss respectively. Hence their assessment and comparison should be done using the same measure and not misclassification error or square error. The hinge loss differs from the 0/1 loss and is a more informative measure because (1) it penalizes those instances in the margin—0/1 loss would not penalize them but hinge loss does because they are not classified with enough confidence, and (2) the misclassified instances on the wrong side of the boundary have equal loss of 1 with 0/1 loss, whereas the hinge loss penalizes them linearly proportional to their distance to the boundary thereby taking into account the confidence of the classifier in its decision. Taking these two into account gives more information about the confidence of the underlying classifier and allows us to distinguish between classifiers which are indistinguishable in terms of error. Indeed as we see in our experiments, statistical tests on hinge loss allow finding differences where the tests on error find no significant difference.

A similar rationale can also be put forward for favoring ϵ-sensitive loss over square loss: (1) It tolerates small, insignificant errors and (2) loss increases linearly as opposed to quadratically hence is more robust to outliers.

Here, we only discuss pairwise tests to compare two algorithms, but hinge and ϵ-sensitive loss can also be used to compare $L > 2$ algorithms, for example, to compare L different kernels. Analysis of variance (ANOVA) can be used to test the equality of the means of $L > 2$ populations. There are also tests that can be used to find cliques of algorithms such that no pairwise test rejects between any two in the clique or ordering them [6]; such tests can also use the hinge loss or ϵ-sensitive loss instead of error. These are possible future directions for research.

Acknowledgments This work is supported by TÜBİTAK EEEAG 109E186 and BAP 5701.

References

1. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning classifiers. Neural Comput. **10**, 1895–1923 (1998)
2. Alpaydın, E.: Combined 5 × 2 cv F test for comparing supervised classification learning classifiers. Neural Comput. **11**, 1975–1982 (1999)
3. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
4. Alpaydın, E.: Introduction to Machine Learning, 2nd edn. The MIT Press, Cambridge (2010)
5. Mardia, K.V.: Measures of multivariate skewness and kurtosis with applications. Biometrika **57**, 519–530 (1970)
6. Yıldız, O.T., Alpaydın, E.: Ordering and finding the best of $K > 2$ supervised learning algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **28**(3), 392–402 (2006)

Self-Adaptive Negative Selection Using Local Outlier Factor

Zafer Ataser and Ferda N. Alpaslan

Abstract Negative selection algorithm (NSA) classifies a given data either as normal (self) or anomalous (non-self). To make this classification, it is trained using normal (self) samples. NSA generates detectors to cover the complementary space of self in training phase. The classification of NSAs is mainly specified by two issues, self space determination and detectors coverage. The boundary of self is ambiguous so NSAs use self samples to calculate a space close to the self space. The other issue is the detectors coverage which should maximize non-self space coverage and minimize self space coverage. This paper introduces a novel NSA and this NSA proposes *k*-nearest neighbor and local outlier factor to determine self space for a given self samples. Beside these, it specifies the detectors coverage using Monte Carlo Integration. The experimental evaluations show that the novel NSA generates comparable and reasonable results.

1 Introduction

The biological mechanisms have always been the main source for the development of computer intelligence. The biological immune system is one of them and it was used to emerge the artificial immune systems (AIS). Inspired from the different biological immunity theories, many various models of AIS have been developed. The most known AIS models are negative selection algorithm (NSA), immune network and clonal selection.

Z. Ataser (✉) · F. N. Alpaslan
Department of Computer Engineering, Middle East Technical University,
Ankara, Turkey
e-mail: zafer.ataser@ceng.metu.edu.tr

F. N. Alpaslan
e-mail: alpaslan@ceng.metu.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_17,

The major mechanism of biological immunity is the discrimination of self (normal) and non-self (anomalous) entities. This discrimination task is performed by T-cells. T-cells are produced and self-reacting cells are destroyed in thymus. After this maturing process, T-cells are released. Inspired from this T-cells generation process, NSAs were developed in AIS. NSAs generate detectors instead of T-cells and eliminate detectors which match self [1].

NSAs consist of two processes, detector generation and detection respectively. In the first step, a detector is generated randomly and then it is checked whether it matches any normal sample or not. If it matches, it is discarded; otherwise it is added to the detectors set. This generation process is repeated until the detectors coverage reaches a sufficient level. In the detection process, a given data is classified as normal or anomalous using the generated detectors. Hence, it can be easily said that NSAs are proper for two class classification problems such as fault detection, and anomaly detection [2]. Although, there are many fields that use NSA, the primary application of NSA have been anomaly detection.

The critical issues that affect the classification accuracy of NSAs are correctness of self space determination and maximizing non-self space coverage by detectors. The main concern of this study is optimizing the classification correctness by improving self space determination and coverage of detectors. Self space is calculated using k-nearest neighbor and a density based outlier method, local outlier factor (LOF). Thus, self space is also determined by the self density and it is expected to provide self adaptability. Beside this, the new NSA also tries to maximize non-self space coverage using Monte Carlo Integration method.

The rest of the paper is organized as follows: the background information about NSAs is given in Sect. 2. Section 3 introduces the proposed NSA. The experimental results and evaluations are presented in Sect. 4. The concluding remarks about the proposed NSA are given in Sect. 5.

2 Background

Negative selection algorithms (NSAs) classify a given data either as normal or anomalous and raise an alarm for an anomalous case. In order to mention the classification correctness of NSAs, positive and negative concepts should be defined. A positive means the detection of anomalous (non-self) while a negative means the detection of normal (self). The positive and negative are categorized into two groups; a true defines correct detection and a false defines incorrect detection. Figure 1 illustrates the general definitions of classification and NSA. In Fig. 1, detected self space consists of the circles which are defined by self samples and a constant self radius parameter.

There are two main factors that affect classification accuracy; coverage of detectors and self space determination. NSAs try to maximize non-self coverage of detectors while they minimize self space coverage. In order to do that, first self

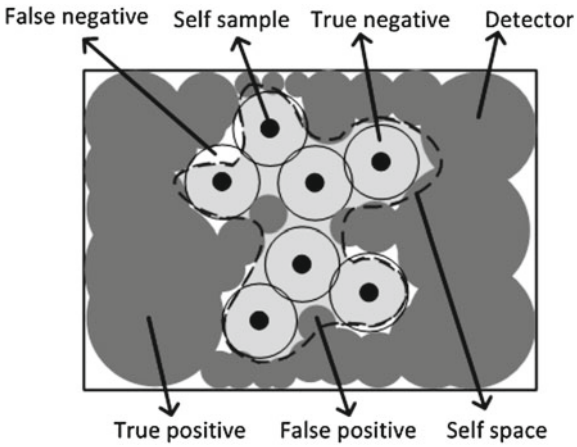


Fig. 1 General definitions for classification and NSA

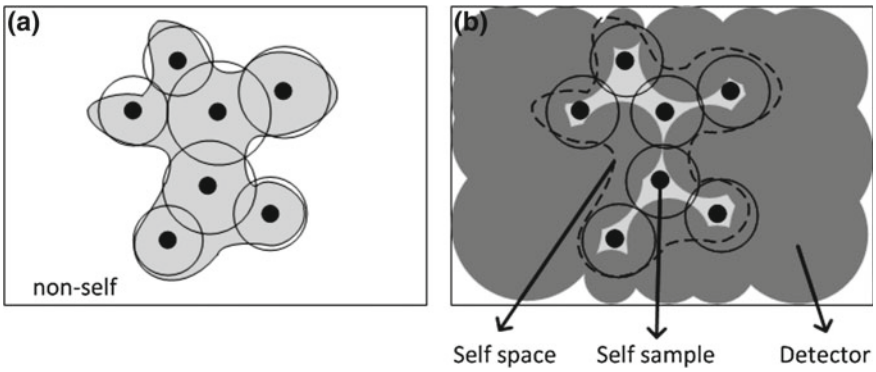


Fig. 2 **a** Variable self radius. **b** Boundary-aware

space should be determined using self samples. Most of NSAs take a radius parameter for all samples and calculate self space using it [3, 4].

To find more accurate self space for given self samples, recent developed NSAs calculate each self sample radius using probabilistic techniques [5, 6]. Thus, a radius of each self sample is specified by the given samples' features and this provides self adaptability to NSAs. Figure 2a presents the variable self radii which give opportunity to optimize the maximization of self space coverage and the minimization of non-self space coverage.

The recent most known NSA, *V-detector*, develops a boundary-aware method to specify the self space using self samples [7, 8]. This method takes self radius (r_s) as a parameter and uses it in detector generation phase. A generated detector is checked whether the distance between its center and a nearest self sample is greater than r_s.

If not, then the generated detector is discarded. Otherwise, the distance is assigned to the radius of the generated detector. Figure 2b illustrates the detected self space and generated detectors by *V-detector* with boundary-aware method.

3 Self-Adaptive NSA

This section introduces the enhanced NSA, self-adaptive negative selection (SANS), in three subsections. In first subsection, problem domain is defined. The second subsection presents the new "self space" definition and the last subsection explains detector generation process.

3.1 Anomaly Detection

The purpose of anomaly detection is to classify a given state as normal or anomalous by forming a profile of the normal state of a system. A set of features can be used to represent the state of a system based on the work by Dasgupta and Gonzalez [3].

Definition 1. (System state space). A vector of features, $x^i = (x_1^i, \ldots, x_n^i)$, represents a state of the system and each feature is normalized to [0.0, 1.0]. The state space is represented by $U \subseteq [0.0, 1.0]^n$ for n dimensional space, where n denotes the number of features. Elements of the set are the features vectors which correspond to all possible states of the system.

Definition 2. (Normal subspace). A normal state of the system is represented by a set of feature vectors, Self U. A complement of this set gives anomalous states of the system, Non-Self = U-Self. The Self (Non-Self) set can be defined by its characteristic function x_{self}: $[0.0, 1.0]^n \rightarrow \{0, 1\}$.

$$x_{self} = \begin{cases} 1 & \text{if x} \in \text{Self} \\ 0 & \text{if x} \in \text{Non-Self} \end{cases} \tag{1}$$

3.2 Self Space

NSAs generate detectors to cover the complementary space of self. The self space is defined by the self set, S, but NSAs have only self samples, $S' \subseteq S$, to find the self space. Therefore, the self set, S, can be modeled as a set $\hat{S}$, which is defined in terms of a set of self samples, S'. There is an assumption in this model; if an element is close enough to a self sample, then it is considered as self [9]. The closeness is determined by a radius of a self sample, r_{self}^i. Each self sample has its radius

which can be different from the others. The self radius, r^i_{self}, specifies the maximum distance between an element and the self sample, s^i. Based on these definitions, a set $\hat{S}$ can be given as follows:

$$\hat{S} = \{x \in U | \exists s^i \in S', \|s^i - x\| \leq r^i_{self}\} \tag{2}$$

The critical issue for a set $\hat{S}$ is the radii of the self samples. A self radius is calculated to include combination of global and local features. Thus, the drawbacks of globality and locality can be minimized.

The global feature is the average distance of kth nearest neighbor, μ_{knn}. The distance of given kth nearest neighbor is computed for each sample and the average of these distance are calculated.

In addition to the global feature, a density based outlier method, local outlier factor, is selected to add a local feature to the self radii. Local outlier factor (LOF) method computes a local outlier value for each self sample which indicates its degree of outlier-ness [10]. LOF values of the self samples are normalized to [0.0, 1.0] using maximum LOF value.

$$r^i_{self} = \mu_{knn} \times LOF^i \tag{3}$$

LOF value adds variability to the self radii and this provides opportunity to avoid the drawbacks of the constant self radius usage.

3.3 Detectors

NSAs generate a set of detectors that covers non-self space after the self space determination. The detectors set should be defined before the generation process. Elements of the detectors set are represented by variable size circles, so a detector is defined as a center of circle, d^j, and a radius, r^j. The detectors set, D, can be defined as follows:

$$D = \{d^j \in U | \forall s^i \in S', \|s^i - d^j\| \leq (r^i_{self} + r^j)\} \tag{4}$$

The other critical issue for detectors is the definition of the detector generation process. To speed up detectors generation with high non-self coverage, Monte Carlo Integration is modified and used as detector generation algorithm. Monte Carlo Integration is mainly applied to compute the value of complex integrals using probabilistic techniques. In other word, it is proper to calculate the area of the complex shapes. Therefore, it was used to estimate self space in [9]. In this work, Monte Carlo Integration checks a randomly generated data whether it close enough to self samples. If it is, hit number is increased by one. This process is repeated until reaching enough estimation precision. The estimated area is calculated as dividing the hit by the number of iteration. Inspired from this work, SANS also uses Monte Carlo Integration to estimate non-self space coverage of detectors.

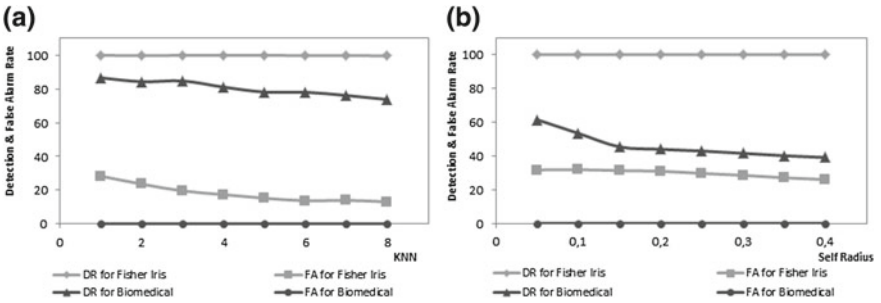


Fig. 3 **a** Outputs generated by SANS. **b** Outputs generated by V-detector

This estimated value is compared with the predefined coverage percentage in the termination condition.

4 Experiments and Evaluations

In order to find out the advantages and disadvantages of SANS, experiments were performed with Fisher Iris and Biomedical datasets. The experimental results were compared with those of the recent known NSA, *V-detector*. *V-detector* is a real-valued negative selection algorithm that specifies self space using boundary-aware approach and represents detectors as variable size circles. Naive estimation option of *V-detector* is disabled in detectors generation process to obtain better classification performance.

The first dataset, Fisher Iris, contains three different species of Iris flowers, setosa, versicolor and virginica. Each sample consists of four attributes, the width and the length of sepal and petal. This dataset was converted into two dimensional one by taking sum of the length and the width of the sepal and petal.

The second dataset, biomedical dataset, includes blood measurement which was used to screen a genetic disorder. The blood measurements of 127 normal patients were selected as training data. All the blood measurements of 127 normal patients and 75 carriers were used as test data. Each patient has four types of blood measurement but in experiments, training and test data consists of the second type and the sum of the third and the fourth types.

Table 1 shows the results and comparisons using Fisher Iris dataset. All results are computed as the average of 100 consequent runs. In each experiment, one of the three species was taken as normal and the other two were accepted as anomalous.

The comparisons were based on the classification accuracy which is specified by detection rate and the false alarm rate. High detection rate and low false alarm rate is expected from a successful NSA. Figure 3a presents the detection rate and the false alarm rate of SANS for Fisher Iris and biomedical datasets separately. In Fisher Iris,

Table 1 Comparison between SANS and V-detector using Fisher's Iris dataset

Training data	Algorithm	Self radius/ KNN	Detection rate (%)	False alarm rate (%)	Number of detectors
Setose (50 %)	V-detector	0.1	99.98	23.82	500
		0.05	100	26.42	500
	SANS	4	100	18.58	38.76
		3	100	22.72	36.82
Setose (100 %)	V-detector	0.1	100	0	500
		0.05	100	0	500
	SANS	4	100	0	74.31
		3	100	0	86.71
Versicolor (50 %)	V-detector	0.1	100	31.7	500
		0.05	100	31.84	500
	SANS	4	99.96	17.36	48.39
		3	100	19.58	49.12
Versicolor (100 %)	V-detector	0.1	97	0	500
		0.05	96.99	0	500
	SANS	4	93	0	81.78
		3	93.8	0	90.38
Virginica (50 %)	V-detector	0.1	100	33.72	500
		0.05	100	35.94	500
	SANS	4	98.66	11.48	72.31
		3	99	13.41	71.03
Virginica (100 %)	V-detector	0.1	98	0	500
		0.05	98	0	500
	SANS	4	97.98	0	92.4
		3	98	0	97.68

training data were a half of versicolor species samples and test data were the all samples of three species. On the other hand, in the second experiment case, all of the normal samples of biomedical dataset were accepted as training data and test data were all of the normal and anomalous samples. Figure 3b shows the detection rate and the false alarm rate of *V-detector* for the same datasets.

SANS and *V-detector* have almost the same detection rates in Table 1. SANS and V-detector have the same false alarm rate for training data selected as 100 % of the normal samples. On the other side, SANS has lower false alarm rate for the training data selected as 50 % of the normal samples. The same inference can be also made from Fig. 3a,b for the training dataset as 50 % of versicolor normal samples.

Table 2 illustrates the results of SANS and *V-detector* using a biomedical dataset. The detection rate of *V-detector* is slightly better than SANS for the training data selected as 25 and 50 % of biomedical normal samples. On the other hand, the detection rate of SANS is much better than *V-detector* for 100 % of biomedical normal samples. Figure 3a,b also present this comparison clearly. There is only one

Table 2 Comparison between SANS and V-detector using biomedical dataset

Training data	Algorithm	Self radius/ KNN	Detection rate (%)	False alarm rate (%)	Number of detectors
25%	V-detector	0.1	84.94	20.03	500
		0.05	85.63	21.11	500
	SANS	4	82.14	10.39	68.29
		3	85.01	15.69	72.58
50%	V-detector	0.1	84.85	10.78	500
		0.05	85.36	12.12	500
	SANS	4	80.44	11.14	76.9
		3	83.84	13.1	80.29
100%	V-detector	0.1	53.41	0	500
		0.05	61.48	0	500
	SANS	4	81.17	0	174.66
		3	84.8	0	177.99

experiment case, 25 % of biomedical normal samples, that the false alarm rate of SANS is better. The false alarm rates of both NSAs are almost same for the other cases.

Overall experiment cases show that SANS generates the comparable results when training data compose of the all normal samples. They also indicate that SANS generates more reasonable results when training data include the partial of normal samples.

5 Conclusion

This paper introduces a novel NSA, self-adaptive negative selection (SANS). The classification performance of NSA is directly affected by self space determination and coverage of detectors. SANS uses kth nearest neighbor and local outlier factor to specify self space. These methods are useful to extract the global and the local features of the given self data. Beside, Monte Carlo Integration is modified to generate detectors and maximizing non-self coverage.

SANS and the recent known NSA, *V-detector*, were compared in the experiments. Although, there is not notable difference between detection rates of these NSAs, SANS has lower false alarm rate when the partial of normal samples is used as training data. The experimental results show that SANS is an appropriate method to build the normal profile of a system. Hence, this NSA can efficiently solve the anomaly detection problems.

References

1. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. In: IEEE Symposium on Research in Security and Privacy, pp. 202–212 (1994)
2. Al-Enezi, J., Abbod, M., Alsharhan, S.: Artifical immune systems—models, algorithms and application. Int. J. Res. Rev. Appl. Sci. **3**(2), 118–131 (2010)
3. Dasgupta, D., Gonzalez, F.: An immunity-based technique to characterize intrusions in computer networks. IEEE Trans. Evol. Comput. **6**(3), 1081–1088 (2002)
4. Balachandran, S., Dasgupta, D., Nino, F., Garrett, D.: A framework for evolving multi-shaped detectors in negative selection. In: IEEE Symposium on Foundations of Computational Intelligence, pp. 401–408 (2007)
5. Zeng, J., Liu, X., Li, T., Liu, C., Peng, L., Sun, F.: A self-adaptive negative selection algorithm used for anomaly detection. Prog. Nat. Sci. **19**, 261–266 (2009)
6. Yuel, X., Zhang, F., Xi, L., Wang, D.: Optimization of self set and detector generation base on real-value negative selection algorithm. Int. Conf. Comput. Comm. Technol. Agric. Eng. **2**, 12–15 (2010)
7. Ji, Z.: A boundary-aware negative selection algorithm. In: Proceedings of IASTED International Conference of Artificial Intelligence and Soft Computing, pp. 379–384 (2005)
8. Ji, Z., Dasgupta, D.: V-detector: An efficient negative selection algorithm with "probably adequate" detector coverage. Inform. Sci. **179**(10), 1390–1406 (2009)
9. Gonzalez, F., Dasgupta, D., Nino, L.F.: A randomized real-valued negative selection algorithm. In: Proceedings of the 2nd International Conference on Artificial Immune Systems, pp. 261–272 (2003)
10. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: Identifying density-based local outliers. In: ACM SIGMOD 2000 International Conference on Management of Data, pp. 93–104 (2000)

Posterior Probability Convergence of k-NN Classification and K-Means Clustering

Heysem Kaya, Olcay Kurşun and Fikret Gürgen

Abstract Centroid based clustering methods, such as K-Means, form *Voronoi cells* whose radii are inversely proportional to number of clusters, K, and the expectation of posterior probability distribution in the closest cluster is related to that of a k-Nearest Neighbor Classifier (k-NN) due to the Law of Large Numbers. The aim of this study is to examine the relationship of these two seemingly different concepts of clustering and classification, more specifically, the relationship between k of k-NN and K of K-Means. One specific application area of this correspondence is *local learning*. The study provides experimental convergence evidence and complexity analysis to address the relative advantages of two methods in local learning applications.

Keywords Clustering · K-Means · K-Medoids · K-NN classification · Local learning

1 Introduction

In machine learning literature, clustering is used for several purposes: (1) labeling data, (2) mapping data into a smaller space therefore extracting features for subsequent supervised learning, (3) sub-sampling via taking a representative object from

H. Kaya (✉) · F. Gürgen
Department of Computer Engineering, Bogazici University,
34342 Bebek, Istanbul, Turkey
e-mail: heysem@boun.edu.tr

F. Gürgen
e-mail: gurgen@boun.edu.tr

O. Kurşun
Department of Computer Engineering, Istanbul University,
34320 Avcilar, Istanbul, Turkey
e-mail: okursun@istanbul.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_18,

each cluster [1]. When it is used as a preprocessing for classification, the information loss can be minimized by representing a cluster not just by its index but with the class distribution it has. These posterior probabilities obtained from clustering can be used to reduce dimensionality and classifier complexity without compromising classification accuracy [2]. Due to the fact that medoid/centroid based clustering methods form *Voronoi cells* whose radii are inversely proportional to number of clusters, K,the expected value of posterior probability distribution is thought to be converging to that of a k-Nearest Neighbor Classifier (k-NN) due to the Law of Large Numbers. In k-NN, posterior probability $P(C_i|x) = k_i/k$ where k_i is the number of neighbors which belong to class i, C_i [1]. The aim of this study is to examine the relationship of centroid based clustering to the task of classification. More specifically, we examine the relationship of K of K-Means clustering to the k of k-Nearest Neighbor (k-NN) classifier. Two datasets from UCI Machine Learning repository [3] were used to investigate the relationship of predictions of these methods for both classification and regression tasks: (1) Car Evaluation Dataset for classification (2) Telemonitoring of Parkinson's Disease Dataset for regression. As expected, a strong inverse correlation was found between k and corresponding K values: around -0.95 for car evaluation dataset (classification problem) and around -0.84 for PD dataset (regression problem). The layout of this paper is as follows: In Sect. 2, the methods used in the study are explained; in Sect. 3 the findings of convergence analysis are provided; Sect. 4 presents conclusions.

2 Methods

The base learners used in this study is k-NN and K-Means which are commonly used in machine learning. The contribution of a recent study [2] is that posterior probabilities for validation set can be extracted from closest clusters. Furthermore these posteriors can serve as features i.e. can be stacked to other supervised learners. The main motivation of this study is to show that it is also possible to extract almost the same quality posteriors using k-NN with appropriate selection of k, therefore to allow a significant reduction in algorithmic complexity of feature extraction preprocess. On the other hand it is possible to create robust local models using K-Means instead of creating costly on-the-fly models with k-Nearest Neighbors.
A semi parametric density estimation method and a non-parametric method were compared with a simplification assumption of equal variance and no covariance in the dataset.

2.1 K-Means Clustering

K-Means clustering is a centroid based algorithm while the alternative K-Medoids is a medoid based algorithm which is known to be more resistant to outliers [4].

Preliminary study was held by k-Medoids however due to its more common use and simplicity, K-Means was used for convergence tests.

2.2 k-Nearest Neighbor (k-NN)

k-NN rule which is referred to as *instance-based learning* in Artificial Intelligence [5] became popular in machine learning after works of [6]. Referring to a rigorous definition [1]: The k-NN classifier assigns the input to the class having most examples among the k neighbors of the input. All neighbors have equal vote, and the class having the maximum number of voters among the k neighbors is chosen. Ties are broken arbitrarily or a weighted vote is taken. k is generally taken to be an odd number to minimize the ties: confusion is generally between two neighboring classes. Posterior probability density of k-NN classifier is given as:

$$\hat{P}(C_i|x) = \frac{\hat{p}(x|C_i)\hat{P}(C_i)}{\hat{p}(x)} = \frac{k_i}{k} \tag{1}$$

where k_i is the number of neighbors which belong to class C_i. This posterior probability is to be compared against the posterior extracted from clustering.

2.3 Extracting Posteriors from Clustering and Analyzing Convergence

In posterior density estimation using unsupervised learning, clustering process is done as is using training set. This process is hard labeling of the training set instances. Following this, instances in validation set are assigned to the closest cluster using the distance to centroids (rather than single-link or complete-link distances to clusters which are computationally more costly). The class distribution of this 'closest cluster' is assumed as posterior density of validation set instance. Mathematically speaking,

$$\hat{p}(C_i|x) = \hat{p}(C_i|G_j) \tag{2}$$

where G_j is the cluster formed around m_j satisfying

$$||x - m_j|| = min_l||x - m_l||, \quad 1 \leq l \leq K \tag{3}$$

Since the process contains random selection of initial centroids, the quality of clustering may vary. Therefore as a reliability measure to attain an expected posterior, the process of extraction is repeated sufficiently many times and then average is taken. In the following step, posterior density can be used directly for discrimination or be stacked to another learner in an ensemble setting. Since multi-view/ensemble

learning is beyond the scope of this paper, for further details on stacking features extracted from multi-view ensembles using this method reader may refer to [2].

2.4 Applications to Local Learning

In machine learning literature, local learning corresponds to dividing the input space into local regions and learning a separate model in each local region [1]. Some well known local models are Radial Basis Function (RBF) network [7, 8], Adaptive Resonance Theory (ART) [9], Mixture of Experts (MoE) [10] and Self Organizing Maps [11].

Feature-extraction in multi-view datasets, as explained in former section, could be a local learning application of the study. It is also possible to utilize the study in terms of local learning to fit local models after dividing the sample space. Bottou and Vapnik [12] propose a method similar to MoE. In their model; upon reception of a test instance, a simple local model is created with k nearest neighbors (with a large k) and discarded after the prediction. In their application, this method was more accurate than MLP, k-NN, and Parzen windows. Therefore, another implication of this study is using (ensemble of local models fit to) K-Means clusters to overcome the computational drawback of on-the-fly model generation.

To compare the computational complexities of Bottou and Vapnik's model (M_1) and the one proposed in this study (M_2); let N be dataset cardinality, K be the number of clusters, k be the approximate number of local instances, L be the number of clusterings for aggregation and $O(X)$ be the complexity of creating any local model (e.g. MLP) with k instances of the dataset. If we use 5×2 cross validation test proposed by [13] we need $5 \times N$ predictions. In this case:

$$O(M_1) = 5 \times N \times O(X) \tag{4}$$

whereas

$$O(M_2) = 5 \times K \times L \times O(X) \tag{5}$$

When $K \times L \ll N$, models created with K-Means will be much more efficient. One may also consider complexity of k-NN and K-Means as preprocessing steps in both approaches. Efficient implementations of both have an upper bound of $O(d \cdot n^2 \cdot \log(n))$ where d is dimensionality and n is cardinality of the dataset split into 2 equal halves, namely for training and validation. Our assumption of *almost same number of instances,* in both approaches, to fit an approximate predictive model having the same complexity $O(X)$, will be empirically verified in coming sections by showing $k \cdot K \sim N$ at convergence.

3 Findings

3.1 Car Evaluation Dataset

The dataset which was introduced by [14] is available in UCI ML Repository. The dataset was prepared for classification using decision trees therefore it was pre-processed before simulations for this study.

3.1.1 Dataset Description

The Car Evaluation Database directly relates a car to the six attributes: buying, maintenance, doors, persons, lug_boot, and safety. It consists of 1728 instances which completely cover the attribute space. In order to simplify this problem to binary classification, the acceptable classes namely *acc*, *good* and *v-good* are merged to form one class (*acc*) with nearly 30 % prior probability and *unacc* is used as the other class with nearly 70 % prior probability.

3.1.2 Experimental Results

The correlation between hyper-parameters of closest k-NN and corresponding K-Means predictors were found to be ~ -0.94. The k of k-NN and the K of K-Means are taken as follows:

$$\text{Set of k-NN hyper-parameters} = \{3, 5, 7, 9, 11, 13, 15, 17, 19, 21\}$$
$$\text{Set of K-Means hyper-parameters} = \{41, 82, 123, 164, 205, 246, 287, 328, 369, 410\}$$

K values picked are multiples of N/M where N is the training set cardinality (the number of samples) and M is the largest value of k used in k-NN. The primary concern of this study is not finding a more accurate learner, however, in order to have an insight about the predictor accuracy; k-NN accuracies are computed for the various k values and found to be around 95 %. Dissimilarity/distance of predictions is calculated using Root Mean Squared Error (RMSE), where the error is the difference between the posterior probabilities of the positive class. The distance matrices formed for two folds are similar (the one for fold 2 given as a representative in Table 1 with minimum distance with respect to each k of k-NN shown in bold).
The correlations of corresponding hyper-parameters are: -0.872 for fold 1 and -0.943 for fold 2. This implies that there is a strong inverse correlation between hyper-parameters of most similar predictors. When the predictor distance matrix is analyzed the best fits are found around bottom left (21-NN and 123-Means) with an implication that with higher values of k better convergence can be obtained. The plots of converging hyper-parameters are given in Fig. 1.

Table 1 RMSE between predictions of k-NN and K-Means on Car Dataset

K / k	41	82	123	164	205	246	287	328	369	410
3	0.2317	0.2043	0.1864	0.1711	0.1702	0.1596	0.1514	0.1485	0.1535	**0.1423**
5	0.1967	0.1735	0.1573	0.1441	0.1452	0.1332	0.1270	0.1264	0.1294	**0.1220**
7	0.1776	0.1526	0.1423	0.1320	0.1311	0.1261	**0.1255**	0.1276	0.1279	0.1270
9	0.1658	0.1470	0.1382	0.1329	0.1341	**0.1285**	0.1288	0.1349	0.1347	0.1341
11	0.1573	0.1388	0.1311	0.1282	0.1302	**0.1267**	0.1302	0.1355	0.1367	0.1373
13	0.1470	0.1317	0.1247	**0.1220**	0.1270	0.1273	0.1297	0.1361	0.1373	0.1382
15	0.1405	0.1267	0.1202	**0.1176**	0.1232	0.1264	0.1276	0.1352	0.1397	0.1402
17	0.1344	0.1223	0.1173	**0.1155**	0.1202	0.1270	0.1294	0.1355	0.1423	0.1435
19	0.1294	0.1194	**0.1155**	0.1164	0.1211	0.1285	0.1317	0.1376	0.1438	0.1467
21	0.1267	0.1179	**0.1152**	0.1170	0.1232	0.1308	0.1338	0.1402	0.1467	0.1491

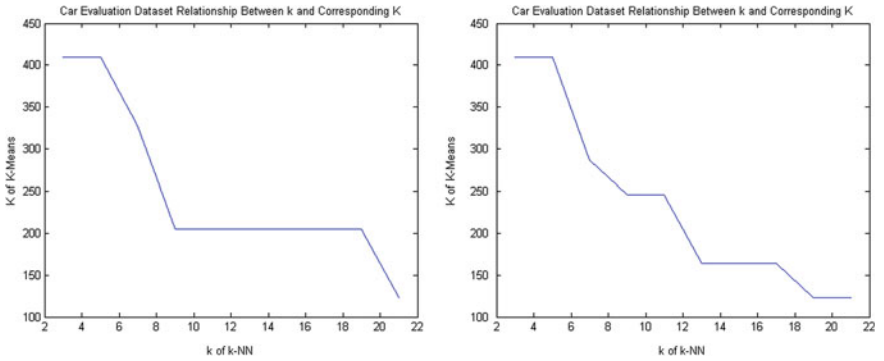


Fig. 1 Plot of k versus K having closest prediction for Car Dataset fold 1 and 2, respectively

3.2 Parkinson's Disease Dataset

3.2.1 Dataset Description

The dataset is composed of a range of biomedical voice measurements from 42 people with early-stage Parkinson's disease recruited to a six-month trial of a telemonitoring device for remote symptom progression monitoring. The target variables are motor UPDRS and total UPDRS, whereas input variables are 16 biomedical voice measures. There are 5.875 voice recordings in total where each of 42 patients has around 200 recordings. The referred acronym UPDRS stands for Unified Parkinson's Disease Rating Scale [15]. It is important to note that each fold contains 21 clients (not just any equal-size split of the samples); that is all samples of a single client were either in the training or the validation set as suggested by [16].

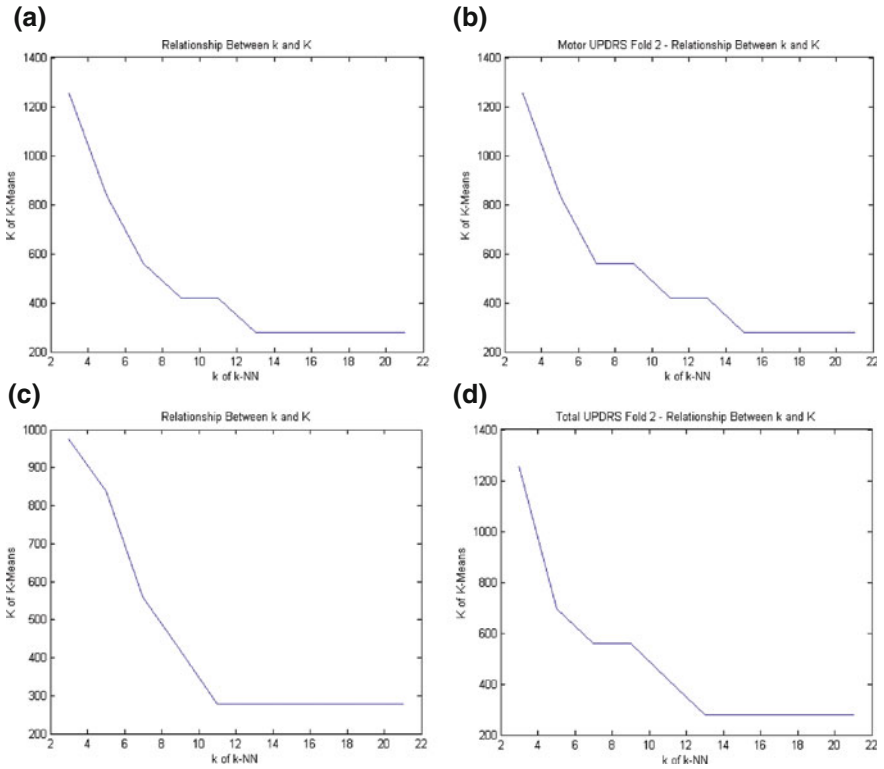


Fig. 2 Converging hyper-parameter values for two folds of Motor and Total UPDRS

3.2.2 Experimental Results

The same set of k values was used for k-NN. Since the dataset was of different cardinality, N, to form quasi-inverse pairs (i.e. $k \times K \sim N$), the set of K for K-Means was selected as follows:

$$\text{Set of k-NN hyper-parameters} = \{3, 5, 7, 9, 11, 13, 15, 17, 19, 21\}$$
$$\text{Set of K-Means hyper-parameters} = \{139, 279, 418, 558, 697, 837, 976, 1115, 1255, 1394\}$$

To have an insight about the problem complexity, again, k-NN tests were carried out, which revealed a Mean Absolute Error (note that this is regression task) around 8.0 for Motor UPDRS and around 10.0 for Total UPDRS, which comply with the results of [15]. In Fig. 2, the quasi-inverse relationship is shown for PD dataset, in which the dissimilarity/distance of predictions is calculated using RMSE over all test examples. For the RMSE calculation, the error between the predictors is taken as the difference between the regression outputs for each example. In Table 2, we observe

Table 2 RMSE between predictions of k-NN and K-Means on Motor UPDRS dataset

K/k	139	279	418	558	697	837	976	1115	1255	1394
3	4.9488	4.3888	3.8718	3.6564	3.3818	3.1987	3.0048	2.8971	**2.8756**	2.9294
5	3.8234	3.2525	2.7840	2.5740	2.4609	**2.4556**	2.4556	2.5525	2.7517	3.0318
7	3.2687	2.7140	2.3156	**2.2079**	2.2671	2.4233	2.5256	2.6763	2.9618	3.3064
9	2.8433	2.3102	2.0301	**2.0194**	2.1755	2.4448	2.6333	2.8056	3.1448	3.4895
11	2.5256	2.0409	**1.8794**	1.9278	2.1863	2.5148	2.7733	2.9348	3.2956	3.6456
13	2.2509	1.8417	**1.8255**	1.9494	2.2779	2.644	2.9294	3.0964	3.4679	3.8126
15	2.0463	**1.7286**	1.8147	1.9978	2.3748	2.7571	3.0479	3.2202	3.6026	3.9526
17	1.9009	**1.6586**	1.8309	2.0517	2.4609	2.8541	3.1502	3.3333	3.7103	4.0603
19	1.7555	**1.5993**	1.8417	2.1002	2.5256	2.9294	3.231	3.4087	3.7964	4.1357
21	1.6532	**1.5617**	1.8524	2.1486	2.5794	2.9833	3.2902	3.4679	3.8557	4.1949

Table 3 Correlations of corresponding hyper-parameters for 2-fold and 2-target variables

	Motor	Total
Fold 1	−0.8299	−0.8421
Fold 2	−0.8299	−0.8675

a similar pattern with Table 1: apart from the negative correlation of most similar predictions, error decreases steadily with increasing k (i.e. with larger neighborhood). The maximal convergence is attained between 21-NN and its counterpart. Table 3 summarizes the correlations attained in two folds for each target variable. These findings support the strong inverse correlation argument.

4 Conclusions

This study aims to analyze the convergence of predictors to elicit the implicit relationship of k-NN (a well known non-parametric classifier) and K-Means (a common unsupervised learner for clustering). The results indicate a strong negative correlation among hyper-parameters when their predictions are closest to each other. This correspondence can be utilized in two ways for decreasing time complexity without compromising the accuracy: (i) k-NN should be preferred for extracting cluster posterior probabilities, instead of K-Means, in multi-view and high-dimensional datasets, (ii) K-Means should be preferred for local model creation, instead of k-NN, in datasets with large number of samples. Our findings have also shown that the correspondence is at maximal level for large k of k-NN and small K of K-Means.

References

1. Alpaydın, E.: Introduction to Machine Learning. MIT Press, Cambridge (2010)
2. Kaya, H., Kursun, O., Seker, H.: Stacking class probabilities obtained from view-based cluster ensembles. In: Rutkowski, L. et al. (eds.) Proceedings of the 10th International Conference on

Artificial Intelligence and Soft Computing, ICAISC 2010, Part I, Springer-Verlag. LNAI 6113, pp. 397–404 (2010)
3. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. Department of Information and Computer Science, University of California, Irvine (2007)
4. Jiawei, H., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Francisco (2006)
5. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 3rd edn. Prentice-Hall, Upper Saddle River (2010)
6. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Mach. Learn. **6**, 37–66 (1991)
7. Broomhead, D.S., Lowe, D.: Multivariable functional interpolation and adaptive networks. Complex Syst. **2**, 321–355 (1988)
8. Moody, J., Darken, C.: Fast learning in networks of locally-tuned processing units. Neural Comput. **1**, 281–294 (1989)
9. Carpenter, G.A., Grossberg, S.: The ART of adaptive pattern recognition by a self-organizing neural network. IEEE Comput. **21**(3), 77–88 (1988)
10. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Comput. **3**, 79–87 (1991)
11. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (1995)
12. Bottou, L., Vapnik, V.: Local learning algorithms. Neural Comput. **4**, 888–900 (1992)
13. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. **10**, 1895–1923 (1998)
14. Bohanec, M., Rajkovic, V.: Knowledge acquisition and explanation for multi-attribute decision making. In: 8th International Workshop on Expert Systems and their Applications, Avignon, France. pp.59–78 (1988)
15. Tsanas, A., Max, A.L., McSharry, P.E., Ramig, L.O.: Accurate telemonitoring of parkinson's disease progression by non-invasive speech tests. IEEE Trans Biomed. Eng. **57**(4), 884–893 (2010)
16. Sakar, C.O., Kursun, O.: Telediagnosis of parkinson's disease using measurements of dysphonia. J. Med. Syst. **34**(4), 591–599 (2010)

Part V
Computer Vision I

Age Estimation Based on Local Radon Features of Facial Images

Asuman Günay and Vasif V. Nabiyev

Abstract This paper proposes a new age estimation method relying on regional Radon features of facial images and regression. Radon transform converts a pixel represented image an equivalent, lower dimensional and more geometrically informative Radon pixel image and it brings a large advantage achieving global geometric affine invariance. Proposed method consists of four modules: preprocessing, feature extraction with Radon transform, dimensionality reduction with PCA and age estimation with multiple linear regression. We conduct our experiments on FG-NET, MORPH and FERET databases and the results have shown that proposed method has better results than many conventional methods on all databases.

Keywords Age estimation · Radon transform · PCA · Regression

1 Introduction

The general topic of facial image processing have been received considerable interest in the last several decades, but facial image based age synthesis and estimation have become interesting topics in recent years because of their emergent real world applications such as forensic art, electronic customer relationship management, security control and surveillance monitoring, cosmetology, entertainment and biometrics.

There have been many researchers working in the area of facial image processing but only a small number of researches study in the area of modeling aging effects on facial images. The reason is that age estimation is much more complicated than

A. Günay (✉) · V. V. Nabiyev
Department of Computer Engineering, Karadeniz Technical University,
61080 Trabzon, Turkey
e-mail: gunaya@ktu.edu.tr

V. V. Nabiyev
e-mail: vasif@ktu.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_19,

recognizing other attributes such as gender, facial expressions and ethnicity. Furthermore facial aging effects display some unique characteristics [12]. There are fundamental difficulties in estimating age even humans have difficulty in determining a person's age correctly. These difficulties are: (1) Age estimation is not a standard classification problem. It can be taken either a multi-class classification problem or a regression problem. (2) A large aging database, especially the chronometrical image series of an individual is often hard to collect. (3) Age progression displayed on faces is uncontrollable and personalized.

2 Related Work

The existing age estimation systems are typically consisting of age image representation and age estimation modules. Age image representation techniques were often based on shape-based and texture-based features that were extracted from facial images. They can be grouped under the topics of anthropometric models, Active Appearance Models (AAM), AGing pattErn Subspace (AGES), Age Manifold and Appearance Models. Then age estimation can be performed with age group classification or regression methods.

The earliest paper published in the area of age classification from facial images was the work by Kwon and Lobo [15]. They computed six ratios of distances on frontal images to separate babies from adults. They also use the wrinkle information to separate young adults from senior adults. They use a very small database containing 45 facial images in their experiments.

AAM is a statistical face model proposed initially in [5] for facial image coding. A statistical shape model and an intensity model are learned separately from training images and combined based on Principal Component Analysis (PCA). Lanitis et al. [16] extended AAMs for face aging by proposing an aging function $age = f(b)$, to explain the variation in age. AAM based approaches consider both the shape and texture rather than just the facial geometry as in the anthropometric model based methods. But they have to deal with each aging face image separately.

Geng et al. [11] proposed a method called AGES that defines a sequence of personal face images of the same person sorted in the temporal order. Then a specific aging pattern is learned for each individual. AGES method can synthesize the missing age images by using an EM-like iterative learning algorithm. The Mean Absolute Error (MAE) was reported 6.77 years on FG-NET [6] database when the algorithm is tested in Leave One Person Out (LOPO) mode. They also used MORPH [18] database in their experiments. MORPH is only used to test the algorithms trained on the FG-NET database. AGES method achieves 8.83 MAE on MORPH database [12].

Instead of learning a specific aging pattern for each individual as in AGES, age manifold methods can learn a common aging trend or pattern from many individuals at different ages. This kind of aging pattern learning makes the task of face aging representation very flexible. The possible way to learn the common aging pattern is age manifold which utilizes manifold embedding technique to learn the low dimensional aging trend from many face images at each age [4, 7, 8, 13].

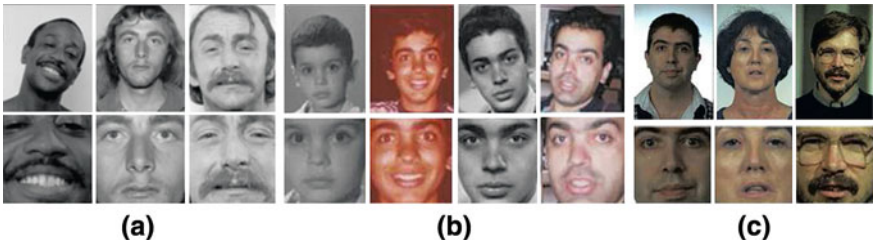


Fig. 1 Samples from databases. **a** MORPH **b** FG-NET **c** FERET

Appearance models are mainly focused on the aging-related facial feature extraction. Both global and local features were used in existing age estimation systems [1, 9, 10, 14] .

As one can see from the previous work, there have been many methods proposed in the age estimation field and most of them are implemented on FG-NET Aging database. In this study we use the Radon transform for age estimation for the first time and make experiments on three databases: FG-NET Aging database [6], MORPH database [18] and FERET database [17]. The results have shown that Radon features are efficient for age estimation on all databases.

This paper is organized as follows: Sect. 3 introduces the proposed method for age estimation including preprocessing, feature extraction, dimensionality reduction and regression modules. These modules are described in the sections of Sect. 3. Section 4 describes the databases used in the experiments. Section 5 discusses the experimental results and finally Sect. 6 concludes the paper.

3 Proposed Method

In this paper we propose a new age estimation method by using local features of facial images. Local features are extracted using regional Radon transform of facial images. This method consists of four modules: preprocessing, feature extraction with Radon transform, dimensionality reduction with PCA and age estimation with multiple linear regression. These modules are explained in the following sections.

3.1 Preprocessing

In the preprocessing module, the facial images are cropped, scaled and transformed to the size of 88×88, based on the eye center locations. Examples from all databases are given in Fig. 1.

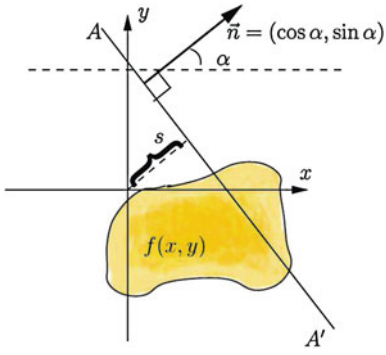


Fig. 2 Radon transform

3.2 Feature Extraction with Radon Transform

The Radon transform [3] compute projections of an image matrix along specified direction. Applying the Radon transform on an image $f(x, y)$ for a given set of angles can be thought of as computing the projection of the image along the given angles. The resulting projection is the sum of the intensities of the pixels in each direction, i.e. a line integral [2]. The result is a new image $R(s, \alpha)$. The Radon transform for an image can be written as;

$$R(s, \alpha) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\delta(s - x\cos\alpha - y\sin\alpha)dxdy \tag{1}$$

where $R(S, \alpha)$ is the line integral of a 2-D function $f(x,y)$ along a line from $-\infty$ to ∞. The position of the line is determined by two parameters s and α. Essentially, $R(s, \alpha)$ is the integral of f over the line $s = x\cos\alpha + y\sin\alpha$. In its discrete form, a Radon transform consists in the summation of pixel intensities along lines of different directions. The process for the calculation of Radon coefficients is visualized in Fig. 2.

The Radon pixel image has more geometric information than the original pixel image. For age estimation field, the regional texture information is more informative than global texture information. Consequently we have taken the Radon projections at $\theta = k\pi/6$ where $k = 0, 1, 2, 3, 4, 5$ from local image regions and concatenated them in a single feature vector. Regional Radon transform produces a feature vector with dimension of 3,920 for 4×4 regions.

3.3 Dimensionality Reduction

After the feature extraction module, Principal Component Analysis (PCA) is performed in order to find a lower dimensional subspace which carries significant infor-

mation for age estimation. Then high-dimensional feature vectors are projected onto a low-dimensional subspace in order to improve the efficiency. Using this technique the p-dimensional feature vector x is transformed into a d-dimensional vector y with $d < p$.

The PCA method finds the embedding that maximizes the projected variance, $p = \arg\max_{\|\mathrm{p}=1\|} \mathrm{p}^{\mathrm{T}}\mathrm{Sp}$, where $S = \sum_{i=1}^{n}(x_i - \bar{x})(x_i, -\bar{x})^T$ is the scatter matrix, and $\bar{x}$ is the mean vector of $\{x_i\}_{i=1}^{n}$. The solution of this problem is given by the set of d eigenvectors associated to the d largest eigenvalues of the scatter matrix. Once the projection subspace is determined, training and testing images were projected on it, allowing thus dimensionality reduction.

3.4 Regression

After finding the low dimensional representation of facial images, we define the age estimation problem as a multiple linear regression problem as $age = f(M)$: $\Leftrightarrow$ $\hat{L} = \hat{f}(Y)$, where $\hat{L}$ denotes the estimated age label, $f(\cdot)$ the unknown regression function, and $\hat{f}(\cdot)$ is the estimated regression function. The age regression function used in this study is a linear function, $\hat{\ell} = \hat{\beta}_0 + \hat{\beta}_1^T y$, where $\hat{\ell}$ is the estimate of age, $\hat{\beta}_0$ is the offset , $\hat{\beta}_1$ is the weight vector and y is the extracted feature vector.

4 Databases

The databases used in this study are FG-NET, MORPH and FERET databases. The Face and Gesture Recognition Research Network (FG-NET) aging database [6] comprises of 1,002 images of 82 subjects (6–18 images per subject) in the age range 0–69 years. Since the images were retrieved from real-life albums of different subjects, aspects such as illumination, head pose, facial expressions etc. are uncontrolled in this dataset.

The MORPH Database [18] is a public available face database, comprises face images of adults taken during different ages. The database records individuals' metadata such as age, gender, ethnicity, height, weight etc., and is organized into two albums. MORPH Album-1 (A1) comprises of 1,690 digitized images of 515 individuals between the age range 15–68 years.

The FERET Database [17], a comprehensive database that addresses multiple problems related to face recognition such as illumination variations, pose variations, facial expressions etc. The database includes 14,126 images from 1,199 individuals. In this study we use 2,294 facial images that are taken from frontal view images.

The age distribution of FG-NET, MORPH and FERET databases is given in Table 1. One can see from the table that the images are not distributed uniformly. This irregularity affects the estimation results negatively.

Table 1 The distribution of images in specified age groups

Database	Age groups							
	≤9	10–19	20–29	30–39	40–49	50–59	≥60	Number of images
FGNET	411	319	143	69	39	14	7	1,002
MORPH-A1	0	433	735	390	107	19	6	1,690
FERET	36	887	493	481	297	78	22	2,294

5 Experiments and Results

In the training phase, we have taken the Radon projections of training samples at $\theta = k\pi/6$ where $k = 0, 1, 2, 3, 4, 5$. We extract Radon features from local image regions and concatenated them in a single feature vector. Regional Radon transform produces a feature vector with dimension of 3,920 for 4×4 regions. Then we apply PCA to reduce the dimension of feature vector. After dimensionality reduction step we define an aging function using multiple linear regression. In the testing phase, the regional Radon features of test samples are extracted similarly. Then age estimation is performed using the predicted aging function.

The evaluation framework is Leave-One-Person-Out (LOPO) mode for FG-NET Aging Database. That is in each fold the images of one person are used as test set and those of the others are used as the training set. After 82 folds, each subject has been used as test set once, and the final results are calculated based on all the estimations. In this way the algorithms are tested in the case similar to real applications.

In the experiments we also use 5-fold cross validation mode for MORPH and FERET in which the 1/5 of the images are selected randomly as test set and the rest are used as training set. After 5-folds the mean of all estimations is determined as estimation performance of the system.

For the performance comparison, we used the Mean Absolute Error (MAE) measurement. MAE is defined as the average of the absolute error between the recognized labels and the ground truth labels:

$$MAE = \frac{\sum_{i=1}^{N_t} \left| \hat{y}_i - y_i \right|}{N_t} \tag{2}$$

where $\hat{y}_i$is the recognized age for the ith testing sample, y_i is the corresponding ground truth, and N_t is the total number of the testing samples. The estimation results of conventional methods and proposed method for FG-NET and MORPH databases are listed in Table 2. We can see from Table 2 that, the proposed method achieves better result than conventional methods like WAS, AAS, KNN on these databases.

Table 2 The comparison of estimation results (MAE) on FG-NET and MORPH databases

	WAS [11]	AAS [11]	KNN [12]	BP [12]	SVM [12]	AGES [11]	$AGES_{lda}$ [12]	Radon
FG-NET	8.06	14.83	8.24	11.8	7.25	6.77	6.22	6.18
MORPH	9.32	20.93	11.3	13.8	12.69	8.83	8.07	6.65

Table 3 The comparison of estimation results on FERET database

Methods	Results
LBP [19]	7.88 % (Error tolerance)
Radon	6.98 (MAE)

Finally the performance of Radon features on FERET database is given in Table 3. There haven't been enough studies reported on FERET database. In [19] the LBP features of facial images are used and the classification error (error rate) for 3 age classes (child, youth and oldness) is 7.88 %. In this study we take the age estimation problem as a regression problem and we achieve 6.98 MAE on FERET database.

6 Conclusion

In this paper, we have presented an age estimation method that uses regional Radon transform for age-related feature extraction from facial images. The contribution of this paper is using regional Radon features for age estimation. The Radon features are extracted from local image regions and concatenated into single vector. Thus the global and local geometrical information of facial images are included in the feature vector. Experimental results on the FG-NET aging database, MORPH and FERET databases have shown that proposed method is better than most conventional methods. Furthermore our result is slightly better than all age estimation results reported previously on MORPH database.

References

1. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: application to face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **28**(12), 2037–2041 (2006)
2. Al-Shaykh, O., Doherty, J.: Invariant image analysis based on radon transform and svd. IEEE Trans. Circuit. Syst. II Analog Digit. Sig. Process. **43**(2), 123–133 (1996)
3. Beylkin, G.: Discrete Radon transform. IEEE Trans. Acoust. Speech Sig. Process. ASSP **35**(2), 162–171 (1987)
4. Cai, D., He, X., Han, J., Zhang, H.-J.: Orthogonal laplacianfaces for face recognition. IEEE Trans. Image Process. **15**(11), 3608–3614 (2006)
5. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE Trans. PAMI **23**(6), 681–685 (2001)

6. FG-Net aging database: http://sting.cycollege.ac.cy/alanitis/fgnetaging/
7. Fu, Y., Xu, Y., Huang, T.S.: Estimating human age by manifold analysis of face pictures and regression on aging features. In: Proceedings of IEEE International Conference on Multimedia and Expo, pp. 1383–1386 (2007)
8. Fu, Y., Huang, T.S.: Human age estimation with regression on discriminative aging manifold. IEEE Trans. Multimedia **10**(4), 578–584 (2008)
9. Fukai, H., Takimoto, H., Mitsukura, Y., Fukumi, M.: Apparent age estimation system based on age perception. In: Proceedings of SICE 2007 Annual Conference Takanatsu, pp. 2808–2812 (2007)
10. Gao, F., Ai, H.: Face age classification on consumer images with gabor feature and fuzzy LDA method. In: Proceedings of 3rd International Conference on Advances in Biometrics LNCS'5558, pp. 132–141. Alghero, Italy (2009)
11. Geng, X., Zhou, Z.H., Zhang, Y., Li, G., Dai, H.: Learning from facial aging patterns for automatic age estimation. In: Proceedings of ACM Conference on Multimedia, pp. 307–316 (2006)
12. Geng, X., Zhou, Z.H., Miles, K.S.: Automatic age estimation based on facial aging patterns. IEEE Trans. Pattern Anal. Mach. Intell. **29**(12), 2234–2240 (2007)
13. Guo, G., Fu, Y., Huang, T.S., Dyer, C.R.: Locally adjusted robust regression for human age estimation. In: IEEE Workshop on Applications of Computer Vision, WACV'08, pp. 1–6. Copper Mountain (2008)
14. Ju, C.H., Wang, Y.H.: Automatic age estimation based on local feature of face image and regression. In: International Conference on Machine Learning and Cybernetics, pp. 885–888. Hebei University, Baoding (2009)
15. Kwon, Y.H., Lobo, N.V.: Age classification from facial images. Comput. Vis. Image Underst. **74**(1), 1–21 (1999)
16. Lanitis, A., Taylor, C., Cootes, T.: Toward automatic simulation of aging effects on face images. IEEE Trans. PAMI **24**(4), 442–455 (2002)
17. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face recognition algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **22**, 1090–1104 (2000)
18. Ricanek, K. Jr., Tesafaye, T.: MORPH: a longitudinal image database of normal adult age-progression. In: IEEE 7th International Conference on Automatic Face and Gesture Recognition, Southampton, UK, pp. 341–345 (2006)
19. Yang, Z., Ai, H.: Demographic classification with local binary patterns. In: Proceedings of International Conference on Advances in Biometrics, LNCS'4642, Seoul, Korea, pp. 464–473 (2007)

Paper and Pen: A 3D Sketching System

Cansın Yıldız and Tolga Çapın

Abstract This paper proposes a method that resembles a natural pen and paper interface to create curve based 3D sketches. The system is particularly useful for representing initial 3D design ideas without much effort. Users interact with the system by the help of a pressure sensitive *pen* tablet. The input strokes of the users are projected onto a drawing plane, which serves as a *paper* that they can place anywhere in the 3D scene. The resulting 3D sketch is visualized emphasizing depth perception. Our evaluation involving several naive users suggest that the system is suitable for a broad range of users to easily express their ideas in 3D. We further analyze the system with the help of an architect to demonstrate the expressive capabilities.

1 Introduction

3D modeling starts with rough sketching of ideas. The latest efforts in research on the field have focused on bringing the natural pen and paper interface to 3D modeling world. The complicated and hard-to-learn nature of current *WIMP* (windows, icon, pointer, menu) based 3D modeling tools is the reason for the search of a better interface. Several authors has already recognized the importance of this problem [18]. In this paper, we present a method that tries to mimic the natural interface of pen and paper for creating 3D sketches that can be used an easier way to represent ideas in 3D without much effort. The system is designed to be as minimalistic and simple as possible, since it targets a broad range of users, from expert designers to

C. Yıldız (✉) · T. Çapın
Department of Computer Engineering, Bilkent University,
06800 Bilkent, Ankara, Turkey
e-mail: cansin@cs.bilkent.edu.tr
URL: http://cs.bilkent.edu.tr

T. Çapın
e-mail: tcapin@cs.bilkent.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_20,

naive users. We test whether we are able to achieve this or not, through several user tests (Sect. 5).

Our system is based on the very idea of *curves*, rather than 3D solid objects. Concern of creating surfaces not in mind, it is much easier to develop complicated 3D scenes and objects. Although there are several other examples of a similar 3D sketching interface, our contribution to the field is to explain an easy to use 3D sketching tool, that is designed with *less is more* [17] thought in mind. Several different sketching tools also developed during implementation of the system.

2 Related Work

Creating 3D objects and curves from 2D user interfaces has been studied for a long time [7, 13, 21]. There are several recent research on the subject that tries to enrich an already existing 3D scene, either by annotating the scene or augmenting the 3D object itself [9, 14, 15]. Bourguignon et al. [9] created a system that can be used both annotating a 3D object or creating an artistic illustration that can be represented from different viewpoints. Although the resulting scenes are pleasingly beautiful, they are not truly 3D. The system mimics a 3D perspective by manipulating the curves' render mechanism according to the angle they make with the viewport. At Kara et al.'s [14, 15] work, a true 3D object is created by augmenting a simpler pre-loaded 3D template of the target object. Simply, if user wants to create a fancy chair, a simpler chair model is loaded beforehand, which the user can edit with a sketch interface.

Bae et al.'s *ILoveSketch* [3], and later extended version *EverybodyLovesSketch* [4], rely on the idea of creating curve-based 3D scenes rather than traditional plane-based 3D models as we do. Their approach uses several different drawing techniques and navigation tools that a user can select from. Although it is easy to learn the entry point to 3D sketching ideas such as *orthographic plane sketching* and *single-view symmetric curve*, it takes some time to learn how to use the system in depth, as they noted [4]. In our system, we have chosen to use only a single way of drawing and navigating (as explained in Sect. 3.1), which makes it much easier to learn.

3 Overview of the System

Users interact with our system using a pressure sensitive pen tablet. In essence, users' pen gestures are captured as a time sequenced tablet coordinates and interpreted. The device has several buttons on the tablet that are used for some basic non gestural abilities like undo, redo, toggle symmetry. The pen also has two buttons, and an eraser at back, that are used as toggles between our gesture modes as detailed in the following Sect. 3.1.

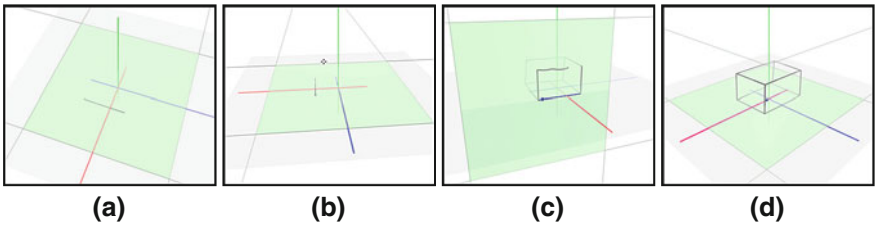


Fig. 1 Overview of the system. **a** User adjusts the drawing plane; and draws the curve using pen tablet. **b** The camera position can be changed. **c** The process is repeated until the desired 3D sketch is formed. **d** Final result

Figure 1 illustrates the overall usage of the system. To be able to draw a curve, the user firsts adjust the *drawing plane* as explained in Sect. 3.1. Any drawing gesture that is made by the user will be reflected on this surface. Once the drawing plane is adjusted, the user can draw a curve with a simple pen gesture on the tablet. The input curve will then be re-sampled and smoothed using the algorithms described at Sect. 4.1. During this process, the user can adjust camera position as well, using the same pen tablet device, if necessary. The user can repeat these steps to complete the 3D object.

3.1 Gesture Modes

The pen tablet acts like a *modal interface* for users, allowing it to be used for several different tasks. The user can switch between different modes by holding down the buttons on the pen. On a regular session with the system, one will use the pen for camera adjustment, plane selection, drawing and erasing.

- **Camera Adjustment** When the pen is in this mode, every movement user does will be mapped to an invisible *Two-Axis Valuator Trackball* [11]. The horizontal pen movement is mapped to a rotation about the up-vector, whereas a vertical pen movement is mapped to a rotation about the vector perpendicular to view and up. As Bade et al. suggested [2], Two-Axis Valuator Trackball is among "the best 3D rotation technique" among several rotational widgets.
- **Plane Selection** When the user draws curves with the tablet, these curves should be reflected onto a virtual surface at 3D scene. To enable this effect, the user should select a *drawing plane* beforehand. In our system, there are only two distinct ways of selecting the drawing plane.
  - In first approach, the user takes assistance from coordinate system lines and current curves on the scene. By selecting any of these curves, the user changes the drawing surface as the plane that selected curve lies on. Further flexibility is enabled with the help of *toggle plane* button on the tablet. Once that button is pushed, the drawing surface will be changed to one of the planes that forms

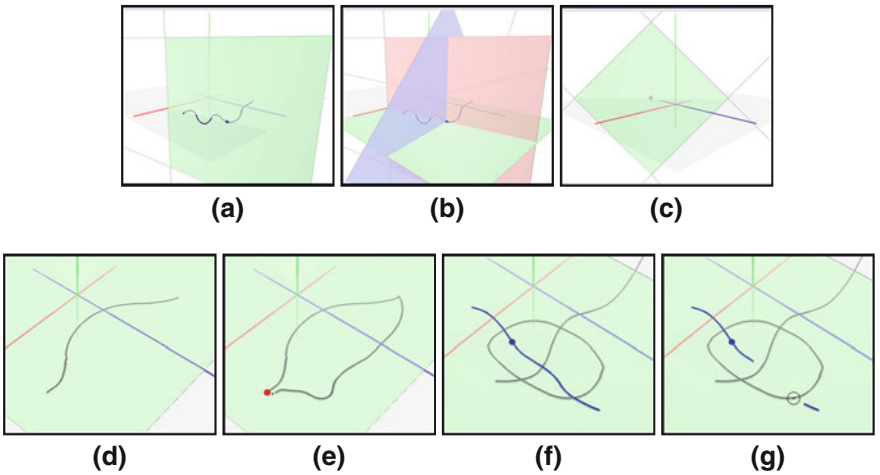


Fig. 2 **a, b** Plane selection with a Cartesian system or (**c**) extruding a picking ray. **d** Drawing gesture with (**e**) snap points. **f, g** Erasing

a Cartesian system with the drawing plane and tangential to the curve at the selected point (Fig. 2a, b).
– To support even more flexibility, we realized a second approach to plane selection. In this method, the user can adjust the drawing surface to a plane that is parallel to the current near plane of the scene's viewport, and x distant from that near plane, where that x is determined by the current pressure on the pen (Fig. 2c).

- **Drawing** The main functionality of the system, is drawing curves (Fig. 2d). In this mode, the user can simply draw several curves using pen tablet. The time sequenced (x, y) data that is collected from the pen tablet is then projected to the current drawing plane. After the projection is performed, several re-sampling and smoothing algorithms are used to ensure a plausible curve shape, as detailed in Sect. 4. Finally, a B-Spline curve is fitted to the stroke data. While in the drawing mode, the user can take advantage of *snap points* that will appear at the start and end points of existing curves (Fig. 2e). These snap points make it even easier to draw closed or connected shapes.
- **Erasing** A paper and pen system cannot be imagined without an eraser. The user can simply turn over his pen device to switch to the eraser mode. Once this is done, the cursor on the screen will get larger to mimic an eraser functionality. Since in a crowded scene, there will be several curves that will lie under eraser's cursor, it will be harder to erase a specific curve's segment. Therefore, erasing can only be performed on the current *selected* curve (Fig. 2f, g).

As mentioned, there are also several buttons on the tablet, that can be used to achieve some misc. operations. When symmetry is toggled, any gesture that's per-

formed with the pen will also be reflected to the symmetry of that gesture. Symmetry is important to product design, since people prefer objects with symmetry, unity and harmony [8].

To prevent errors that the users might make, the system changes the pen's cursor's image to reflect the current gesture mode [1]. For instance, it's a single dot for drawing, a bigger circle for erasing, a cross-hair for plane selection etc. Similarly, our system also supports undo/redo actions using the tablet buttons as well. This functionality is really essential for basic error recovery. As can be seen in Sect. 5, undo is widely used among our users.

4 Implementation Details

There is a common pipeline [18] for *sketch based interfaces*, which our system also follows. The first step is to acquire input from the user, by means of an input device, a pen tablet in our case. That step is followed by sketch filtering, where the data is re-sampled and smoothed. Finally, the sketch is interpreted appropriately.

4.1 Sketch Acquisition and Filtering

Obtaining a sketch from the user is the first step a sketch based interface should perform. Our system collects free hand sketches from the user using a pen tablet. A tablet display would be even a better choice, since the user will be able to see what he draws just at the drawing surface he is using.

It is important to perform filtering before storing a sketch to the system, since there will be some error caused by both user, and the input device itself [19]. Therefore, the input data should be interpreted knowing that it is imperfect. To overcome this imperfection, our system applies below approaches:

- **Re-sampling and Smoothing** The distance between data samples that are acquired from the pen tablet is not always the same (Fig. 3b). Therefore a re-sampling mechanism is needed to normalize distances (Fig. 3c). To further smooth out the given input, we use a local Gaussian filter (Fig. 3d) to any upcoming data point [20].
- **Fitting** After re-sampling and smoothing is performed, the resulting curve consists of hundreds of data points. To simplify this representation, we fit a curve onto these data points, using Reverse Chaikin Subdivision [6]. At every iteration of this algorithm, the data size halves. After appropriate number of iterations, these coarse points are used as control points for a B-Spline curve (Fig. 3e). Assuming fine points are denoted as $p_1, p_2, \ldots, p_n$, and coarse points are denoted as $c_1, c_2, \ldots, c_n$, a coarse point c_j can be computed as follows:

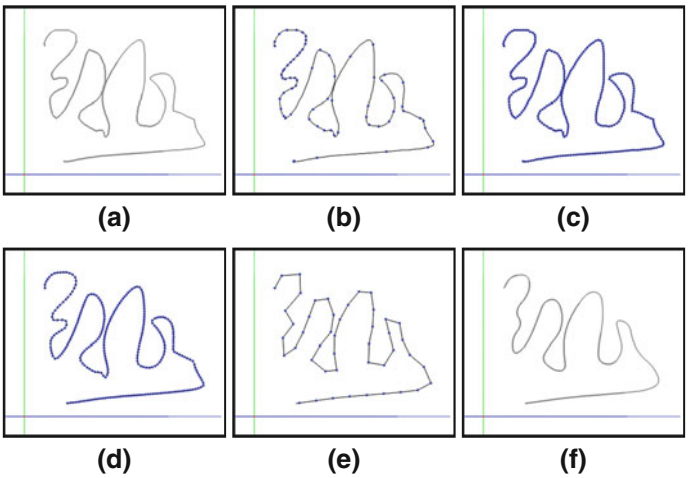


Fig. 3 **a** Initial user input. **b** Non-uniform distribution (706 points). **c** Re-sampled (2877 points). **d** Gaussian filtered. **e** Reverse Chaikin subdivided (47 points to *represent*). **f** Final B-spline curve (188 points to *render*)

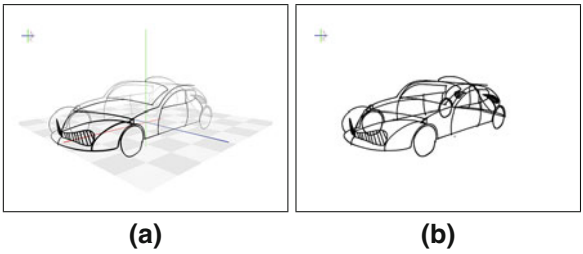


Fig. 4 Visualization at our system: **a** with depth cues; **b** without depth cues

$$c_j = -\frac{1}{4}p_{i-1} + \frac{3}{4}p_i + \frac{3}{4}p_{i+1} - \frac{1}{4}p_{i+2} \tag{1}$$

4.2 Visualization

Correct visualization of a scene is fairly important to make it easier for users to understand the 3D information behind the scene. In technical illustrations, there are three line conventions suggested by Martin [16]: use single line weight throughout the image; use heavy line weights for out edges, and parts with open space between them; or vary line weight to emphasize perspective (i.e. thicker is closer).

Since our concern is to emphasize 3D recognition as much as possible, we find third convention most suitable for the system. As can be seen at Fig. 4, by varying

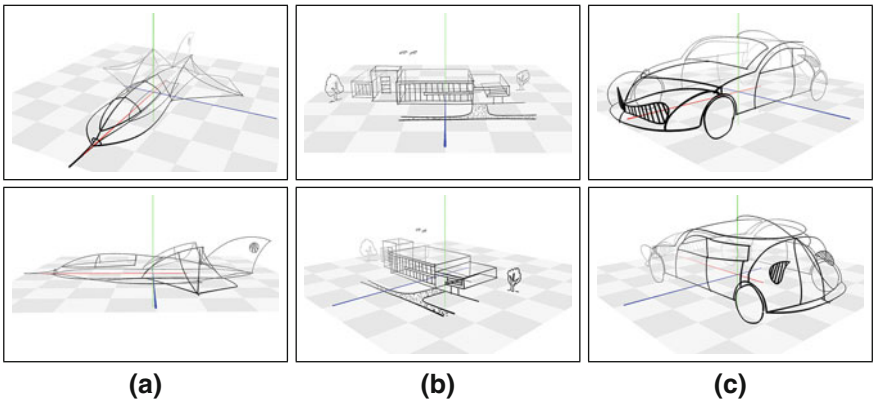


Fig. 5 Sample results. **a** A jet fighter. **b** A building complex. **c** A car

both line thickness and opacity with respect to z distance from the camera, our system makes it a lot easier to recognize the shape of 3D objects.

5 Results and Discussion

We invited an architect to perform a subjective expert evaluation. After an brief introductory explanation of the system for 15 min, the architect is left alone with the system for a full day. The resulting objects of the day can be seen at Fig. 5. The architect stated that he did like using the system, but he thinks such a system is more suitable for product design rather than architectural design. We agree on this comment, since our system tries to emphasize the power of free form curves, it is actually a bit harder to create regular shapes such as cubes and pyramids. One usability issue that we noted was that the architect preferred using undo function instead of erasing gesture most of the time. Only for some small adjustments, like shortening a curve which is a little too long, he used erasing.

We have also performed objective formal experiment to evaluate the usability of the system. We have selected twelve users that do not have prior experience with technical or artistic drawing, and pen tablets. In a standard test case, we introduced the system to each user briefly within five minutes. Then, we asked them to exactly copy the object they see on the scene. The test has twelve objects, some of which are 2D regular shapes, while others consist of 3D objects.

On average it took 67 s to draw a 2D object for all users, whereas it took 301 s for a 3D one. The slight complexity of 3D objects, and the need to adjust drawing plane several times, made 3D objects need more time to draw. We evaluate the resulting scenes with the goal objects using Modified Hausdorff Distance, as described in [12]. Evaluation suggests that the error for 3D objects is not that different from 2D object

errors, and on average that error is 0.12 for 2D scenes, and 0.14 for 3D scenes. Given a common object had at most 8 length dimension, 1.62 % (0.13 over 8) is indeed not a significant error. Therefore, we can say that it was as easy to draw 3D objects as it is for 2D objects.

We also conduct a System Usability Scale Survey for each user, at the end of the test. System Usability Scale is a simple, ten-item Likert scale giving a global view of subjective assessments of usability [10]. Over twelve test users, our system got 83.75 out of 100 which can be referred as an "excellent" or "B" grade system, according to Bangor et al.'s work [5].

6 Conclusions

We have created a 3D sketching system that can be broadly used by any user, almost like a 3D *paint*. We did push the limits of the system by working with a professional architect to see what the system is capable of, whereas we also test the system with naive users with a more simplistic way. These evaluations show that our system is an easy to use, yet capable 3D curve sketching interface that requires little learning effort.

References

1. Andre, A., Degani, A.: Do you know what mode you're in? an analysis of mode error in everyday things. In: Mouloua, M., Koonce, J.M. (eds.) Human-Automation Interaction: Research and Practice. Lawrence Erlbaum, Mahwah (1997)
2. Bade, R., Ritter, F., Preim, B.: Usability comparison of mouse-based interaction techniques for predictable 3d rotation. In: Proceedings of Smart Graphics (2005)
3. Bae, S., Balakrishnan, R., Singh, K.: ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. In: Proceedings of UIST '08 (2008)
4. Bae, S., Balakrishnan, R., Singh, K.: Everybodylovessketch: 3d sketching for a broader audience. In: Proceedings of UIST '09 (2009)
5. Bangor, A., Miller, J., et al.: Determining what individual sus scores mean: adding an adjective rating scale. J. Usability Stud. **4**(3), 114–123 (2009)
6. Bartels, R. Samavati, F.: Reversing subdivision rules: local linear conditions and observations on inner products. J. Comput. Appl. Math. **119**, 29–67 (2000)
7. Baudel, T.: A mark-based interaction paradigm for free-hand drawing. In: Proceedings of UIST '94 (1994)
8. Bloch, P.: Seeking the ideal form: product design and consumer response. J. Mark. **59**, 16–29 (1995)
9. Bourguignon, D., Cani, M., Drettakis, G.: Drawing for illustration and annotation in 3d. Comput. Graph. Forum **20**, 114–122 (2001)
10. Brooke, J.: Sus-a quick and dirty usability scale. In: Weerdmeester, B.A., McClelland, I.L. (eds.) Usability Evaluation in Industry. Taylor and Francis, London (1996)
11. Chen, M., Mountford, S.J., Sellen, A.: A study in interactive 3-d rotation using 2-d control devices. In: Proceedings of SIGGRAPH '88 (1988)

12. Dubuisson, M., Jain, A.: A modified hausdorff distance for object matching. In: Proceedings of the 12th IAPR (1994)
13. Igarashi, T., Matsuoka, S., Tanaka, H.: Teddy: a sketching interface for 3d freeform design. In: Proceedings of SIGGRAPH '99 (1999)
14. Kara, L., Shimada, K. Construction and modification of 3d geometry using a sketch-based interface. In: Proceedings of SBIM 06 (2006)
15. Kara, L., Shimada, K.: Sketch-based 3d-shape creation for industrial styling design. IEEE Comput. Graph. Appl. **27**, 60–71 (2007)
16. Martin, J.: Technical Illustration: Materials, Methods and Techniques/Judy Martin. Child and Associates, Frenchs Forest (1989)
17. Nielsen, J.: Usability Engineering. Morgan Kaufmann, San Francisco (1994)
18. Olsen, L., Samavati, F., Sousa, M., Jorge, J.: Sketch-based modeling: a survey. Comput. Graph. **33**, 82–109 (2009)
19. Sezgin, T., Davis, R.: Scale-space based feature point detection for digital ink. In: ACM SIGGRAPH 2007 courses (2007)
20. Taubin, G. Curve and surface smoothing without shrinkage. In: Proceedings of the 5th ICCV (1995)
21. Zeleznik, R., Herndon, K., Hughes, J.: Sketch: an interface for sketching 3d scenes. In: Proceedings of SIGGRAPH '96 (1996)

Perceptual Caricaturization of 3D Models

Gokcen Cimen, Abdullah Bulbul, Bulent Ozguc and Tolga Capin

Abstract Caricature is an illustration of a person or a subject that uses a way of exaggerating the most distinguishable characteristic traits and simplifying the common features in order to magnify the unique features of the subject. Recently, automatic caricature generation has become a research area due to the advantageous features of amusement in the fields such as network, communications, online games, and the animation industry. The aim of this study is to present a perceptual caricaturization approach practicing the concept of exaggeration, which is very common in traditional art and caricature, on 3D mesh models synthesizing the idea of mesh saliency.

1 Introduction

Caricature is a visual art that greatly exaggerates certain features of a subject to create a comic effect. The aim of the caricature is to find individual differences and more attractive features than other parts of the subject and exaggerate these features to create humorous look.

To automatically generate a caricature, a system should attempt to imitate the traditional artistic skills. Identification of the unique features is essential for creating caricatures since these features will eventually be exaggerated. To accomplish this challenge, we employ the idea of mesh saliency, proposed by Lee et al. 2005 [7], which is a technique to measure regional importance of 3D meshes. After extracting salient parts of the mesh feature points, the caricature is generated by warping the original 3D model from original feature points to saliency based exaggerated feature points by using a deformation technique. The deformation of 3D meshes is maintained by Free Form Deformation (FFD) method which is a common method

G. Cimen (✉)
Bilkent University, 06800 Ankara, Turkey
e-mail: gokcen.cimen@cs.bilkent.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_21,

used for mesh deformation [11]. With automatically calculated saliency values and deformation through these values, the method is fully automatic, but the user can determine the scale of feature exaggeration.
Among existing methods, most of them are based on manipulating 2D facial photographs or images. However, one major drawback of 2D facial caricaturization is that they are limited to make exaggeration with one missing dimension so that every feature deformed on a face is on the same plane. This creates pixel-based deformation, so results undesired distortion artifacts by pulling or pushing adjacent pixels. By performing the caricaturization in 3D space, we have successfully overcome these problems. With respect to this, in this study, we present a perceptual approach applying the exaggeration by using automatically perceptual based feature extraction.
The rest of this paper is organized as follows. In the next section, related work is discussed. In Sect. 3, the overview of the proposed framework is given and the steps of the technique are explained. The results and the conclusions are presented in Sect. 4.

2 Related Works

Susan Brennan's work is perhaps the milestone in formally defining the first caricature generation system in 1982 [3]. After Brennan's work, several cartoon caricature generation methods have been proposed with or without a training process [10]. The basic idea of these approaches is exaggerating the difference from the mean face, but forming a mean face requires large databases of random faces with hundreds of features identified in everyone.
Another proposed approach is training the computer to imitate a specific drawing style from examples. Based on a set of training images and their related hand drawn sketches by an artist, Chen et al. [4]'s method automatically generates an artist's styli tic caricature from an input image. In another study, Liang et al. [8]'s work attempts to learn the style of a caricature artist from example caricatures drawn by the artist using partial least squares. Besides, a recent study managed to generate digital caricatures from facial photographs that capture artistic deformation styles from hand-drawn caricatures by Clarke et al. [5]. Even if this approach observes and learns from the artist's products, they need artist's hand drawn style and can only get the copy of the style and results with limited success.
There have been a number of studies to interactively generate facial caricatures. Akleman [1, 2]. However, the limitation of these techniques is that they require prior knowledge of how a face would be deformed to generate caricature and they are designed for artists or art students. In addition, recently Fu et al. [6] proposed an interactive system that decomposes various facial components of a 3D head model and replace these components with other head and facial components stored in a database. Since these method does not make use of a ref-

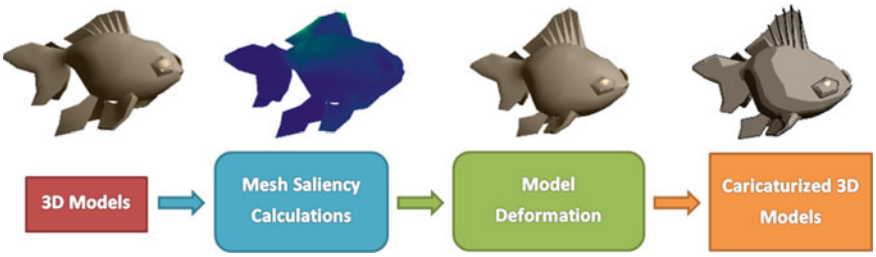


Fig. 1 Overview of the system

Fig. 2 Saliency map for three different 3D models. *Green regions* are more salient than *blues regions*, so they are far more deformed

erence head, the resulting caricature are lack of the likeliness to the original model.

3 Automatic Caricature Generation

Our framework, shown in Fig. 1, includes two main components—mesh saliency calculation and free form deformation. In the saliency calculation step, visually important regions are determined and in the free-form deformation step these regions are enhanced to obtain the caricaturized style.

3.1 Mesh Saliency Calculation

Saliency is the property of objects that attracts attention based on the difference of the object from its surroundings. In Perceptual Caricaturization of 3D Models, for the mesh saliency calculation, a center-surround mechanism proposed by Lee [7], which is a perceptual approach, is applied to determine salient parts of an object. Figure 2 shows results with colorized saliency map of three different models.

Mesh saliency calculation process as follows:

- The first step of saliency calculation involves computing surface curvatures. To compute the curvature of a mesh at a vertex v, the approach of Meyer [9] is applied.
- The Gaussian-weighted average of the mean curvature of a vertex is calculated with the Eq. (1).

$$G(C(v), \mathrm{s}) = \frac{\Sigma_{x \epsilon N(v,2\mathrm{s})} C(x) \exp[-||x-v||^2/(2s^2)]}{\Sigma_{x \epsilon N(v,2\mathrm{s})} \exp[-||x-v||^2/(2s^2)]} \tag{1}$$

  C(v) denotes the mean curvature of vertex v and N(v, 2s) is the neighborhood function which gives the set of points within the distance s for a vertex v.
- Then the computation of the saliency S(v) of a vertex v is calculated by taking the absolute difference between the Gaussian-weighted average computed at fine (small) and coarse (large) scales in Eq. (2)

$$S_i(v) = |G(C(v), s_i) - G(C(v), 2s_i)| \tag{2}$$

3.2 Free Form Deformation

Fundamentally, FFD is a kind of deformation of an object that changes its originally modeled appearance which is known as the object's rest state to other state which is known as deformed state.
To define the lattice space which can be called bounding box in free form deformation technique, bezier volumes are used. Bezier volume defines a volume with the control points within it and can be freely altered to deform the object. Equation (3) shows the calculation for the Bezier volume.

$$Q(\mathrm{u}, \mathrm{v}, \mathrm{w}) = \Sigma_{i=0}^{3} \Sigma_{j=0}^{3} \Sigma_{k=0}^{3} P_{ijk} B_{i,3}(u) B_{i,3}(v) B_{i,3}(w) \tag{3}$$

3.3 Integration

At this stage, the pre-calculated saliency values of the 3D object are mapped to the control points of the bounding box surrounding the object. Then the control points are extended from an estimated pivot points according to new calculated saliency values. To accomplish this, the algorithm works in three steps:

1. The Eq. (4) calculate the saliency value for one control point by computting the weighted average of the saliency values (s_v(i)) of the vertices of the object within an estimated distance threshold.

$$S_c(j) = \frac{\Sigma_{i=0}^{m} \frac{S'_v(i)}{d(P'_v(i)P_c(j))^2} * (\max(d) - \min(d))^2}{m} \tag{4}$$

In the equation, P_c(j) gives the position of the control points. S'_v(i) is the saliency values and P'_v(i) are the positions of the objects vertices under the estimated distance threshold. If the distance between the control point and vertex is smaller than the threshold, the saliency value of this vertex is taken into account for calculation of the saliency value of the control point.

$$P'_v(\mathrm{i}) = \{P_v(\mathrm{i}) | \mathrm{d}(P_v(\mathrm{i}); P_c(\mathrm{j})) < \mathrm{distThres}\}$$
$$S'_v(\mathrm{i}) = \{S_v(\mathrm{i}) | \mathrm{d}(P_v(\mathrm{i}); P_c(\mathrm{j})) < \mathrm{distThres}\}$$

2. Pivot points are calculated to determine the exaggeration direction of the control points. Equation (5) shows the calculation of the pivot point position for one control point.

$$V_c(j) = \Sigma_{i=0}^{n} \frac{P'_v(i)}{n} \tag{5}$$

3. After saliency values of all control points are calculated with the Eq. (4) and virtual pivot points of them are computed with the Eq. (5), Eq. (6) gives the calculation for the new positions of the one control point.

$$P'_c(j) = \frac{F_d * (P_c(j) - V_c(j)) * S_c(j)}{\max(S_c) - \min(S_c)} \tag{6}$$

In the equation, F_d is the deformation factor which is the only parameter that can be changed by user is used to determine the scale of the exaggeration of the caricaturized 3D model.

4 Results and Conclusion

Figures 3 and 4 show the results of our technique on different 3D models. Each figure starts with the saliency colorization of the models and through (b) to (d), the deformation factor F_d is incremented respectively. As the deformation factor is increased, the salient parts of the models are far more deformed and results in more exaggerated caricaturized models.

In the Perceptual Caricaturization of 3D Models, we have implemented the exaggeration concept by using free form deformation technique with respect to the mesh saliency by trying to extract perceptually more attractive features of the 3D objects to imitate an artist thinking when drawing caricature. This technique can be extended to generate multiple caricatures which can be employed in caricature animation and non-photorealistic animation in the future.

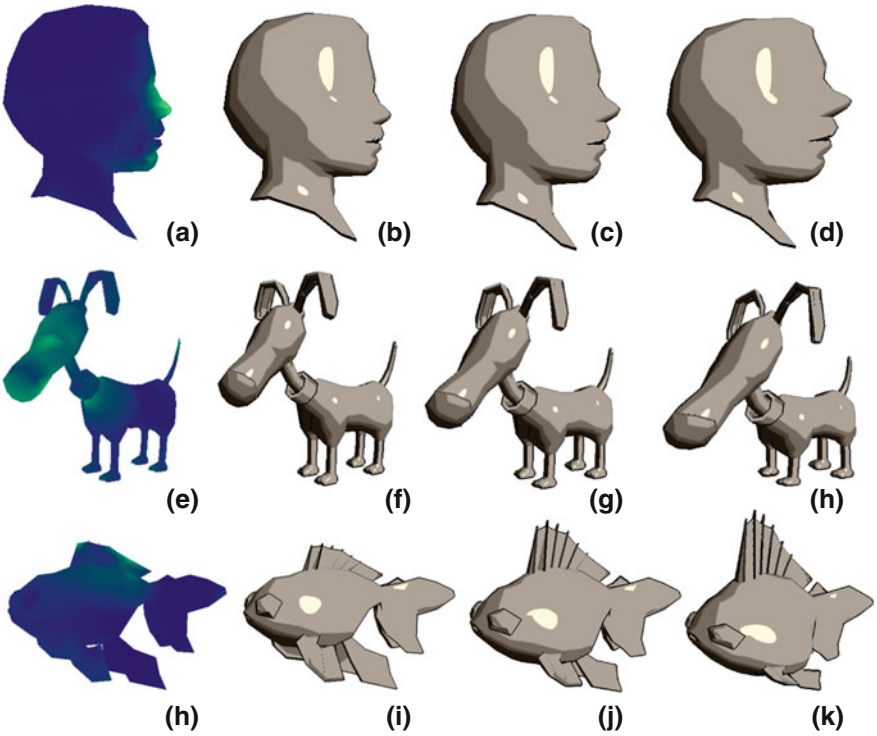


Fig. 3 Image **a** shows saliency. Images from **a** to **d** shows deformation at different deformation factors

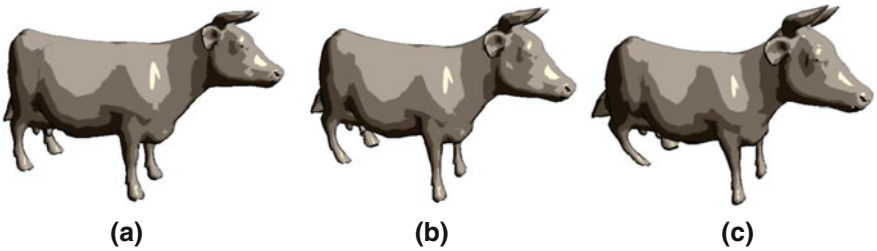


Fig. 4 Images from (**a**) to (**c**) shows deformation at different deformation factors

Acknowledgments We thank Ufuk Celikcan and Bengu Kevinc for their support during our study. This work is supported by the Scientific and Technical Research Council of Turkey (TUBITAK, Project number: 110E029).

References

1. Akleman, E.: Making caricatures with morphing. In: ACM SIGGRAPH 97 Visual Proceedings: The Art and Interdisciplinary Programs of SIGGRAPH'97. SIGGRAPH'97, p. 145 (1997)
2. Akleman, E., Reisch, J.: Modeling expressive 3D caricatures. In: ACM SIGGRAPH 2004 Sketches. SIGGRAPH '04, p. 61 (2004)
3. Brennan, S.E.: Caricature the generator: faces dynamic exaggeration of by computer. Leonardo **18**, 170–178 (1985)
4. Chen, H., Xu, Y.Q., Shum, H.Y., Zhu, S.C., Zheng, N.N.: Example-based facial sketch generation with non-parametric sampling. In: ICCV '01, pp. 433–438 (2001)
5. Clarke, L., Chen, M., Mora, B.: Automatic generation of 3D caricatures based on artistic deformation styles. Vis. Comput. Graph. IEEE Trans. **17**(6), 808–821 (2011). doi: 10.1109/TVCG.2010.76
6. Fu, G., Chen, Y., Liu, J., Zhou, J., Li, P.: Interactive expressive 3D caricatures design. In: Proceedings of Conference IEEE Multimedia and Expo, pp. 965–968 (2008)
7. Lee, C.H., Varshney, A., Jacobs, D.: Mesh saliency. ACM transactions on graphics. Proc. SIGGRAPH **24**(3), 659–666 (2005)
8. Liang, L., Chen, H., Xu, Y.Q., Shum, H.Y.: Example-based caricature generation with exaggeration. In: 10th Pacific Conference on Computer Graphics and Applications PG'02, p. 386 (2002)
9. Meyer, M., Desbrun, M., Schroder, P., Barr, A.H.: Discrete differential-geometry operators for triangulated 2-manifolds. Vis. Math. III **3**(7), 35–57 (2003)
10. Mo, Z., Lewis, J.P., Neumann, U.: Improved automatic caricature by feature normalization and exaggeration. In: Proceedings ACM SIGGRAPH 2004 Sketches, p. 57 (2004)
11. Sederberg, T.W., Parry, S.R.: Free-form deformation of solid geometric models. Proc. ACM SIGGRAPH **20**, 151–160 (1986)

Face Alignment and Recognition Under Varying Lighting and Expressions Based on Illumination Normalization

Hui-Yu Huang and Shih-Hang Hsu

Abstract In this paper, we propose an efficient approach to perform face alignment and recognition under lighting and expression conditions based on illumination normalization. For face representation, lighting influence and variable expressions, especially the accuracy of facial localization and face recognition, are the important factors. Hence, the proposed approach aims to overcome these problems. This approach consists of two parts. One is to normalize illumination for face image. The other is to extract feature by means of principal component analysis and recognize face by means of support vector machine classifiers. Experimental results demonstrate that our approach can obtain a good facial alignment and face recognition with varying lighting, local distortion, and expressions.

Keywords Gabor-based filter · Improved active shape model (IASM) · Principal component analysis (PCA) · Face alignment · Face recognition · Support vector machine (SVM)

1 Introduction

It is well known that the variations of illumination may change face appearance dramatically so that the variations between the images of the same face will cause the mistake recognition. Hence, there are many studies have been worked this effect on face recognition recently [1–5]. If these factors are considers, the face recognition rate can be improved and be more robust.

Makwana [2] proposed a survey of passive methods for illumination invariant face recognition. Authors discussed some methods, such as subspace-based statistical methods, illumination invariant representation methods, model based methods, etc.

H.-Y. Huang (✉) · S.-H. Hsu
National Formosa University, 64, Wun-Hua Road, Huwei, Yunlin 632, Taiwan
e-mail: anne.huang@ieee.org; hyhuang@nfu.edu.tw

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_22,

Marcel et al. [3] proposed a combined ASM with the different local binary patterns (LBP) method to address the problem of locating facial features in images of frontal faces. Lin et al. [4] proposed a face recognition scheme using Gaborface-based 2D-PCA classification based on 2D Gaborface matrices instead of transformed 1D feature vectors. In addition, in order to detect illumination effect, Zhang et al. [5] proposed a wavelet-based face recognition method to reduce lighting factor.

Owing the effect of lighting factor, it is very challenging problem for face alignment and recognition, in this paper, we propose an approach to against lighting conditions based on template matching method for précising location of facial features. In face recognition, we combine the support vector machine to classify and still keep a good recognition rate.

The remainder of the paper is organized as follows. Section 2 presents the proposed method. Experimental results and performance evaluation are presented in Sect. 3. Finally, Sect. 4 concludes this paper.

2 Proposed Method

Facial localization of landmark feature points often suffers from a variety of illumination and occlusion influences, in order to reduce these factors, we propose our approach to solve these problems.

2.1 Normalizing Illumination

The problem of illumination variation is usually existent and an important factor in the study of face recognition. Recently, the diversification of light conditions under the theme of face recognition [2] indicated, feature extraction is imperfect in the case of light exposure or low light whether the Gaussian filter or histogram equalization method were used. Hence, in this paper, we will propose an effective method to solve this factor to improve the recognition rate and localization accuracy of face image.

The advantage of wavelet transform is to overcome the limitations of traditional Fourier transform, so that it can work in time domain and frequency domain to analyze the data. Gabor filters, which are generated form a wavelet expansion of the Gabor kernels [4], exhibit desirable attributes of spatial locality and frequency domains optimally. In this paper, we use the Gabor-based wavelet filter to modulate the orientations and frequencies in order to reduce the lighting influence. The Gabor-based wavelet filter is defined as

$$\varphi_{\mu,v}(z) = \frac{||k_{\mu,v}||^2}{\sigma^2} \mathrm{e}^{\left(-\frac{||k_{\mu,v}||^2||z||^2}{2\sigma^2}\right)} \left(\mathrm{e}^{(ik_{\mu,v}z)} - \mathrm{e}^{\left(-\frac{\sigma^2}{2}\right)}\right), \tag{1}$$

where μ and v denote the orientation and scale of the Gabor kernels, $z = (x, y)$ including φ (frequency) and σ (bandwidth) parameters. The wave vector $k_{\mu,v}$ is defined as $k\mu, v = kv \cdot e^{i\phi\mu}$, where $k_v = \frac{k_{\max}}{f^v}$ and $\phi_\mu = \frac{\pi\mu}{8}$. $k_{\max}$ is the maximum frequency, and f is the spacing factor between kernels in frequency domain. The term $k_v = 2^{-\frac{v}{2}}(\frac{\pi}{2})$ represents each scale value is Gabor-based wavelet transform. The term $-e^{\left(-\frac{\sigma^2}{2}\right)}$ represents the deduction illumination noise.

Assuming that the image is filtering by Gabor filter within different scales and phase angles, the convolution operation is performed and expressed as

$$G_{\mu,v}(x, y) = f(x, y) * \varphi_{\mu,v}(x, y) \tag{2}$$

where $f(x, y)$ represents the input image, $\varphi_{\mu,v}(x, y)$ represents 2-D Gabor filter.

Generally, considering the frontal face image processing using Gabor filter, the orientation (μ) divided into eight different phase angles, and the scale (v) classified five different scales to obtain forty kernels in Gabor filter. Using these forty kernels to filter the frontal face image, we can get forty feature images and then compute the average feature image [5] by Eq. (3).

$$O(x, y) = \frac{1}{N} \sum_{i=0}^{4} \sum_{\theta=0}^{7\pi/8} G_{i,\theta}(x, y), \tag{3}$$

where N denotes the total number of kernels.

For frequency field, the characteristics of high-frequency information represent the contour and texture features of an image. However, these properties are important information in face recognition. Hence, in this paper, we utilize local Gabor filter to extract face features, it not only reduces the computational time in storage for the amount of data but also decreases the extracting time of features.

Owing to Gabor filter has the characteristic of angle symmetry, in order to avoid the redundancy operation and retain the characteristic of high-frequency information, here, we chose the orientation in the range [90, 180] and three smaller scales to cover more high frequency signal. Assuming an original image is of size 46×56 pixels, in our experiments, we select the last four orientations $\mu = (4, 5, 6, 7)$ and three scales $v = (0, 1, 2)$ to retain the important information and perform our experiments.

2.2 Hierarchical Image Blending

In addition, the average feature image obtained which usually needs to complex computation, hence, we propose image blending technology to overcome this problem, at the same time, this technology can retain much edge information to against illumination factor. About image blending technology, we take the real-part response of the Gabor filter and hierarchical image blending to recover the lost edge features.

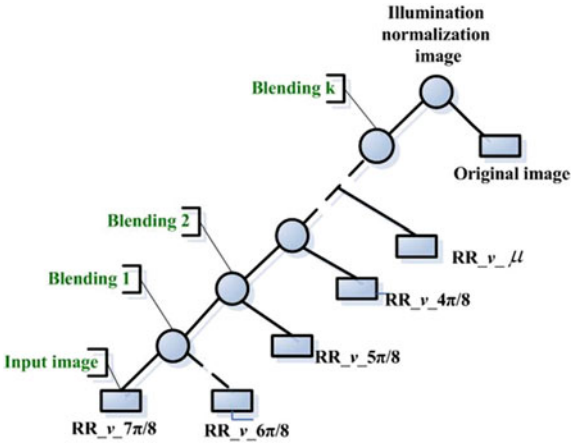


Fig. 1 Schema of hierarchical image blending process. The real response image (RR), scale (v) and orientation (μ) in Gabor filter are formed as RR_v_μ

Generally, image blending [6] is used to the source and destination images to synthesize a new image. According to Eq. (4), it makes the pixels to interact blending to get the virtual image (I_v).

$$I_v = (1 - M)I_s + M \times I_t, \quad 0 \leq M \leq 1, \tag{4}$$

where I_s and I_t represent pixel value on source image and destination image, respectively. Parameter M denotes the percentage of interaction process.

After adequately adjusting interpolation parameter M for the source image and destination image, we can obtain the blended image. Thus an illumination normalization image can be achieved. Figure 1 shows a hierarchical diagram of image blending; the determination of parameter M is depended on the user. In this study, it is set 0.5.

2.3 Aligning Processing

Active shape models (ASM) [3] is a model-based feature matching method to constrain the shape of an object in an image. The ASM is primary based on the shape of objects as training samples, and then uses this information to find the best match of the mean shape model to the data in a new image and to obtain the transformation matrix about deformation process.

In this paper, we use the template with 68 landmarks as the align feature points, assuming a given training sample set $\Omega = \{X_1, X_2, \ldots, X_N\}$ where N is the number of training samples, X_i is the shape vector with (x_i, y_i) coordinate, the coordinates of all feature points are concatenated into 2×68-dimensional vector

represented as X

$$X = (x_1, x_2, \ldots, x_{68},\ y_1, y_2, \ldots, y_{68})^T, \tag{5}$$

where (x_i, y_i) are as coordinates of the ith landmark in a face image.

Then, we sum up feature points of all training images denoted as a vector, next the coordinates about those of points are adjusted to the centroid of feature points. And the training sets are aligned to the specified reference image selected from the training sample, thus it can make the scale regularization. After alignment processing, the principal component analysis can be presented the aligned average shape vectors. The mean of the n aligned shapes (X_i) is expressed as $\overline{X}$

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i, \tag{6}$$

and the covariance matrix S is defined as

$$S = \frac{1}{n-1}\sum_{i=1}^{n} (X_i - \overline{X})(X_i - \overline{X})^T \tag{7}$$

The eigenvalues and the corresponding eigenvectors of the covariance matrix S denote as $(\lambda_i, \ldots, \lambda_s)$ and $(p_i, \ldots, p_s)$, respectively. The first t eigenvalues satisfying the $\sum_{i=1}^{t} \lambda_i \geq \alpha \sum_{i=1}^{n} \lambda_i$ are selected, where α is a selected feature ratio within the total number of features. Here, it is set 0.95 to 0.98. The recording first t eigenvectors can be formed as a matrix $\phi = (p_{1,} p_2, \ldots, p_t)$. Finally, the shape vector can be obtained as the following formula.

$$X = \overline{X} + \phi b, \tag{8}$$

where b is the eigenvector corresponding to the formation of the shape parameter set. The allowance of the similar shape is in the range of $-3\sqrt{\lambda_i} < b_i < 3\sqrt{\lambda_i},\ \ i \leq t$.

After processing training model, the mean shape and the transformation matrix can be obtained and further applied to search the facial features. The mean shape will be projected on the target area by using a two-dimensional structure profile to accurately locate to feature points; Fig. 2 shows a two-dimensional profile diagram for the location of feature points.

After adjusting all feature points, in order to refine the location of feature points, we use an adaptive affine transform (AT) of global shape model to update position accuracy of feature points. The adaptive affine transform is defined as Eq. (9). By this way, let the initial position of next search be close to the positioned place. At the same time, it can modify the shape parameter b which could be the shape model more fitting the updated feature points. By iterations of the above procedure until model convergence, then we can get the consistent with the shape which fits to the current target. For the adaptive affine transform procedure, the scale value (s_c) and rotation angle (θ_c) are firstly computing corresponding to the reference image x_{image}

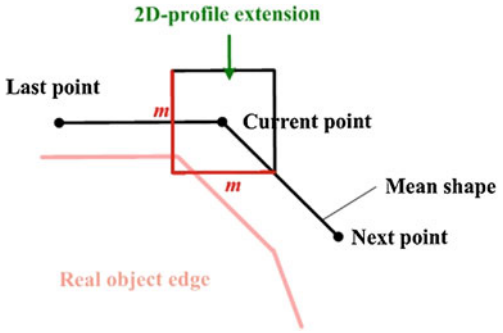


Fig. 2 Face alignment using two-dimensional profile

and test image y_{image} by using affine transform, respectively. Then the scale factor and rotation angle by using the Eq. (9) can be obtained. The x_c and y_c denote the gravity difference between x_{image} and y_{image}, respectively. The s is a scalar factor corresponding to affine transform.

$$AT\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} x_t + x_c \\ y_t + y_c \end{bmatrix} + \begin{bmatrix} s \cdot s_c \cos(\theta + \theta_c) & s \cdot s_c \sin(\theta + \theta_c) \\ -s \cdot s_c \sin(\theta + \theta_c) & s \cdot s_c \cos(\theta + \theta_c) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{9}$$

For the current feature point (x, y), the $\theta + \theta_c$ is the angle of rotation and $s \cdot s_c$ is the scale factor, the displacement unit is $(x_t + x_i)$ and $(y_t + y_i)$. Next, in order to calculate feature points of image, we utilize the 4-level multi-resolution pyramid strategy to search the number of rectangular side of the sampling points, and simultaneously to update the shape parameter b until the shape fits the model as a new point. Until over 95 % feature points in *1/2* image are found or equal to the number of iterations of the maximum. In our experiments, the number of iterations of the termination condition is set 24.

2.4 Recognition Processing

Based on radial basis function (RBF) kernel function for SVM, it is only necessary two parameters (cost function (c) and test kernel function (*Gamma*)) to adjust the model calibration. Because the input vector value is stayed in the range [0, 1], the system can greatly reduce the complexity computation and has a high predictive ability. However, in order to avoid improper selection of parameters which may be easy to cause the over-fitting occurrence, here, we adopt K-fold cross-validation method [7] to evaluate the classification performance. All samples are divided into training set and test set.

Table 1 Dataset with two database

Dataset	JAFFE	Yale_B
Total samples	200	2280
Class	10	38
Sample no. of each class	20	60

3 Experimental Results

In our experiments, we adopt the JAFFE [8] database including rich expressions and Yale_B [1] including a variety of lighting conditions to estimate the system performance. Table 1 shows the total number of samples in the database, number of in a class, and the number of samples in a single-class. The experiment procedures are firstly splitting the samples into the training and test classes by means of 10-fold cross-validation method and then to compute the average recognition rate.

The rule of 10-fold cross-validation strategy is that all samples are divided into 10 parts in which the nine-tenths of samples as training set and the rest part as an identification of test set, after operating ten times, the average recognized results can be more credible. In order to evaluate the difference of facial feature points location between ASM and IASM, we adopt the average localization error (E) to measure, and it is defined as

$$E = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} |P_{i,j} - P'_{i,j}|, \tag{10}$$

where $P_{i,j}$ denotes the jth manually-labeled feature point in the ith test image from m samples. $P'_{i,j}$ denotes the correspond fitting position by ASM searching. In addition, based on Eq. (10), the improved ratio (I) is used to present the improved percentage of the proposed IASM algorithm compared with that of ASM method. It is expressed as

$$I = \frac{E_{ASM} - E_{IASM}}{E_{ASM}} \times 100\,\%. \tag{11}$$

In Eq. (10), when I is positive, the proposed IASM method is better than the ASM method.

In this paper, we adopt two labeled modes for face image to estimate the performance of the location of feature points, mode 1 (Database_1) is selected manually feature points positioned on the two eyes shown in Fig. 3a, mode 2 (Database_2) is used the Viola-Jones (V-J) detector [9] to detect and locate the face region shown in Fig. 3b. Table 2 shows the localization errors (E) and improved result for JAFFE and Yale_B database compared ASM to IASM. From Table 2, it is evident that our proposed IASM method has the significant improvement effect in JAFFE or Yale_B database. Figure 4 presents the located results for some of cases of JAFFE database compared ASM to IASM.

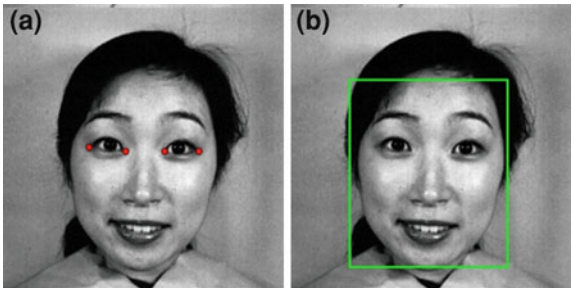


Fig. 3 Mode of locations. **a** Mode 1 locating on the two eyes, **b** mode 2 V-J face location

Table 2 Improved performance with different database

Mode	*E*		*I* (%)
	ASM	IASM	
JAFFE_1	3.72	**2.13**	42.74
JAFFE _2	11.63	**5.33**	54.14
Yale_B_1	5.56	**4.07**	26.79
Yale_B _2	30.44	**10.98**	63.92

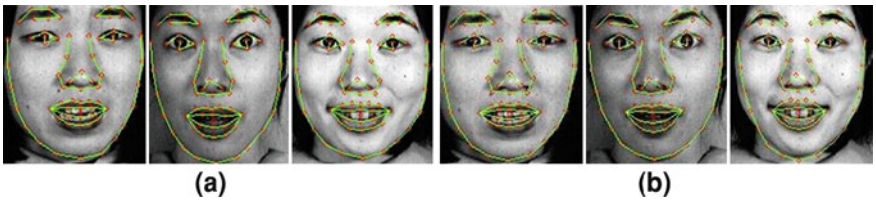


Fig. 4 JAFFE face localization comparison of ASM (**a**) and IASM (**b**). **a** ASM result **b** IASM result

For recognition process, we use the receiver operating characteristic (ROC curve) [10], as shown in Fig. 5, to analyze the recognizing results. The ROC is a graphical plot of sensitivity, or true positive rate versus false positive rate for a binary classifier system. For Fig. 5, the recognition result of Yale_B after processing illumination normalization can increase the sensitivity. Table 3 presents the recognition accuracy rate, and mean execution time (MT) compared whether illumination normalization or not. After processing illumination normalization, recognition rate can be increased to 83.33 %.

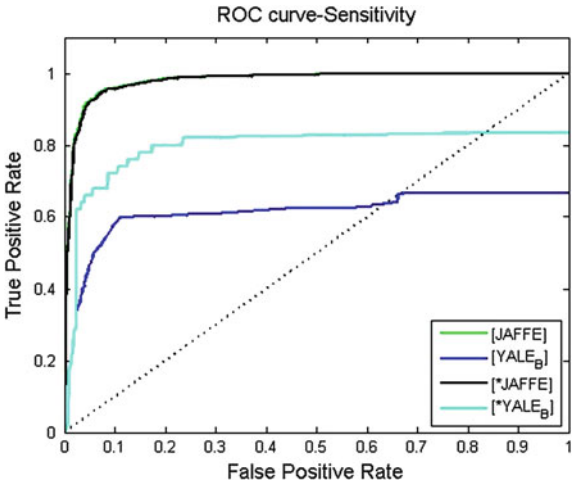


Fig. 5 Recognition results using the ROC curve

Table 3 Recognition evaluation

Database	Samples	Accuracy (%)	*MT* (*s*)
JAFFE	200	100.0	0.371
Yale_B	2280	66.71	6.254
***JAFFE**	**200**	**100.0**	**0.385**
***Yale_B**	**2280**	**83.33**	**4.660**

Symbol * indicates that the database has been processed by illumination normalization

4 Conclusions

In this paper, we have presented a face alignment and recognition under varying lighting conditions and multi-expressions based on illumination normalization. The purpose of this system is to align the facial features more exactly and to improve the recognition rate. Experimental results show that our proposed method is feasible and efficient to achieve face alignment and recognition.

Acknowledgments This work was supported in part by the National Science Council of Republic of China under Grant No. NSC99-2221-E-150-064.

References

1. Georghiades, A.S., Kriegman, D.J., Belhumeur, P.N.: From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intel. **23**, 643–660 (2001)
2. Makwana, R.M.: Illumination invariant face recognition: a survey of passive methods. Procedia Comput. Sci. **2**, 101–110 (2010)
3. Marcel, S., Keomany, J.: Robust-to-illumination face localization using active shape models and local binary patterns. IDIAP Research Report, pp. 6–47 (2006)
4. Lin, W., Li, Y., Wang, C., Zhang, H.: Face recognition using Gabor face-based 2DPCA and (2D) 2PCA classification with ensemble and multichannel model. In: Proceedings of IEEE Conference of Computational Intelligence in Security and Defense Applications, pp. 1–6 (2007)
5. Zhang, T., Fang, B., Yuan, Y., Tang, Y.Y., Shang, Z., Li, D., Lang, F.: Multiscale facial structure representation for face recognition under varying illumination. Pattern Recogn. **42**, 251–258 (2009)
6. Ling, X., Wang, Y., Zhang, Z., Wang, Y.: On-line signature verification based on Gabor features. In: Proceedings of the 19th Annual Wireless and Optical Communications, pp. 1–4 (2010)
7. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of Fourteenth International Joint Conference on Artificial Intelligence, pp. 1137–1143 (1995)
8. Lyons, M.J., Akamasku, S. Kamachi, M., Gyoba, J.: Coding facial expressions with Gabor wavelets. In: Proceedings of International Conference on Automatic Face and Gesture Recognition, pp. 200–205 (1998)
9. Viola, P., Jones, M.: Robust real-time object detection. Int. J. Comput. Vis. **57**, 137–154 (2004)
10. Fawett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**, 861–874 (2006)

Part VI
Communication Systems

Fixed-Mobile Convergence in an Optical Slot Switching Ring

J.-M. Fourneau and N. Izri

Abstract We study the convergence of fixed and mobile traffic in the time-slotted ring. We analyse the manner in which traffic stemming from mobile and fixed networks share the capacity of the metro network. We propose several approaches for the convergence of fixed and mobile traffic and analyse their performance in terms of delay and bandwidth efficiency while using two traffic models. Our goal is to determine whether it is necessary to separate fixed traffic from mobile traffic, giving priority to the latter. We show that as long as the CoS of packets is taken into account, fixed and mobile traffic can be mixed inside the same time-slot without impairing the QoS requirements of delay-sensitive applications.

1 Introduction

Next-generation networks are faced with the convergence of mobile and fixed services, i.e. the ability to carry multiple types of media stemming from different access points onto the same network [5]. We present a manner of providing Fixed-Mobile Convergence (FMC) in a time-slotted WDM (Wavelength Division Multiplexing) ring suitable for the metropolitan area. The wavelength channels are divided into time-slots of equal duration on a synchronous basis [8]. Resources are thus allocated and switched at time-slot granularity.

In this paper, we investigate whether a simple QoS-aware assembly scheme is sufficient to fulfill the latency requirements of mobile traffic in the context of FMC. We analyse the manner in which traffic stemming from mobile and fixed net-

J.-M. Fourneau · N. Izri (✉)
PRiSM—MR 8144, Université de Versailles-St-Quentin,
45, Avenue des Etats-Unis, 78035 Versailles, France
e-mail: noiz@prism.uvsq.fr

J.-M. Fourneau
e-mail: jmf@prism.uvsq.fr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_23,

works share the capacity of the optical ring and whether a CoS-aware assembly scheme is able to provide a satisfactory QoS level. We also evaluate the impact of the resource allocation mechanism on the network performance. We considered a reservation-based one with two approaches: (1) *dedicated optical containers (OCs)* are allocated to each type of traffic and (2) mobile and fixed traffic is *mixed* inside the same OC while considering the CoS of data packets. We show that it is not necessary to separate fixed and mobile traffic in order to give priority to the latter. This paper is organized as follows. Section 2 presents the network architecture, the data packets aggregation into OCs, and the considered access mechanism. Then, we describe the associated assembly scheme as well as the proposed FMC approaches. Section 3 presents the performance analysis of the proposed approaches by simulation.

2 Network Description and Traffic

We consider a time-slotted WDM ring consisting of $N+1$ stations. Each network link carries W *data* wavelength channels and a separate *control* wavelength dedicated to carrying control information such as slot reservation status. This information is sent via control packets. One control packet is associated to time slot across all W data wavelengths. The considered network is able to provide subwavelength granularity by means of Optical Slot Switching (OSS) [4]. The network consists of access nodes that connect the access networks to the ring and a specific hub node that connects the ring to the backbone network. These nodes are interconnected by unidirectional fiber links (the other direction is used for safety), each carrying multiple wavelength channels. The metro ring needs to provide a unified manner of carrying traffic originating from both fixed and mobile networks. The network nodes are Optical Packet Add-Drop Multiplexers (OPADM). An OPADM is able to insert or to extract data to/from a specific wavelength channel. It can also transparently forward the transit traffic to downstream nodes. A specific station, say station 0, corresponds to the *hub* station that allocates the time-slots. The other N stations are access nodes in charge of connecting the access networks, and thus the end-users, to the metro ring.

Each node i is equipped with $t_i \in \{1, \ldots, W\}$ tunable transmitters and $r_i \in \{1, \ldots, W\}$ fixed receivers. In this way, each station can simultaneously transmit data on t_i different wavelengths and receive data on r_i wavelengths. In our simulations, we assume that access nodes are equipped with a single tunable transmitter and a single fixed receiver, i.e. $t_i = r_i = 1$, for $i \in \{1, \ldots, N\}$, while the hub station has W fixed receivers and W fixed transmitters meaning that it can simultaneously send and receive on all wavelengths. Time is slotted so that there are S slots circulating on the ring, visiting nodes in a cyclic manner in the order $0, 1, \ldots, N$. Each slot carries a single optical payload that we refer to as OC. An OC's size is given by the slot duration times the transmission speed on each wavelength, that is 12.5 KB for a transmission speed of 10 Gbps. It is equivalent to 8 Ethernet Maximum Transmission

Units (MTU). A data packet is emitted within an OC on a wavelength, it does not change its OC during the optical transport.

We consider three distinct classes of service based on the QoS requirements of the different applications supported by our network. **Interactive Real Time** (IRT): this class includes applications that require a short response time and low jitter, such as voice over IP. These applications are generally characterized by low throughput. **Simple Real Time** (SRT): this class concerns applications such as video or TV on demand. It is characterized by high throughput requirements and long duration. The jitter and loss rate remain important parameters but they are not critical. **Non Real Time** (NRT): this category includes services that do not have time constraints. It concerns applications that are delay and/or jitter tolerant, such as FTP transfers or e-mails. We consider IRT to be the highest priority class, while NRT has the lowest priority. The CoS is considered during the aggregation of data packets into the OCs. Specifically, we consider that every node has N waiting queues, that is, one waiting queue per destination. Each of them is then divided into 3 subqueues, one for each class of service. The queued data packets are then used to fill in the OCs. This assembly process is based on a CoS upgrade mechanism similar to the one proposed in [1]. OCs are firstly filled by electronic packets of the highest priority and the remaining capacity (if any) is then completed by lower priority data packets. The resulting OCs are classless; their role is simply to carry data among the network nodes. This leads to a simple and transparent transport layer, able to keep up with technological developments.

The access mechanism is a reservation-based in which resources are allocated to each node using a centralized reservation protocol introduced in [2], and a scheduler grants free time-slots to requesting stations. The reservation mechanism needs to ensure fairness among network nodes and provide continuous slot distribution based on the QoS requirements of each station. All reservation requests in the ring are managed by the hub. Each wavelength is logically considered as a circular sequence of slots. We define a *transmission window* to be a fixed-size set of consecutive time-slots on different wavelengths; we assume that one transmission window corresponds to half a cycle. When a station wants to transmit data on the ring, it first sends a request to the hub node indicating the number of slots required in each transmission window and the total number of required transmission windows. The hub will respond by allocating time-slots in each transmission window by set of rectangles that we refer to as *patterns*. Figure 1 (left) illustrates a pattern. Generally, a pattern is described by: $a(b \times c + d)$ where $b \times c$ represents the rectangle of allocated slots, d represents the free slots, while a is a repetition factor. More specifically, c is the number of wavelengths over which the slots are spread out (height of rectangle) and b is the number of successive allocated time-slots (length of rectangle).

A delay of at least 2 ms is required for the source to send its reservation request to the hub and receive the corresponding slot allocation pattern. The hub is responsible for finding the required number of slots with the best representation across all transmission windows; it tries to satisfy all reservation requests while minimizing the number of rejection on future requests, see [2] for details. Once it receives its allocated time-slots, the access station can fully utilize them to send its queued data

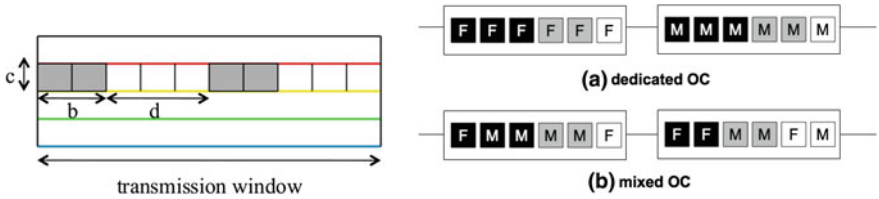


Fig. 1 Representation of pattern (*left*), Fixed-mobile convergence approaches (*right*) **a** dedicated oc. **b** mixed oc

packets. We note that no preemption is allowed in our system. No OC is displaced from its slot before reaching its destination, and each access station will use only the slots allocated by the hub station.

2.1 Fixed-Mobile Convergence Approaches

Different approaches to FMC are possible with respect to the manner in which the OCs are shared between fixed and mobile services. We can have either *mixed OCs* in which there is both fixed and mobile traffic (see right of Fig. 1b) or *dedicated OCs* for each type of traffic (see right of Fig. 1a). In the case of mixed OCs, the capacity of the OC can be shared either deterministically or non-deterministically. Static allocation of bandwidth is likely to lead to an inefficient utilization of the available capacity. On the other hand, non-deterministic sharing allows statical multiplexing of mobile and fixed traffic which is expected to enhance bandwidth utilization. Here, we only consider dedicated OCs and mixed OCs with non-deterministic sharing of the OC capacity. For the mixed OC case, we analyse two separate scenarios: a scenario in which mobile and fixed traffic are queued into individual waiting buffers and priority is given to mobile traffic and a scenario in which mobile and fixed traffic share the same waiting queues and priority is based only on the CoS of data packets.

2.2 Traffic Characteristics and Traffic Scenarios

For the sake of conciseness we report the results of the most realistic mixture of traffic. We assume that mobile traffic represents 20 % of the total traffic. We consider that 10 % of the total traffic is IRT, 60 % of the traffic corresponds to SRT, while the remaining 30 % corresponds to best-effort (NRT). The packet size depends on the type of traffic. In the simulations presented below, we assume that mobile packets have sizes of 46 and 53 bytes, while fixed traffic corresponds to sizes of 557 and 1,500 bytes. These packet sizes match traffics statistics shown in [6]. We use two models for the arrivals of data packets. ***Poisson model:*** We consider that for

Table 1 Parameters of the arrival processes

Distribution	ON IRT-1	OFF IRT-1	ON IRT-2	OFF IRT-2
Parameters	350 ms	650 ms	1 s	1.5 s
Distribution	ON SRT-1	OFF SRT-1	ON SRT-2	OFF SRT-2
Parameters	(5.08, 2.03)	(3.09, 1.43)	(4.29, 1.28)	(1.263, 0.008)

the three service classes, the data packets arrive according to independent Poisson processes. This represents a reasonable assumption for a metropolitan area network in which every access station is connected to potentially thousands of terminals [3]. ***ON/OFF model:*** The NRT arrivals follow a Poisson process as they model the superposition of a very large number of independent flows. IRT and SRT arrivals are more complex as it is usually assumed that they exhibit long range dependencies. According to [7], we model IRT arrivals by the superposition of independent sessions. Each session is modeled by an ON/OFF process and represents UMTS or VoIP. Both periods are distributed according to exponential distributions whose parameters are reported in Table 1. The SRT arrivals are also bursty and based on the superposition of independent ON/OFF sessions which follow a lognormal or a Pareto distribution. We implement a well known arrival models published in [10] to represent such a session. The parameters (mean and standard deviation in seconds) are also reported in Table 1. This process ON/OFF arrivals of packets is more realistic at the sources but it is much more complicated to simulate.

3 Simulation and Performance Analysis

The network was implemented in OMNET++ [9]. The number of slots S is 200 which corresponds to a cycle of 2 ms for a slot duration of 10 µs. Each cycle contains 2 transmission windows. Each simulation run lasts 10^4 transmission windows (10 s), with more than 8×10^6 electronic packets generated and 4×10^5 OC transmitted per source. We assume that we have 10 nodes in the ring and 4 wavelengths. The received wavelengths are allocated in a cyclic manner from node 1 to N. We consider in our simulations the ***Hub upstream scenario*** which is the worst configuration, because N access nodes send all of their traffic to the hub station resulting in an *all-to-one* traffic scenario. Since the hub station is able to receive on all W wavelength channels, the access nodes will compete for slots on all wavelength channels. The best performing patterns shown in [2], considers that the allocated time-slots are uniformly distributed inside window. We consider two such uniformly distributed reservation patterns, one for dedicated approach $D = 4 \times (2 \times 1 \times Mobile + 8 + 8 \times 1 \times Fixe + 7)$ and other $M = 4 \times (5 \times 1 + 7 + 5 \times 1 + 8)$ for mixed approach.

As the propagation time of the OCs is constant, we focus on the queueing delay incurred by data packets. Figure 2 shows the average delay of fixed packets, using the Poisson model (left of figure) and the ON/OFF model (right of figure). We

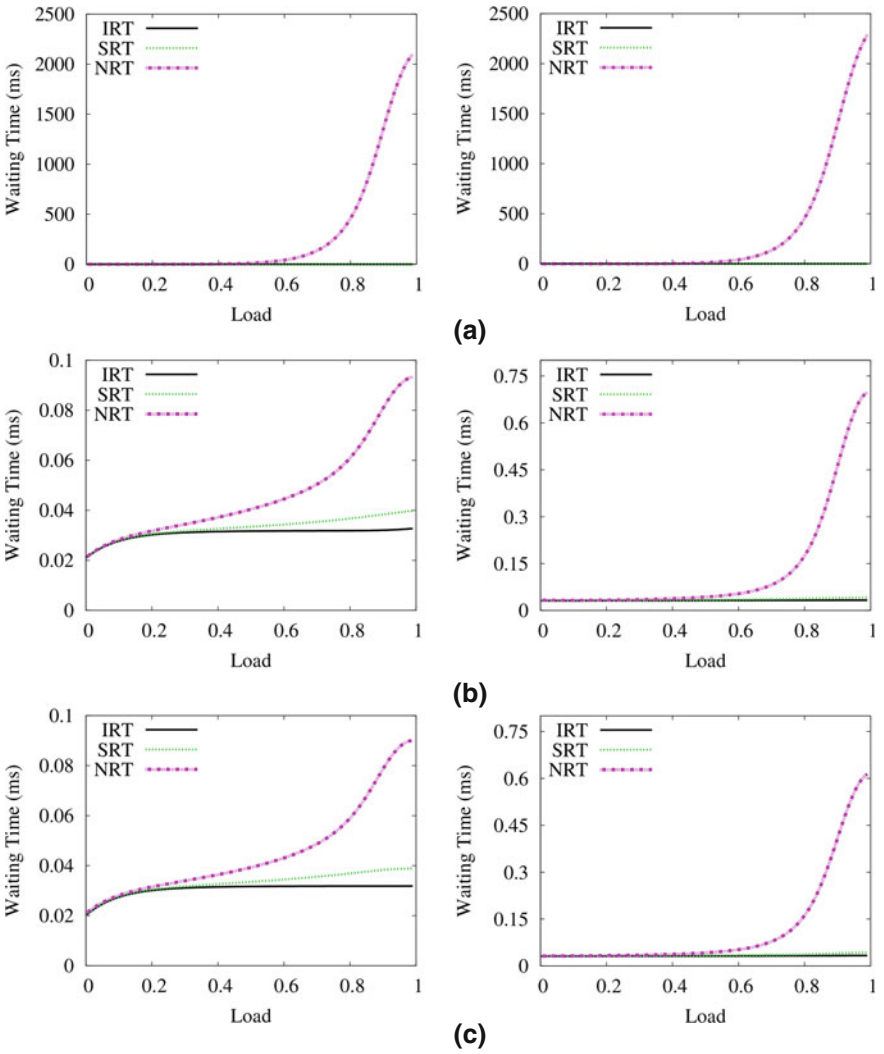


Fig. 2 Mean delay of fixed traffic by CoS using Poisson model (*left*) and ON/OFF model (*right*), for the three convergence approaches. In (a), IRT and SRT delays are confused with the Load axis **a** Dedicated. **b** Mixed with 6 subqueues. **c** Mixed with 3 subqueues

first observe that the delays of packets are several orders of magnitude higher in the dedicated OC (Fig. 2a). The delays of the lowest priority classes (SRT and NRT) increase considerably as the network load increases, indicating that the corresponding queues have a very large number of waiting data packets. By allowing fixed traffic to share the same OC as mobile traffic, the mixed OC is able to alleviate the contention of fixed traffic and thus reduce the waiting time (Fig. 2b, c). It is noteworthy that using either 3 or 6 subqueues for data packets will lead to quite similar results,

Table 2 Average waiting time of mobile packets by CoS in ms, using Poisson model

Pattern	No.Queue	IRT	SRT	NRT
D	6	0.068	0.092	0.116
M	6	0.031	0.038	0.08
M	3	0.030	0.031	0.038

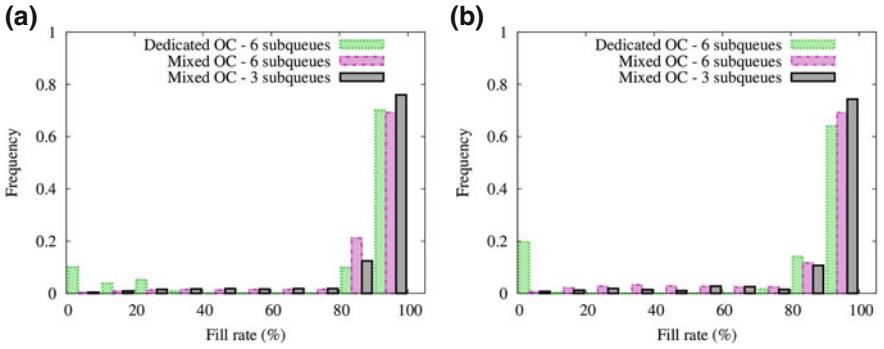


Fig. 3 Histogram of OC fill rate at a network load of 0.9 **a** Poisson model. **b** ON/OFF model

indicating that mixing fixed and mobile packets in the same waiting queues does not significantly impact the performance. Table 2 gives the mean delay of mobile packets for convergence approaches at a network load of 0.9. As indicated by the presented values, low delays can be provided for mobile traffic for any of the considered approaches even at high network load. We conclude that by using this centralized reservation scheme, it is possible to mix data packets originating from either fixed or mobile into the same waiting queues while still fulfilling the QoS requirements of mobile traffic. **However**, we showed that the dedicated OC may be interesting for low load of mobile traffic. We considered a same distribution of traffic as before, but with a load of 0.1 for mobile and 0.95 for fixed. We note that at low load of mobile traffic, the use of dedicated OC provides better results of waiting time for mobile packet CoS compared to mixed OC.

The fill rate of OCs constitutes another important performance indicator as it represents the level of bandwidth utilization. The average fill rate is simply equal to the network load, regardless of the number of per-destination subqueues used. In the dedicated OC, as the load of the network increases, data packets start to accumulate in the lower priority queues of fixed traffic, and the fill rate is smaller than the network load as not all of the offered traffic is being transmitted. Figure 3 shows the histogram of the deciles of the OC fill rate distribution. We observe that for the dedicated OC 10% of the OCs are nearly empty. The proportion of quasi-empty OCs is reduced to only 1 % in the mixed OC. Furthermore, fully-filled OCs represent approximately 78 % of the total number of optical payloads in the mixed OC.

The simulation results clearly indicate that as long as the centralized reservation is used at the access layer, mixing mobile and fixed traffic inside the same OC yields high performance in terms of OCs fill rate and delay.

We have replaced the centralized reservation scheme by a purely opportunistic access scheme and we kept the same evaluation metrics (average latency of data packets and OC fill rate). This opportunistic scheme is a fully-distributed access mechanism in which any node is allowed to use any unoccupied time-slot in order to send a waiting OC. This simple access scheme provides no fairness control and is likely to lead to high access delays. We showed by simulation the impact of the timer value when creating containers. Then, we investigate whether this simple access mechanism is able to transmit mixed OCs without impairing the QoS requirements of high-priority traffic. We have also investigated whether this result holds true for a purely opportunistic (i.e., greedy) access scheme. We showed that due to the lack of fairness of the opportunistic scheme, mixing fixed and mobile traffic within a single container is likely to impair the QoS requirements of delay-sensitive applications. Since the opportunistic access scheme does not implement any fairness control, the number of slots used by each station greatly depends on the station's position on the ring. Indeed, a node may be starved from free slots on a specific wavelength that serves a given destination. This node is then unable to satisfy the corresponding queue, leading to high access delay.

4 Conclusion

Our simulation results show that mixing fixed and mobile traffic within a same OC leads to higher bandwidth utilization, while yielding negligible packet delays for both fixed/mobile traffic, whatever the access protocol. We were also able to show that high performance level is achieved even when fixed/mobile traffic pertaining to a given CoS share the same FIFO queue.

Acknowledgments This work is supported by ANR ECOFRAME project (ANR06-TCOM002).

References

1. Bonald, T., Indre, R., Oueslati, S., Rolland, C.: On virtual optical bursts for QoS support in OBS networks. In the 14th International Conference on Optical Network Design and Modeling ONDM, Japan, 2010
2. Cadéré, C., Izri, N., Barth, D., Fourneau, J., et al.: Virtual circuit allocation with QoS guarantees in the ecoframe optical ring. In ONDM, Japan, 2010
3. Cao, J., et al.: Internet traffic tends toward Poisson and independent as the load increases. In Nonlinear Estimation and Classification, Springer, New York, 2002
4. Chiaroni, D., Neilson, D., Simonneau, C., et al.: Novel optical packet switching nodes for metro and backbone networks. In Invited ONDM, Japan, 2010

5. Fitzek, F., Sheahan, R., et al.: Fixed/mobile convergence from the user perspective for new generation of broadband communication systems, In CICT (2005)
6. Murkherjee, B.: Optical WDM Networks. Springer, Heidelberg (2006)
7. Naja, R.: Mobility management and resource allocation in multiservice wireless networks. Ph.D. thesis, 2003
8. Uscumlic, B., Gravey, A., Morvan, M., Gravey, P.: Impact of peer-to-peer traffic on the efficiency of optical packet rings, In WOBS (2008)
9. Varga, A., Hornig, R.: An overview of the OMNET++ simulation environment. In ICST Networks and Systems Workshops, Marseille, 2008
10. Veloso, E., Almeida, V., et al.: A hierarchical characterization of a live streaming media workload, In IEEE/ACM Transactions on Networking, (2006)

An Ultra-Light PRNG for RFID Tags

Mehmet Hilal Özcanhan, Gökhan Dalkiliç and Mesut Can Gürle

Abstract This work presents a pseudo-random number generator for RFID tags. The proposed generator takes non-random seeds and produces output that passes popular randomness tests. The generator uses true random numbers extracted from a newly discovered property of the tag memories. Unlike previous work, the proposed scheme uses very little die area and clock, leaving space for other security applications as well. The scheme is inspired from a well founded pseudo random number generator, which has many variations for computer security applications. Even though it is simple and sequential, the proposed scheme's performance equals similar previous work, even though others depend on random number inputs.

Keywords RFID, Low cost tags · Random numbers · Randomness tests · Ubiquitous

1 Introduction

Every consumer good in a supply chain has an identification sticker used to keep track of it, from its manufacture to the basket of a consumer. The traditional barcode stickers are being replaced by RFID tags [1]. According to a report RFID is a booming technology, as part of ubiquitous systems [2]. RFID rests upon wireless technology where a tiny integrated circuit is energized and inquired through its antenna coil, by a reader. The unique and sensitive identification number (ID) of the tag is acquired

M. H. Özcanhan · G. Dalkılıç (✉) · M. C. Gürle
Faculty of Engineering, Dokuz Eylül University, 35160 Izmir, Turkey
e-mail: dalkilic@cs.deu.edu.tr

M. H. Özcanhan
e-mail: hozcanhan@cs.deu.edu.tr

M. Can Gürle
e-mail: mesut.gurle@deu.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_24,

through air and sent to a remote database. The issue of security arises when the tag and the reader try to authenticate each other over the insecure, air medium. Generated pseudo random numbers (PRNs) and the tag ID are transmitted, in weakly obscured messages.

The low-cost passive-UHF tags are reportedly the most popular [2]; their main advantage being "capable of identifying items of different types and also distinguishing between items of the same type without mistake, while not requiring physical or visual contact" [3]. Low-cost is the reason of the popularity of passive tags but it limits the memory, computational capacity and energy consumption. Also little resources can be spared for security, causing vulnerabilities for attacks [4, 5]. Low cost tags should not be confused with the high cost e-passports tags.

After long efforts, the properties supported on passive tags have been ratified in ISO-18000-6 [6] and the EPCglobal Class-1 Generation-2 (Gen-2) standards [7]. PRNs used in many security algorithms are supported in the Gen-2 standard. Efficient Pseudo Random Number Generators (PRNGs) are needed where only a few thousand gates for security are dedicated [3, 8]. This is the motivation behind our proposal detailed in Sect. 3.

In the rest of this paper, Sect. 2 gives an account of related work. Section 3, reveals our proposed scheme. Section 4 contains the performance and testing results, followed by an evaluation and comparison of the results with similar works. We conclude and list future work, in Sect. 5.

2 Related Work

The work on generating random numbers in RFID tags can be divided into three categories. First is the work on True Random Number Generators (TRNGs), where a physical characteristic of the tag is used. Second category has the PRNGs, where a deterministic algorithm is used. The third is a blend of the first and second categories, where an output obtained from a TRNG is used as a seed of a PRNG. TRNGs mostly fail to provide good quality random numbers or tend to output the same numbers, if physical conditions are reproduced. In PRNGs on the other hand, the generated RN can be guessed if the initialization value or seed is guessed, as the whole process is mathematically deterministic. Therefore, a PRNG by itself is declared insecure without good sources for seeding [9, 10]; hence a source of true randomness is required for seeding [9]. Thus, TRNG outputs are used as inputs to PRNG algorithms in many works [11–13]. In reality, random numbers produced by some tags are shown to look alike, repeat or be predictable [13, 14]. Other schemes which produce good quality random numbers by using hashing and encryption algorithms overwhelm the limited capabilities of resource stricken RFID tags [15].

Two previous works support their schemes with widely used randomness tests [3, 8] and have detailed design and performance information, which we can compare with our work. Two other previous works falling into third category, use LFSR functions with only one bit TRN-input and do not provide the same information but

```
x := TRN
x := ROTl(x,x)
y := x ⊕ ROTr (x,u)
y := y ⊕ (ROTl (y,s) AND b)
y := y ⊕ (ROTl (y,t) AND c)
y := y ⊕ ROTr (y,l)
z := y ⊕ ROTl (y AND b, p)
u = 7Fh, s = 07h, t = 1Fh, l = 3FFFFh, p=7Fh.
b=9D2C568016 and c = EFC6000016.
```

Fig. 1 The proposed scheme, MeTuLR

concentrate on physical characteristics [12, 13]. Many attacks on LFSRs have been announced, but they are outside our scope.

Our work comes in at this point, which uses a TRNG extracting method that feeds a PRNG algorithm. The SRAM memory of a tag can be a source of TRNs [12]. Some bits settle randomly to an either low or high voltage level. The obtained TRNs have low entropy and fail the randomness tests. For this reason, the TRNs are fed to a hash function, which provide good quality PRNs. Assuming hashing is not suitable for low-cost tags; we attempt to replace it with an ultra-light scheme. The definition of "ultra-light" schemes and tags is given in [5].

3 The Proposed Scheme

Our work is inspired from Mersenne Twister (MT) which is a proven to be a good PRNG [16]. MT achieves fast generation of PRNs with a long period and has many variants [17]. Our work is based on using extracted TRNs from hardware as input seeds for the MT. We aim to remove the deterministic, iterative, pseudo part of MT and replace it with TRNG seeding. Our proposed scheme is shown in Fig. 1. It consists of simple bitwise XOR, AND and rotation (circular shift) operations replacing hashing in [11] which consumes less die area and clock cycles.

MT is a special case of Wells function [18]. In short, it is an algorithm which treats an input as a matrix and twists it right and left with corrective temperings to produce good PRNs. For a complete mathematical analysis, the reader is referred to the original documents [16]. All versions of MT pass the randomness tests.

The matrix iterations of the original MT are eliminated in our scheme, as the tag is not capable of performing them. To compensate, our scheme improves the shift operations with rotation operations. ROTr(x,y) is a simple bitwise shift operation to the right, where the least significant bit (LSB) is wrapped around to most significant bit (MSB). The value x is the number to be rotated and y is the hamming weight (number of ones in) of y. The operation is simple because y is loaded into a control register and if the tested bit is a one, x is rotated once.

Overall, rotation performs permutation while XOR, AND provide substitution. Thus, the input goes through a sequence of permutations and substitutions, as in modern hashing and encryption algorithms. The direction of rotations and coefficients of MT are carefully designed and well defended in [16, 18]. In our scheme, an extra rotation is necessary to compensate for the lost iterations, which brings little computational cost. Different number of rotation operations, different directions against different number of rotations and AND operator coefficients have been tested, until the scheme with the best randomness results was obtained. As it will be revealed, the original directions and masking coefficients yield the best results. The scheme can be executed as a sequential algorithm, where the coefficients are given as immediate operands.

4 Performance, Testing and Evaluation of Results

According to Gen-2 specifications, a tag supports only a 16-bit PRNG whose period is much shorter than a 32 bit PRNG. This short period and the overall security supported by Gen-2 are highly inadequate [3, 8]. But we claim that good quality, 32 bit PRNGs are possible in tags, as proven in Sect. 4.1. Moreover, only 32 bit tests are accepted, by the community. Our proposal uses 16-bit architecture to obtain 32-bit PRNs, because 32 or 64 bit production technologies cannot be used in low-cost tag production. We assume that 16-bit PRNs can be obtained from 32 PRNs easily, as discussed in works [3, 8].

The maximum number of gates and clock cycles allocated for security in tags are a few thousands gates and 1,800 clocks [4, 19]. Additional space and time are needed for authentication steps, as well. Keeping the above guidelines in mind, the performance and randomness test results are compared below.

4.1 Performance Results

Estimation for the die area of an integrated circuit can be obtained by using the gate equivalents (GE) [20]. The GE of each logic gate and the total GE for each operator are known [21]. The same GE metrics are used in the work that we compare our results. On the other hand, the common timing metric is the clock cycles used. A 16-bit ALU requires two clock cycles to finish a 32 bit AND or XOR operation; but n clocks for an n number of circular shifts. And by using the results of the above metrics, the area-delay product (i.e. complexity) can be determined.

Our scheme of Fig. 1 requires only three simple bitwise operations being executed, sequentially. The multiplication and the finite state machine (FSM) of Akari-x overload the tags. The Lamed scheme uses three operations like ours but requires extra input, control and rotation units for iteration loops. Both Akari-x and Lamed schemes

Table 1 Total gate equivalent of our proposal

Operator	# Used	Logic	GE	16-Bit total
Register	2	Flip flop	5.33	170.56
Shifter	1	Flip flop	5.33	85.28
AND	1	Gates	1.33	21.28
XOR	1	Gates	2.67	42.72
Total				319.84
Control	1	Gates	30 %	95.95
Grand total				415.79

Table 2 Comparison of performance results

Scheme	Area (GE)	Die area (μm^2)	Power consumption	Delay cycles	Throughput (Kbps)	Complexity (GE × Delay)
Akari1A	476	1494	47.35	644	24.24	306,544
Akari1B	524	1643	54.61	644	3.55	337,456
Akari2A	824	2582	57.38	466	31.37	383,984
Akari2B	891	2794	76.95	466	5.50	415,206
Akari2C	903	2831	72.33	466	3.01	420,798
Lamed	1566	4916	157.19	220	71.35	344,520
Ours	416	1306	41.75	72	18.95	29,952

require memory for an initialization vector (iv) [3, 8]. Lamed requires a total of two 32-bits space while ours requires three 32-bits.

The total GE required for our scheme is calculated to be 416 gates, as shown in Table 1. Every 1000 GE adds $0.01 to the cost and power consumption is proportional to the total number of gates [22]. With less than 500 gates our scheme is well in the low cost category [4].

In Table 2, the result is compared to previous work together with clock cycles used. The number of clock cycles spent while obtaining a random number is obtained from Fig. 1 as a total of five XOR, three AND operations and the sum of x, u, s, t, l, p number of rotations. The TRN, x is read from the SRAM into register during energizing the tag and is not in the RN generation phase. The XOR and AND cost a total of 16 clocks. The sum of u, s, t, l and p is constant, while rotation is dependent on the hamming weight of x. Assuming 16 ones on the average, the total of rotations is 56. Thus, the proposed 72 clock scheme is definitely in the ultra-light category [4].

Table 2 shows that the delay of our scheme is the smallest, by a big margin. Our GE equivalent is also smaller than the others. The area-delay product is a measure of complexity [23]. Table 2 shows that our scheme has the lowest complexity, leaving enough space for other security functions. The die area, power consumption and throughput values have been calculated with the same metrics used in Akari-x, in order to base the comparisons on equal ground. Having the smallest GE, our proposal has a smaller die area and consumes less power. Also only our scheme has simple

control as it is sequential, requiring no iteration loops. Because it is based on MT and shown to be ultra-light, we would like to call our scheme MeTuLR: MT based ultra-light RNG for RFID tags.

The area-delay product for hashing functions is very high, even for 32-bit architectures [19, 23]. Their complexity values indicate clearly that encryption and hashing schemes are not in the ultra-light category. Therefore, the work of [11] has not been included in the comparisons.

4.2 Testing Results

Two sets of inputs were used to reach the final scheme, in the randomness tests. At first, a set from http://random.org was used to expose the failing schemes. Then a set with low entropy (0.00) was used, similar to that of the set in [11]. Many schemes that performed well with RN inputs faltered with low entropy inputs. Only those schemes that passed the randomness tests with low entropy inputs were improved, reaching MeTuLR of Fig. 1. It should not go unnoticed that LAMED and Akari-X use random.org inputs for obtaining a RN from a RN, but RN seeds are not available in tags.

In testing PRNGs, the ENT [24], Diehard (versions 1, 2) [25] and NIST [26] randomness tests are used for comparison with the previous work. Diehard 1 has 15 different tests, each with a "p-value" output. The p-values are in the range 0.0–1.0, where a result close to either extreme is considered unsatisfactory. Diehard 2 gives an overall p-value, where a result less than 0.1 is a fail. The NIST test outputs p-values expected to be in the range 0.1–0.9 and "proportion" values that should be above 0.95; any undesirable result is marked by a "*".

According to the ENT results; MeTuLR, Akari-x and Lamed all pass satisfactorily. ENT is relatively a more relaxed test suite than the Diehard and NIST tests. Since the results are very close, a comparison table omitted here is posted on http://srg.cs.deu.edu.tr/publications/2012/prng/, for reference.

To expose the difference in the schemes, we move on to the more strict tests. Because of the detailed output of the Diehard and NIST tests, all of the results and inputs are posted at http://srg.cs.deu.edu.tr/publications/2012/prng/. For Diehard 1 test, we used the evaluation criteria used in [27] where a score is given for each test result and the sum of the scores is calculated. We calculated the scores of the previous work by examining their declared results and summarized the overall sums in Table 3. For 15 tests, the maximum score is 15. It is worth to reiterate that our preliminary schemes which scored high with RN inputs, scored very low with low entropy inputs. After MeTuLR scored 10.94 compared to 11.84 of original MT, the previous works were also calculated; even though their inputs are RNs and not low entropy values.

The 32-bit version of Lamed performs best, in Diehard 1. Akari1 versions are also satisfactory, but Akari2 versions perform well below MeTuLR. MeTuLR's result is far from being unsatisfactory, when considered with its very low complexity value.

Table 3 Comparison of diehard and NIST tests

	Diehard1 score	Diehard2 score	NIST
Akari1A	12.4	0.353	Pass
Akari1B	12.4	0.353	Pass
Akari2A	7.5	0.082	Pass
Akari2B	7.5	0.082	Pass
Akari2C	7.5	0.082	Pass
Lamed	13.0	0.778	Pass
MeTulR	10.94	0.224	Pass*

Meanwhile, Diehard version 2 scores of all schemes are in the satisfactory range except Akari2, which are below the accepted limit of 0.1.

The NIST test results of previous works are also posted on their referenced pages. The final results are compiled in Table 3. The previous works declare to have passed the tests. Our test results however, show that our scheme fails the Rank and Universal tests, pointed by the "*" marks. All of the other NIST test results are good and MeTuLR can be considered to pass the NIST test. It is acceptable for a scheme to fail a few tests out of 188 tests; i.e. a scheme failing two individual tests cannot be considered to fail the overall, strict NIST test [28].

5 Conclusion and Future Work

This paper outlines a new random number generator feasible in low-cost RFID tags. The obtained low complexity, power consumption and die area results indicate that the proposed scheme does not violate the resource limits of ultra-light tags. Our scheme takes non random numbers as seeds and produces random numbers. Previous works use random numbers which are not available in RFID tags. The randomness test results of our proposed scheme are satisfactory, considering the non random inputs used for seeding. Future work involves the design and implementation of the proposed scheme. Also additional work is needed until the scheme passes all NIST tests. Efforts of hashing and encryption designs for random numbers in tags are intensified, but our scheme is available now.

References

1. Robert, C.M.: Radio frequency identification. Comput. Secur. **25**, 18–26 (2006)
2. Das, R., Havrop, P.: RFID forecasts, players and opportunities 2011–2021. IDTechEX (2010)
3. Lopez, P.P., Castro, J.C.H., Tapiador, J.E., Ribagorda, J.: LAMED a PRNG for EPC Class-1 Generation-2 RFID Spec. Comput. Stand. Interfaces **31**(1), 88–97 (2009)
4. Sarma, S.E., Weis, S.A., Engels, D.W.: RFID systems and security and privacy implications. In: Proceedings of the 4th International Workshop on CHES. LNCS, vol. 2523, pp. 454–470 (2002)

5. Chien, H.Y.: SASI: a new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. In: Transactions on Dependable and Secure Computing, vol. 4, pp. 337–340. IEEE (2007)
6. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=46149
7. Class-1 Generation 2 UHF Air Interface Protocol Standard "Gen-2", Version 1.2.0, http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2
8. Martin, H., Millan, E.S., Entrena, L., Lopez, P.P., Castro, J.C.H.: AKARI-x: a pseudorandom number generator for secure lightweight systems. In: 17th IEEE IOLTS, pp. 228–233 (2011)
9. Jun, B., Kocher, P.: The Intel ®Random number generator. Cryptography Research, White Paper (1999)
10. Menenez, A.J., Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography, Chapter 5, pp. 169–190. CRC Press, Boca Raton (1996)
11. Holcomb, D.E., Burleson, W.P., Fu, K.: Power-up SRAM state as an identifying fingerprint and source of true random numbers. IEEE Trans. Comput. **58**(9), 1198–1210 (2009)
12. Che, W., Deng, H., Tan, X., Wang, J.: A random number generator for application in RFID tags. In: Cole, P.H., Ranasinghe, D.C. (eds.) Networked RFID Systems and Lightweight Cryptography, Chapter 16, pp. 279–287. Springer, Berlin (2008)
13. Segui, J.M., Alfaro, J.G., Joancomarti, J.H.: Analysis and improvement of a pseudorandom number generator for EPC Gen2 Tags. FC'10, LNCS, pp. 34–46. Springer-Verlag (2010)
14. Segui, J.M., Alfaro, J.G., Joancomarti, J.H.: A practical implementation attack on weak pseudorandom number generator designs for EPC Gen2 tags. Int. J. Wirel. Pers. Commun. **59**(1), 27–42 (2011)
15. Alomair, B., Lazos, L., Poovendran, R.: Passive attacks on a class of authentication protocols for RFID. In: ICISC'07, pp. 102–115 (2007)
16. Matsumoto, M., Nishimura, T.: Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Trans. Model. Comput. Simul. **8**(1), 3–30 (1998)
17. Matsumoto, M., Nishimura, T., Hagita, M., Saito, M.: Cryptographic Mersenne Twister and Fubuki Stream/Block Cipher (2005)
18. Panneton, F., L'Ecuyer, P., Matsumoto, M.: Improved long-period generators based on linear recurrences modulo 2. ACM Trans. Math. Softw. **32**(1), 1–16 (2006)
19. Feldhofer, M., Dominikus, S., Wolkerstorfer, J.: Strong authentication for RFID systems using the AES algorithm. In: LNCS, vol. 3156, pp. 357–370. Springer (2004)
20. Moradi, A., Poschmann, A.: Lightweight cryptography and DPA countermeasures: a survey. In: LNCS, vol. 6054, pp. 68–79. Springer (2010)
21. Paar, C., Poschmann, A., Robshaw, M.J.B.: New designs in lightweight symmetric encryption. In: RFID Security: Techniques, Protocols and System-on-Chip Design, vol. 3, pp. 349–371. Springer, Berlin (2009)
22. Lopez, P.P., Lim, P.T., Li, T.: Providing stronger authentication at a low-cost to RFID tags operating under the EPCglobal framework. In: IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, pp. 159–167 (2008)
23. Feldhofer, M., Wolkerstorfer, J.: Hardware implementation of symmetric algorithms for RFID security. In: RFID Security: Techniques, Protocols and System-on-Chip Design, vol. 3, pp. 373–415. Springer (2009)
24. Walker, J.: Randomness battery. http://www.fourmilab.ch/random/ (1998)
25. Marsaglia, G.: The Marsaglia Random Number DIEHARD Battery of Tests of Randomness. ver1:http://stat.fsu.edu/pub/diehard, 1996, ver2: http://i.cs.hku.hk/~diehard/ (2003)
26. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., Vo, S.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. http://csrc.nist.gov/rng/ (2001)
27. Alani, M.M.: Testing randomness in ciphertext of block-ciphers using dieHard tests. Int. J. Comput. Sci. Netw. Secur. **10**(4), 53–57 (2010)
28. Kohlbrenner, P., Gaj, K.: An embedded true random number generator for fpgas. In: Proceedings of the 12th FPGA 2004, ACM/SIGDA, pp. 71–78 (2004)

Minimization of the Receiver Cost in an All-Optical Ring with a Limited Number of Wavelengths

David Poulain, Joanna Tomasik, Marc-Antoine Weisser and Dominique Barth

Abstract A new all-optical node architecture, known as POADM, may lead to a considerable cost reduction for the infrastructure of the rings ensuring at the same time their excellent performance. We present a dimensioning problem which consists of minimizing the total number of receivers in nodes for a ring with a fixed number of wavelengths and a given traffic matrix. We prove the problem NP-completeness and provide a heuristic whose principle is to match and to group transmissions instead of considering them independently. We justify the group matching approach by confronting the results of our algorithm with its version without matching. The results obtained allow us to recommend the heuristic in the planning of POADM rings.

1 Introduction

All-optical networks offer both better performance and lower energy consumption than the classical opto-electronic networks [2]. For these reasons, they have been chosen to be the next generation of metropolitan networks. The DOROthéE project is a Digiteo project financed by the Ile-de-France region. It aims to reduce the *CAPital EXpenditure* (CAPEX) of these networks by properly dimensioning them. There are

D. Poulain (✉) · J. Tomasik · M.-A. Weisser
Computer Science Department, SUPELEC (E3S),
91192 Gif-sur-Yvette, France
e-mail: david.poulain@supelec.fr

J. Tomasik
e-mail: joanna.tomasik@supelec.fr

M.-A. Weisser
e-mail: marc-antoine.weisser@supelec.fr

D. Barth
UVSQ, PRiSM, 78000 Versailles, France
e-mail: dominique.barth@prism.uvsq.fr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_25,

two main parameters that have to be considered in the dimensioning process, the number of wavelengths and the equipment required in the nodes. Thus, this is a bi-criteria problem.

In all-optical networks, the ADD/DROP ability is provided by *Optical Add-Drop Multiplexers* (OADM). The electronic conversion is no longer necessary as an OADM allows a wavelength to bypass a node. A very promising OADM based architecture, the *Packet Optical Add-Drop Multiplexer* (POADM) is proposed in [4]. A POADM node is equipped with fixed receivers and a single fully-tunable transmitter providing both flexibility and low CAPEX. We give a proper dimensioning solution for POADM rings.

In [12] we considered the number of receivers as a constraint. There, we defined the single criterion *Minimum WaveLength Problem* (MWLP). The problem was to find the minimum number of wavelengths (λs) needed for a given traffic matrix when the total number of receivers in the network is minimal. We showed that the problem is NP-complete and provided a heuristic. In the present work, we define the problem called the *Minimum Receiver Problem* (MRP) consisting in finding the minimum number of receivers needed for a given traffic matrix in a network where the total number of available λs is fixed. As long as we do not use more than this number of λs, the cost of using one more λ is negligible since the fiber is already present. In that case, the number of receivers becomes the only parameter, whereas the maximum number of λs becomes an additional constraint. There is a relation between these two problems. Actually, if a solution to the MWLP also respects the λ constraint then this solution is an optimal solution to the MRP. In the article we thus attempt to treat the MRP.

The rest of this paper is organized as follows. Section 2 contains an overview of all-optical technologies and details of the POADM architecture. In Sect. 3 we introduce the MRP, study its complexity and present a heuristic algorithm which solves the MRP and comment on the results in Sect. 4. Finally, we conclude and outline perspectives.

2 OADM Based Architectures

Optical networks use WDM to carry a vast amount of traffic through the fiber. Each λ is considered as a high speed channel with a fixed transmission rate. The assignment of a lightpath consists in finding the route and the λs on which the lightpath is to be set. We focus our attention on rings without λ conversion capability. Under this condition, the problem of the lightpath assignment is to find a single λ for each lightpath. Since the λ capacity is large, most of the dimensioning solutions consist in grouping lightpaths to reduce the number of λs required. These methods, known as *traffic grooming* methods [6, 10], use TDM to divide λs into time slots. The source and destination nodes of a lightpath require ADD and DROP capability, respectively, on the λ onto which the lightpath will be assigned. In the OADM architecture, ADD and DROP functionalities are separated and handled by transmitters (Tx) and receivers

(Rx), respectively. The ADD/DROP capacities determine the performance of the networks but the cost of the Tx and Rx cards represents also a large part of the CAPEX.

A Tx can be either coloured or uncoloured depending on the fact that it is associated with a single λ or not. The ADD part of an OADM is composed of Txs/lasers. A coloured Tx is linked to a fixed laser tuned onto a single λ. An uncoloured Tx is linked with a tunable laser that provides the node with the capacity of adding onto any λ (but only onto a single λ in a time-slot).

Receivers can be either coloured or uncoloured. There are two types of architecture for the DROP part: *Switch-Based* (SB) and *Broadcast and Select* (B&S) [14]. The first one uses a demultiplexer on the optical signal and switches the λ to a coloured or uncoloured Rx. The second one uses a splitter to tap off a portion of the optical signal power from the ring to make all λs available to a node. Then the desired λ is selected by a filter and dropped by an uncoloured Rx. The SB approach provides a high λ capacity as it allows *vertical access* concurrency (possibility to access several λs in the same time slot). Simultaneously, it becomes costly if the number of Rxs involved is not adapted to the capacity demand of a node. In contrast, the B&S is cheaper but could reveal itself inefficient if the capacity demand of the node is high.

We list projects which studied the OADMs. In DAVID [5], the authors proposed two architectures, the first "short-term" one is made up of commercially available technologies. It uses fixed Txs and a B&S architecture. The second "long-term" architecture used a tunable laser and SB approach for the DROP part. The number of λs visible for each node is limited to four. In RingO [3], a proposed ring must have an equal number of nodes and λs. Each node has only one coloured Rx and the ADD part is made of fixed lasers, one laser per λ. In HOPSMAN [14], the authors adopt both B&S approach for the DROP, and a fully fast tunable laser for the ADD. The bandwidth available for each node is limited due to the vertical access constraint.

POADM architecture We study the ECOFRAME ring [7]. Each ring node is equipped with a POADM which is a fully tunable Tx and a SB DROP device with one or more coloured Rxs. Optical gates are used to remove packets at a destination in order to control the QoS. They also allow a POADM node to control the variation of the optical signal power providing the possibility of node cascading. This physical issue has always, in the past, limited the use of SB architectures. The POADM solution is cheaper than every other OADM architecture with comparable performance as it has a smaller number of line cards. Since it uses tunable lasers and a SB approach, the architecture is both flexible and efficient.

3 Minimum Receiver Problem and its Heuristic

Problem 1. Minimum Receiver Problem (MRP) A circuit $G = (V, E)$, a traffic matrix T where $T[i, j]$ is an amount of traffic to be sent from a node i to a node j, a set $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_W\}$ of wavelengths, $W \in \mathbb{N}$, a wavelength capacity $C \in \mathbb{N}$, a number of receivers $z \in \mathbb{N}$.

An *assignment* is a decomposition of T into a set of W matrices T_k. Each T_k is associated with a $\lambda_k \in \lambda$. Is it possible to find an assignment of the traffic T on the wavelengths of λ that respects simultaneously the *flow* and *capacity* constraints and that use at most z receivers?

Flow constraint: For any couple of nodes (i, j), the sum traffic carried on all λs has to be equal to the total amount of traffic between i and j.

Capacity constraint: Let $load_k(x)$ be equal to the amount of traffic carried by the arc x on the λ_k. $load_k(x) \leq C$.

We use the *Partition Problem* (PP) [8] to prove the MRP NP-Completeness.

Theorem 1 *The MRP is NP-complete.*

Proof The certificate of the MRP is obviously in P. From an instance of PP we build an instance of the MRP. For each integer x_i of X we create two vertices in an initially empty circuit G, s_i (source) and d_i (destination). Nodes in G are ordered so that $s_1 \prec \cdots \prec s_m \prec d_1 \prec \cdots \prec d_m$ where $a \prec b$ means that a precedes b. We build the matrix T with the only non-zero elements $T[s_i, d_i] = x_i$. We fix $W = 2$, $C = (\sum_{k=1}^{m} x_k)/2$ and $z = m$.

We associate the subset A (respectively B) with the first λ_1 (λ_2) wavelength. In the PP solution, x_i is associated with A or B. The whole traffic $x_i = T[s_i, d_i]$ is thus assigned to the λ which corresponds to this subset. As we assign the entire traffic flow to a single λ, the flow constraint is respected and the solution uses at most z receivers. For each λ the amount of traffic passing through an arc is no more than the amount of traffic passing through the arc (s_m, d_1) on the same λ. Since $\text{load}_1(s_m, d_1) = \text{load}_2(s_m, d_1) = C$, the capacity constraint is respected. If the PP has a solution then the MRP has a solution too. Inversely, if we have a solution to the MRP, then each traffic flow is assigned to a single λ due to the number of receivers limited to z. The assignment of traffic on the arc (s_m, d_1) provides the partition of integers.

Heuristic: For a given n-node ring with nodes numbered from 1 to n, we consider each λ as an n-dimension cube. Dimension i is associated with the arc i (the arc between nodes i and $i + 1$). The length of each edge of this cube is equal to C. Such a cube is called a *box* and the number of boxes is equal to the number of λs available, w. A request $r^{(x,y)}$ is a traffic flow between nodes x and y. It is represented as an n-dimension vector. The height $h(r)$ of request r is equal to the amount of traffic this request is carrying and its length $l(r)$ is equal to the number of arcs between its origin and its destination. For example, in a 4-node ring we consider the request $r^{(1,3)} = (3, 3, 0, 0)$. This request passes only through arcs 1 and 2. Thus, we have $h(r^{(1,3)}) = 3$ and $l(r^{(1,3)}) = 2$. A *unitary* request has a height equal to one. As non-unitary requests can be split over several λs, we may consider that all requests are unitary requests.

A set of transmission requests with the same destination forms an *element*. Inside the element, requests are decreasingly ordered by length. An element e may be seen as a vector sum of the request vectors it contains. An element composed of all the requests towards a given destination is called a *complete* element. The length of an

element e is $l(e) = \max_{r \in e}(l(r))$ and its size is $s(e) = \sum_{r \in e} l(r)$. Its height $h(e)$ is equal to the number of unitary requests it contains. A *rectangular element* contains only requests of the same length.

To minimize the number of the Rxs, the traffic associated with a given complete element should be carried by the smallest number of λs. Our goal is therefore to cut the complete elements into elements that fit into the smallest number of λs. To discover the shape of a complete element e_d, we have to compute the amount of traffic towards d. Let us note l_i^d the amount of traffic in the arc i destined to d. Under the assumption that a node sends nothing to itself and taking into account a circular architecture we obtain: $l_i^d = 0$ if $i = d, l_i^d = l_n^d + t_{i,d}$ if $i = 1$ and $d \neq 1$, finally $l_i^d = l_{i-1}^d + t_{i,d}$ otherwise.

The three steps, that compose our algorithm, are repeated until the assignment of all traffic. The variable C_h represents the height of the cut.

Initialisation $C_h = C$.

Step 1 Generally, heuristics of packing obtain better results when the elements to be packed have regular shapes. A *cut* provides a partition of an element e into a set of k *resultant* elements $\{e_0, e_1, \ldots, e_{k-1}\}$ with $k = \lceil h(e)/C_h \rceil$. We want a cut to have some special properties to produce resultant elements with regular shape. Firstly, resultant elements should (as much as possible) be the same height. Ideally, this height is a sub-multiple of C. Secondly, resultant elements should be as low as possible to reduce the space they will take when packed into a box. We decide to measure how much an element differs from the closest rectangular element. The measure of the *irregularity* of an element e is its irregularity number $\text{irr}(e) = hl(e) - s(e)$. The cut has to minimize $\sum_i \text{irr}(e_i)$.

Informally, we compose groups of C_h requests from the bottom to the top of e. Since the requests are ordered in e, the resultant element e_i contains longer requests than the resultant element e_{i+1}. The element at the top can be smaller than h. Formally,

$$e_i = \begin{cases} \{r_{iC_h}, r_{iC_h+1}, \ldots, r_{iC_h+C_h-1}\}, & \text{if } i \neq k-1 \\ \{r_{(k-1)C_h}, \ldots r_{h(e)}\}, & \text{otherwise} \end{cases}$$

Step 2 only if $C_h > 1$, We use here an acceptance–rejection method to select groups of elements. The acceptance rate is noted τ. Ideally, we would like to consider each possible set of elements. Nevertheless, as we want the complexity to remain reasonable, we consider hereafter only pairs of elements (or single elements). If the elements of a pair do not fit together in an empty box of capacity C_h then the pair is rejected regardless of τ. For elements a and b, from a non-rejected pair, we compute $\text{fit\_rate}(a, b) = \frac{s(a)+s(b)}{nC_h}$ which measures the fraction of space occupied a and b when packed in a virtual box of capacity C_h. A pair with a *fit rate* greater than τ is selected. As an element can appear in more than one pair, we use a maximum matching algorithm [9] (on the selected pairs) to get the biggest subset of accepted pairs that does not contain a same element twice. We notice that the elements of a same accepted pair will be treated as a single element from this moment. In the remaining subset of elements, an element a is accepted if $\text{fit\_rate}(a) = \frac{s(a)}{nC_h} > \tau$.

Step 3 We use a FFD method [13] to pack all the accepted elements (or pairs) into the w boxes. The height of the cut is modified so that $h = \lfloor h/2 \rfloor$.

4 Results

We discuss the performance of the heuristic algorithm. Firstly, we show, on instances, the influence of τ. Afterwards, we compare two variants of our heuristic: with and without pairing the elements in step 2. We show, thereby, the influence of the pairing of elements and explain in which cases it should be used. The experiments have been done for 16-node ring with λ capacity $C = 32$. We use *All-To-All* (ATA) spatial traffic distribution, in which the sizes of the connections are generated following uniform or normal (N(μ, 0.2μ), $\mu = 16$) distribution. In another series we use *Rich-Get-Richer* (RGR) [1] spatial distribution for which the mean volume of traffic received by each node is equal to μ. An RGR distribution represents realistic traffic conditions, as in a metropolitan ring some nodes may attract more traffic.

Influence of the acceptance rate τ: As said before, the number of Rxs required for a given node is equal to the number of parts in which the associated complete element has to be cut. To minimize the number of Rxs we want this number of parts to be as small as possible (we want the resultant elements to be as high as possible). Nevertheless a too high resultant element may be difficult to pack if it does not fit well with others. The acceptance rate τ allows us to select elements that, despite their height, do not lead to the degradation of the λ utilization. It seems obvious that the numbers of Rxs and λs are factors that evolve in the opposite direction but it is not clear how τ will affect these numbers. We represent the solution quality by the value of z/W as it exhibits the compromise that must be made between the number of Rxs and λs. The table below depicts the evolution of the ratio z/W.

Acceptance rate	0.96	0.94	0.92	0.90	0.87	0.83	0.82	0.8	0.75	0.69
Number of Rxs/ Number of λs	357/85	328/66	318/67	276/68	246/69	195/70	195/71	173/72	154/73	125/74

We see that if τ increases then the ratio z/W increases too. If τ is high then the pairs of elements tend to be rejected and packed later. The number of Rxs thus increases whereas the number of λs decreases. So, we know that when the load of traffic is high in the ring, solutions can be found by increasing τ. Symmetrically, if the number of λs is large, we can save Rxs by decreasing τ.

Influence of the pairing method: Figures 1 and 2 depict the influence of using the pairing method or not in step 2 of our algorithm. Figure 1 has been computed on an instance with ATA spatial distribution and N(μ, 0.2μ) distribution for the size of the connection whereas Fig. 2 has been computed on an instance with RGR traffic distribution. The total amount of sent/received traffic is equal in both cases. For this reason the minimal number of Rxs is the same (123 Rxs on the following example).

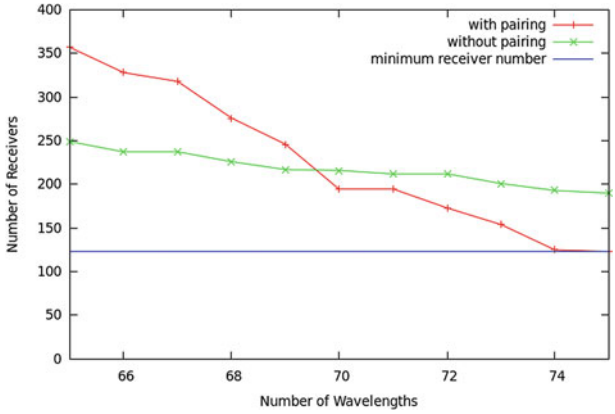


Fig. 1 ATA: with and without pairing

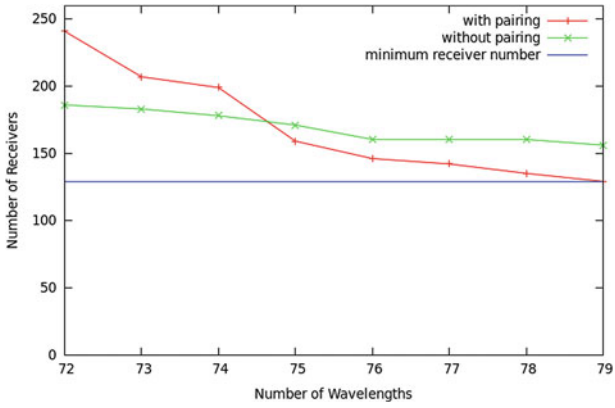


Fig. 2 RGR: with and without pairing

The straight line represents the minimal number of Rxs ($z_{\min}$) for a given traffic [12]. We notice that this lower bound, as it is not dependant on the number of λs, may not be close to the optimal solution when the number of λs is small. The two other curves represent the number of Rxs required for a given traffic and a given number of λs, with or without the pairing method. The solution without pairing is not highly affected by the number of λs. On the other hand, the solution with pairing performs extremely well when the λ constraint is not tight but leads to worse results when the assignment of traffic becomes the bigger problem.

As we want to assess on the performance of the heuristic with pairing, we have to study a large number of instances. Nevertheless, for a given traffic matrix T, it is difficult to say whether the λ constraint is either tight or open as the λ assignment problem is itself a difficult problem.

Below we explain our methodology to find the maximal $W_{\max}$ and minimal $W_{\min}$ numbers of required λs and our idea of classifying studied instances.

Our MWLP heuristic [12] provides a solution that has the minimum number of Rxs, $z_{\min}$ and a number of λs, $W_{\max}$. For each MRP instance that has more than/exactly $W_{\max}$ of λs we are able to find an optimal solution. Thus, our interest is in the MRP instances that have less than $W_{\max}$ of λs.

Let $W_{\min}$ be the number of λs used by a solution of an efficient λ assignment for the given matrix T. Such a solution can be provided, for instance, by the arc-colouring based heuristic [11]. It seems reasonable to consider $W_{\min}$ as the minimal number of λs required. In other words, we will consider the MRP instances which have more than $W_{\min}$ available λs.

We generated 500 random traffic matrices for 16-node rings using RGR. From a same matrix we generated three types of instances. In the first type, the number of λs is a *strong constraint*: $W \in \left(W_{\min}, W_{\min} + \frac{W_{\max}-W_{\min}}{3}\right)$. In the second type, the number of λ is *tight*: $W \in \left[W_{\min} + \frac{W_{\max}-W_{\min}}{3}, W_{\min} + \frac{2(W_{\max}-W_{\min})}{3}\right]$, and finally in the third one, the number of λs is *open*: $W \in \left(W_{\min} + \frac{2(W_{\max}-W_{\min})}{3}, W_{\max}\right)$. The results of this experiment are shown in:

$C = 32$	Open (%)	Tight (%)	Strong (%)
With pairing	5.60	20.00	29.70
Without pairing	14.00	23.50	27.50

$x\%$ means that the average solution has $x\%$ more Rxs than the optimal one for a number of available λs equal to $W_{\max}$. These results confirm the observations made on Figs. 1 and 2 when the λ constraint is open. In that case the heuristic with pairing increases the minimum number of Rxs by only 5.6 % whereas the heuristic without pairing gets 14 %. When the λ constraint is tight the pairing method also gets slightly better results than its opponent. Finally, the performance is comparable when the λ constraint is strong.

5 Conclusion and Further Work

The paper presents a part of our work on to the dimensioning of all-optical rings with the POADM nodes. Our goal was to minimize the number of receivers when the number of wavelengths is limited. We proved that the corresponding decision problem is NP-complete. The heuristic we proposed is based upon a preliminary matching of pairs of grouped transmissions which attempts to "wipe out" their shape irregularities. The simulation results show the advantage of pairing. The results converge to the optimal solution when the number of wavelengths is unlimited.

We consider studying together the problem we discussed here with the one previously treated in [12] to formulate and solve the bi-criteria problem.

References

1. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**, 509–512 (1999)
2. Bonetto, E., Chiaraviglio, L., Cuda, D., Castillo, G.G., Neri, F.: Optical technologies can improve the energy efficiency of networks. In: ECOC (2009)
3. Carena, A., et al.: RingO: an experimental WDM optical packet network for metro applications. IEEE JSAC **22**(8), 1561–1571 (2004)
4. Chiaroni, D., Neilson, D., Simonneau, C.: Antona, J.C.: Novel optical packet switching nodes for metro and backbone networks. In: ONDM (2010)
5. Dittmann, L., et al.: The European IST project DAVID: a viable approach toward optical packet switching. IEEE JSAC **21**(7), 1026–1040 (2003)
6. Dutta, R., Rouskas, G.: On optimal traffic grooming in WDM rings. IEEE JSAC **20**, 110–121 (2002)
7. ECOFRAME demonstration plateform. In: ECOC (2010). NTT and ALu Bell Labs.
8. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, New York (1979)
9. Gondran, M., Minoux, M.: Graphes et Algorithmes. Eyrolles, Paris (1995)
10. Hu, J., Leida, B.: Traffic grooming, routing, and wavelength assignment in optical WDM mesh networks. In: IEEE INFOCOM, pp. 495–501 (2004)
11. Kumar, V.: Approximating circular arc colouring and bandwidth allocation in all optical ring networks. LNCS **1444**, 147–158 (1998)
12. Poulain, D., Tomasik, J., Weisser, M.A., Barth, D.: Optimal receiver cost and wavelength number minimization in all-optical ring networks. In: ConTEL (2011)
13. Yao, A.: New algorithms for Bin Packing. J. ACM **27**(2), 207–227 (1980)
14. Yuang, M., et al.: HOPSMAN: an experimental testbed system for a 10-Gb/s optical packet-switched WDM metro ring network. Commun. Mag. **46**(7), 158–166 (2008)

Resilient Emergency Evacuation Using Opportunistic Communications

Gokce Gorbil and Erol Gelenbe

Abstract We describe an emergency evacuation support system (ESS) that employs short-range wireless communications among mobile devices carried by civilians. Emergency information is disseminated via opportunistic contacts between communication nodes (CNs), and each CN provides adaptive step-by-step navigation directions for its user during evacuation. Using mobile devices and opportunistic communications (oppcomms) allow ESS to operate when other means of communication are destroyed or overloaded. In this paper, we evaluate the resilience of oppcomms as used to enable evacuation support in ESS; we specifically consider the effect of CN failures on evacuation and communication performance. Our simulation experiments of evacuation of a three-floor office building show that ESS is highly resilient to node failures, and failure ratios up to 20 % are well-tolerated.

1 Introduction

Evacuation is an urgent and important component of emergency response that requires spatio-temporal decision making by the civilians affected in the emergency. The unknown impact of the event, incomplete and incorrect information on the situation, dynamic conditions of the emergency (such as a spreading hazard) and destroyed and inaccessible communication infrastructure introduce significant challenges for evacuation. We propose a resilient emergency support system (ESS) to provide evacuation support to civilians in the emergency area. ESS uses opportunistic communications [13] between pocket devices carried by people to disseminate

G. Gorbil (✉) and E. Gelenbe
EEE Department, ISN Group, Imperial College London,
SW7 2BT, London, UK
e-mail: g.gorbil@imperial.ac.uk

E. Gelenbe
e-mail: e.gelenbe@imperial.ac.uk

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_26,

information on the emergency. Using this shared local information, each device maintains a partially updated view of the environment and provides alerts and adaptive navigation directions to its user for evacuation purposes.

In this paper, we evaluate the resilience of opportunistic communications for evacuation support as employed in ESS. We specifically consider the effect of device failures on evacuation and communication performance. Our proposed emergency support system is targeted for densely populated urban areas and it can be deployed in both outdoor and indoor environments. In this paper, we describe ESS as deployed in a large multi-floor building.

1.1 Design Assumptions

The spatial configuration of the emergency area is important for evacuation. We represent the physical area as a graph $G(V, E)$: vertices V are locations where civilians can congregate, such as rooms, corridors and doorways, and edges E are physical paths that civilians can use to move inside the building. Multiple costs are associated with each edge $(i, j) \in E$:

- the edge length $l(i, j)$, which is the physical distance between the vertices;
- the hazard level $h(i, j)$, which represents the condition of the edge in relation to its danger level for evacuation; and
- the effective edge length $L(i, j) = l(i, j) \cdot h(i, j)$, which is a joint metric representing the total cost of an edge for evacuation, including hazard and physical distance.

We assume that the graph is known for a building. We also assume that there are **sensor nodes (SNs)** installed at fixed positions in the building, where each SN monitors its immediate environment for a hazard. A sensor can potentially monitor multiple edges in the building graph based on its sensing capabilities and location. In our simulations, we assume that each SN monitors a single edge. Each SN is battery powered and has a unique device ID, a location tag that represents the area (i.e. edge) monitored by the sensor, and short-range wireless communication capability. When requested, an SN sends its latest measurement for its edge (i.e. its $h(i, j)$ value).

1.2 Emergency Support System

The emergency support system (ESS) consists of **mobile communication nodes (CNs)** carried by civilians. Each CN is a simple pocket device with short-range wireless communication capability, a processor and local storage. CNs form an opportunistic network that exploits node mobility to communicate over multiple hops. Such opportunistic communications (oppcomms) are characterized by the "store-carry-forward" paradigm [13] where messages received by a CN are stored in local memory and carried with the CN as a result of human mobility. Messages stored

on behalf of others are then forwarded to other CNs as they come into contact. Thus, a message is delivered to its destination via successive opportunistic contacts. Because the opportunistic network (oppnet) can be disconnected for long periods of time, CNs may need to carry messages for long durations and delivery of messages is not guaranteed.

Oppcomms are used to disseminate hazard information among CNs in the form of **emergency messages (EMs)**. Hazard information is generated by sensor nodes (SNs) deployed in the building as described above. Each significant hazard measurement is stored in a new **measurement message (MM)** created by the SN monitoring the affected area (e.g. edge). An MM contains the source ID (SN ID), location information (edge ID or (i, j)), the hazard intensity $h(i, j)$, and measurement timestamp. The latest MM created by an SN is forwarded to any CN that comes in contact with the SN. When an MM is received by a CN, it is used to update the local view of the CN as discussed below. The MM is also translated into an EM that contains the source ID (CN ID) and information from the MM (intensity, edge (i, j), timestamp). Multiple MMs are combined into a single EM when possible. In contrast to MMs, which are sent from SNs to CNs via single-hop communications, EMs are sent from CNs to CNs over multiple hops using oppcomms. Each EM is destined for all CNs.

The first MM or EM received by a CN acts as an alarm, indicating that there is a hazard and the user of the CN should evacuate the building. Each CN stores the building graph in local storage and uses received MMs and EMs to update edge costs on its local graph. An update triggers the calculation of shortest paths from the current CN location to all building exits, and the path with the lowest cost is used as an evacuation path. Any shortest path (SP) algorithm can potentially be used; CNs employ Dijkstra's SP algorithm. Since effective edge lengths ($L(i, j)$ values) are used in SP calculation, the "shortest" path minimizes exposure to the hazard while also minimizing travel distance, with priority given to the safety of the civilian.

A CN uses the latest evacuation path it has calculated to provide step-by-step directions to its user. In order to do this, the CN needs to know its location in the building. Indoor localization is achieved using the fixed SNs: each SN contains a location tag; we use the edge ID (i, j) monitored by the SN in this implementation as the SN location tag. Once notified of the emergency, each CN periodically sends a **beacon** using local broadcast. SNs that receive this beacon reply with a **localization message (LM)** that contains the source ID, location tag and timestamp. Very accurate localization is not required since the location of CNs are approximated by the graph vertices. The short communication range of CNs and SNs also decreases localization error. The location of a CN is updated as it moves in the building via LMs, and at each location update the CN updates the directions given to its user based on its current location and evacuation path.

CNs use epidemic routing [15] for the dissemination of EMs, coupled with *timestamp-priority queues*, where EMs with the earliest creation timestamps are dropped from the queue when it is full. Although epidemic routing is an early oppnet routing protocol, our evaluations [12] have shown that it is very suitable for emergency support due to its flooding based approach. Epidemic routing is known to have high message delivery ratios and low message latencies at the cost of high

communication overhead [14]. However, communication overhead due to flooding does not seem to be applicable to ESS since each EM is targeted for all CNs, and good communication performance is desirable for emergency communications.

2 Resilience of Opportunistic Communications

Resilience of an emergency support system is an important property considering the critical nature of its application. Through the use of mobile devices and oppcomms, ESS operates independently of existing communication infrastructure. However, ESS is still susceptible to failures of its components. Our general intuition is that ESS would be quite resilient to failures due to the disruption tolerant nature of oppcomms. Our aim in this paper is to verify this view by evaluating the effect of node failures on evacuation and communication performance of ESS.

We have evaluated the resilience of ESS to CN failures with simulation experiments conducted with the distributed building evacuation simulator (DBES) [1]. We use a three-floor building model based on the EEE building at Imperial in our simulations. The ground floor is 24 m $\times$ 45 m and contains the two exits, the 2nd and 3rd floors are 24 m $\times$ 60 m. We simulate a spreading fire and associated effects such as smoke. The fire starts at the intersection of two corridors on the second floor near the staircases, and probabilistically spreads in the area along edges following a Bernoulli trial model and affects the health of civilians on adjacent vertices. Each civilian starts with a health of 100 and her health decreases as she is exposed to effects of the hazard. For each simulation, people initially start at random locations in the building following a uniform distribution on vertices. Civilians follow a probabilistic mobility model intended to simulate the movement of people during working hours when they are not evacuating. When a civilian is notified of the emergency, she follows directions provided by her CN to evacuate. Civilians move at 1.39 m/s within floors and 0.7 m/s at staircases. Simulations take physical congestion into account during civilian movement.

In these simulations, we assume that traditional means of communication have broken down, possibly due to the hazard. We assume that CNs cannot communicate when they are located on different floors; this may be due to physical factors that affect wireless signal strength, such as thickness of the inter-floor walls. We also assume there is no central alarm in the building (e.g. it has failed due to power failure). Therefore, ESS provides both alerting and navigation services to building occupants. All communication entities (CNs and SNs) are simulated as IEEE 802.15.4-2006 compliant devices. CN and SN data transfer rate is set to 100 kbits/s and 20 kbits/s, respectively. We do not explicitly simulate the PHY layer in our simulations, but we do take into account contention for the wireless medium as accessed through CSMA-CA (carrier sense multiple access with collision avoidance). CN communication range is assumed to be either 6 or 10 m; SN communication range is 5 m. These ranges have been chosen based on expected indoor communication range of 802.15.4 devices that transmit at 0 dB or less. In addition to the area graph and edge costs, each

CN can store 100 EMs. Messages used by ESS are very short, with most message types $\leq$16 bytes. EMs have an average length of 52 bytes; this means that average storage requirements for oppcomms is about 5 kB per CN.

We assume that some of the CNs have failed before the emergency starts, most probably due to battery depletion. We look at four different cases in our evaluation: 20 and 40 people per floor (pf) with CN ranges of 6 and 10 m. These cases allow us to evaluate the effect of CN failures in different population densities (medium and high) and with different CN ranges. Simulation results are an average of 50 simulation runs for each data point, and 95 % confidence intervals are provided. Each simulation run has different initial locations for people, mobility patterns, hazard spread pattern, and CNs randomly chosen as the failed nodes. In order to isolate the effect of evacuation strategy used by users of failed CNs, we present our results where data relating to such users have been removed. In practice, such users can follow a static evacuation strategy or follow people with functional CNs.

2.1 Simulation Results

We see that evacuation ratio (Fig. 1a) is affected less from failures when nodes have more frequent contact opportunities and when connected subnetwork sizes are larger, i.e. when population density and/or communication range is high. For example, with 40 pf and 10 m range, evacuation ratio is practically unaffected by failures. Effect of failures on evacuation ratio increases as population density and/or communication range decreases. With more failures in the system, evacuation ratio decreases in general. We see that ESS is fairly resilient to node failures in terms of evacuation ratio and that failure ratios of up to 20 % are well-tolerated. An important observation is that range has a greater effect on the resilience of ESS than population density. Average evacuee health (Fig. 1b) is generally quite high despite the failures. A general trend of decreasing health is observed as failures increase but the differences in average health are very small. We again observe that networks with better connectivity (higher density or range) are more resilient and less affected by failures.

Figure 1c, d present average and worst-case evacuation times[1] versus node failure ratio. Our results show that evacuation times increase as failure ratio increases, except for the 20 pf with 10 m range case, which shows decreasing average evacuation time until 10 % failure ratio. The effect of failures on evacuation time comes from two factors: (i) with more failures, people are alerted later of the fire and therefore start to evacuate later, and (ii) more people need to change paths during evacuation because of incomplete or outdated information, which both increase evacuation time.

Figure 2 presents how node failures affect communication performance in ESS; these metrics are calculated using EMs only. We see that message delivery ratio (Fig. 2a) decreases as failures increase due to fewer contact opportunities. These

[1] Average evacuation time is the mean of the evacuation times of all successfully evacuated civilians. Worst-case evacuation time is the evacuation time of the last person to leave the building.

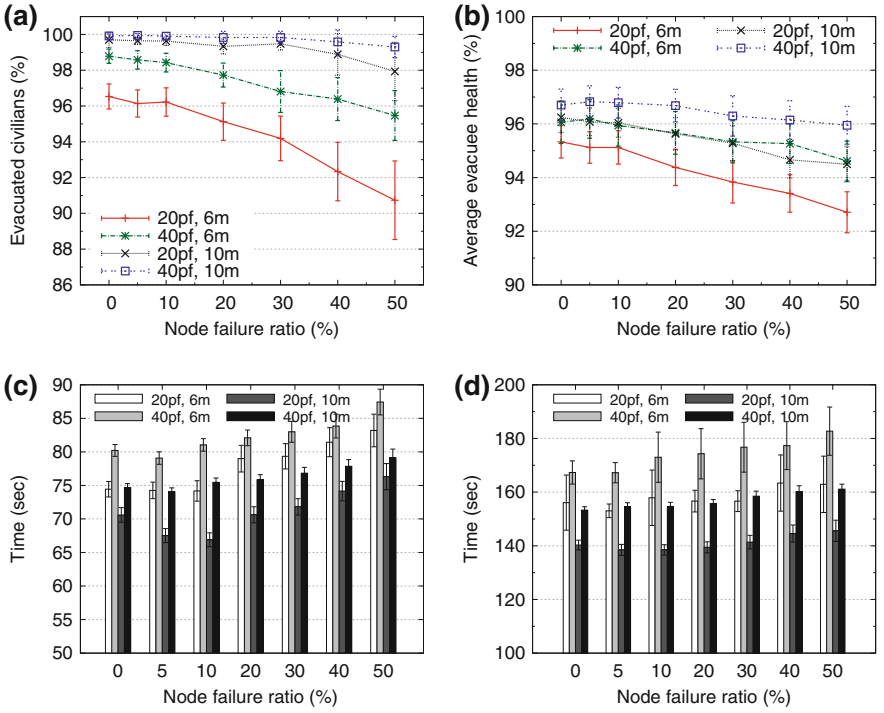


Fig. 1 Effect of node failures on evacuation performance. **a** Evacuation ratio. **b** Average evacuee health. **c** Average evacuation time. **d** Worst-case evacuation time

results show that oppnets are in general more resilient to node failures than wireless networks that require end-to-end connectivity for message delivery. We observe that communication range is more effective at maintaining high delivery ratio in the face of node failures than node density. Similar behavior is observed for average message delivery delay in Fig. 2b. We observe that average message delay increases with failure ratio, with the exception of the (20 pf, 6 m) scenario, which does not show any significant change, but the increase is less when communication range is high (i.e. 10 m). The increase in delay is more noticeable for the high density, medium range (40 pf, 6 m) scenario than others.

Both average message hop count (Fig. 2c) and average queue length[2] (Fig. 2) show similar trends with increasing node failures. For both metrics, we see that results are grouped based on population density and that range has less effect than density as opposed to our previous observations with other metrics. We observe considerable decrease in both hop count and queue length as failures increase. Hop count and message delay are loosely related in oppnets due to the "store-carry-

[2] Queue length is the number of EMs stored and carried by a CN for oppcomms. In ESS, CNs do not forget (drop) messages so the queue length increases monotonically until the queue is full. Average queue length is the mean of the maximum queue lengths of all CNs.

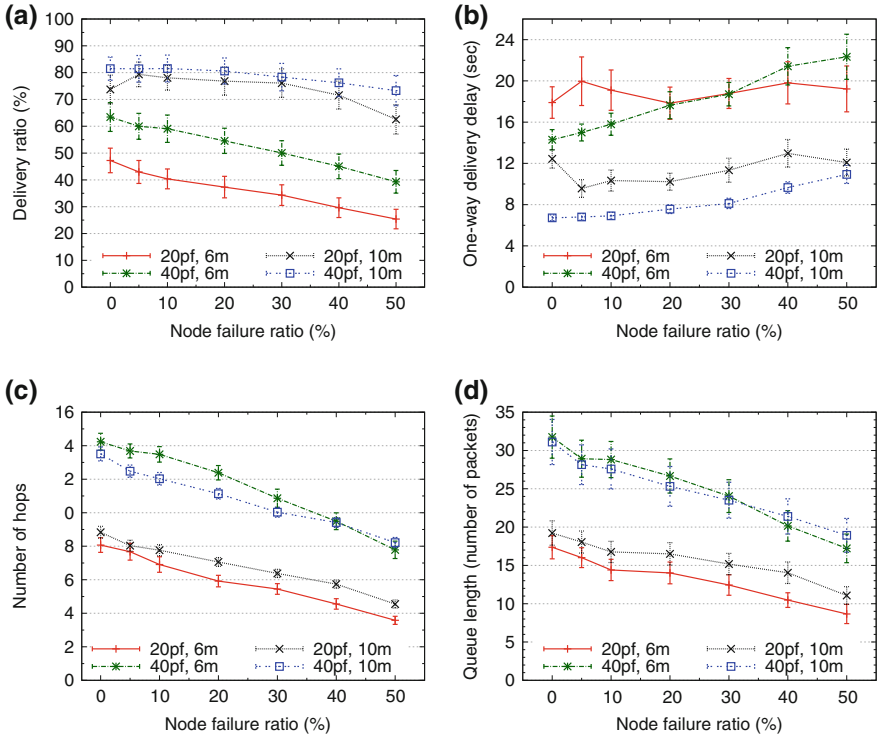


Fig. 2 Effect of node failures on communication performance. **a** Delivery ratio. **b** Average delay. **c** Average hop count. **d** Average maximum queue length

forward" message dissemination paradigm. A low hop count does not always mean a low delay since end-to-end delivery delay can be dominated by storage delay. This behavior is observed in our results: although increasing failures noticeably decreases hop count, delay increases. Hop count and queue length decrease as the number of failed nodes increases, mostly because there are fewer CNs to relay and receive messages and therefore messages reach fewer CNs.

3 Conclusions and Future Work

We have described an emergency evacuation support system (ESS) that employs opportunistic communications between pocket devices carried by people to disseminate emergency messages. Each communication node (CN) provides alerts and navigation directions to its user for evacuation based on its local view, which is updated via oppcomms. In indoor urban areas, fixed sensor nodes (SNs) are used to monitor the environment in real-time and for indoor localization of CNs. Due to the

use of oppcomms, ESS is a highly resilient system which can operate when other means of communication have broken down. ESS easily handles intermittent connectivity, link failures and node mobility. In this paper, we evaluated the resilience of ESS and oppcomms to node failures. Our simulation results have shown that ESS tolerates CN failures well, especially when connectivity is high. As CN communication range or the number of nodes in the area decreases, both network connectivity and resilience to failures decrease.

An ESS is a complex distributed system that provides spatio-temporal analyses and decisions to improve the outcomes for human beings that are affected by a small or large scale emergency [10]. Thus in this area we encounter all the classical and long-standing research questions related to distributed systems, including the incomplete and incorrect information on the distributed situation [4], and the inconsistent state information due to synchronisations and delays [5]. The distributed decision algorithms, for instance including the allocation of emergency teams and supporting mobile units, are also very challenging and require further work [7–9].

Our results in this paper focus only on communication issues and we have seen that the communication range has a greater effect on the resilience of ESS than node density. This suggests that the dynamic adjustment of communication range can be an effective way to improve resilience to node failures. We aim to investigate this approach in future work. In addition to node failures, we also need to consider deliberate attacks on the network. We believe that security of oppcomms is important due to the critical nature of emergencies and we will investigate the effect of network attacks in the context of emergency support in future work. Many of the problems that we have discussed would also benefit from a probability analysis as has been traditional in communication systems and other areas of information engineering [6]. Furthermore, techniques such as CPN [2, 3, 11] not only can benefit oppcomms, but have also proved useful in guiding evacuees. These aspects too will be included in future work.

References

1. Dimakis, N., Filippoupolitis, A., Gelenbe, E.: Distributed building evacuation simulator for smart emergency management. Comput. J. **53**(9), 1384–1400 (2010)
2. Gelenbe, E.: Cognitive packet network (CPN). US Patent 6,804,201 Oct 11 2004
3. Gelenbe, E.: Steps towards self-aware networks. Commun. ACM **52**(7), 66–75 (2009)
4. Gelenbe, E., Hébrail, G.: A probability model of uncertainty in databases. In: Proceedings of 2nd International Conference on Data Engineering (ICDE'86), pp. 328–333 (1986)
5. Gelenbe, E., Sevcik, K.C.: Analysis of update synchronisation algorithms for multiple copy data bases. IEEE Trans. Comput. **C-28**(10), 737–747 (1979)
6. Gelenbe, E., Stafylopatis, A.: Global behavior of homogeneous random neural systems. Applied Mathematical Modelling **15**(10), 534–541 (1991)
7. Gelenbe, E., Timotheou, S., Nicholson, D.: Fast distributed near-optimum assignment of assets to tasks. Comput. J. **53**(9), 1360–1369 (2010)
8. Gelenbe, E., Timotheou, S., Nicholson, D.: A random neural network approach to an assets to tasks assignment problem. SPIE Sig. Process. Sens. Fusion Target Recogn. XIX **7697**(1), 76,970Q (2010)

9. Gelenbe, E., Timotheou, S., Nicholson, D.: Random neural network for emergency management. In: Proceedings of Workshop on Grand Challenges in Modeling, Simulation and Analysis for Homeland, Security (MSAHS'10) (2010)
10. Gelenbe, E., Wu, F.J.: Large scale simulation for human evacuation and rescue. In: Computers and Mathematics with Applications (2012). doi:10.1016/j.camwa.2012.03.056
11. Gelenbe, E., Xu, Z., Seref, E.: Cognitive packet networks. In: Proceedings of the 11th International Conference on Tools with Artificial Intelligence (ICTAI'99), pp. 47–54 (1999)
12. Gorbil, G., Gelenbe, E.: Opportunistic communications for emergency support systems. Procedia Comput. Sci. **5**, 39–47 (2011)
13. Pelusi, L., Passarella, A., Conti, M.: Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. IEEE Commun. Mag. **44**(11), 134–141 (2006)
14. Song, L., Kotz, D.F.: Evaluating opportunistic routing protocols with large realistic contact traces. In: Proceedings of 2nd ACM Workshop on Challenged Networks, pp. 35–42 (2007)
15. Vahdat, A., Becker, D.: Epidemic routing for partially-connected ad hoc networks. Technical Report CS-2000-06, Duke University, Department of Computer Science (2000)

Part VII
Network Science I

Improving Hash Table Hit Ratio of an ILP-Based Concept Discovery System with Memoization Capabilities

Alev Mutlu and Pinar Senkul

Abstract Although ILP-based concept discovery systems have applications in a wide range of domains, they still suffer from scalability and efficiency issues. One of the reasons for the efficiency problem is high number of query executions necessary in the concept discovery process. Due to refinement operator of ILP-based systems, these queries repeat frequently. In this work we propose a method to improve hash table hit ratio for repeating queries of ILP-based concept discovery systems with memoization capabilities. The proposed method introduces modifications on search space evaluation and covering steps of such systems. Experimental results show that the proposed method improves the hash table hit count of ILP-based concept discovery systems with an affordable cost of extra memory consumption.

Keywords Concept discovery · ILP · Memiozation · Hash table hit.

1 Introduction

Inductive Logic Programming (ILP) [1] is a research area at the intersection of machine learning and logic programming and provides tools to infer general theories that describe a given set of facts. One of the most commonly addressed tasks in ILP is concept discovery [2], which aims to find rules that define a target relation in terms of other relations given as background data.

Although they have applications in a wide range of domains [3], concept discovery systems still sustain scalability and efficiency issues. In several studies [4–6] it has

A. Mutlu (✉) · P. Senkul
Department of Computer Engineering, Middle East Technical University, Ankara, Turkey
e-mail: mutlu@ceng.metu.edu.tr

P. Senkul
e-mail: senkul@ceng.metu.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_27,

been reported that evaluation of the search space constitutes majority of the total execution time of such systems. To improve running time of this step, techniques such as query optimization [5, 7], memoization [4, 6], and parallelization [8] have been proposed.

In ILP-based concept discovery systems with memoization capabilities, queries and their results are saved into look-up tables for later uses [4, 6, 9]. In this work we propose a method to improve hash table hit ratio of such systems by catching repeating queries that are generated not only within the same epoch but also at different epochs. It involves modifications on the search space evaluation process and the covering approach.

The proposed method is embodied in an ILP-based concept discovery system with memoization capabilities, which is called Tabular CRIS [6]. Compared to Tabular CRIS, experimental results show that the proposed approach has a higher hash table hit count for the experiments that find solution clauses in multiple epochs. Although the proposed approach requires more memory compared to Tabular CRIS, the memory requirement is significantly less than some other similar systems.

The rest of this paper is organized as follows. In Sect. 2 we briefly introduce ILP-based concept discovery and summarize two techniques that aim to improve running time of concept discovery systems by employing memoization. In Sect. 3 we provide detailed explanation of Tabular CRIS. In Sect. 4 we explain the proposed method, namely *Tabular CRIS with Extended Features* (*Tabular CRIS wEF*). In Sect. 5 we discuss the experimental results. Section 6 concludes the paper.

2 ILP-Based Concept Discovery and Efficiency Improvement

ILP-based concept discovery systems input a set of target instances that are either true or false, a set of background knowledge that is directly or indirectly related to the target instances and some quality metrics to evaluate the induced concept descriptors. Such systems output the concept descriptors in the form of Horn clauses where the negated literal is the target relation and the positive literals are from the background knowledge [10]. If recursion is supported, the target relation may appear as a positive literal in the induced Horn clause.

A generic concept discovery system starts with an initial hypothesis set and refines it until it meets the quality measures. Evaluation of a concept descriptor is basically about the number of positive and negative target instances it explains [11]. If a concept descriptor satisfies the quality measures, it is added into solution clauses set and the target instances it explains are removed from the target instance set, this process is called the covering algorithm. Then, the process restarts with the uncovered target instances and repeats until all target instances are removed from the target instance set or no more concept descriptors can be found.

Blockeel et al. [4] reported that WARMR [12] spends 87 % of its total execution time to evaluate concept descriptors with length of at most 1 for the Muta data set.

Similar results [5, 8] are reported for the PTE data set. These observations suggest that improving the total running time of a concept discovery system is highly dependent on improving search space evaluation process.

Several studies such as query optimization [5, 7], data sampling [13, 14], and parallelization [8] have been proposed to improve search space evaluation of concept discovery systems. Another direction for this aim is employing memoization. In memoization, computations are saved in look-up tables for later uses [15].

Query Packs [4] and Common Prefixes [9] are two approaches that incorporate memoization to improve running time of search space evaluation step of concept discovery systems. In Query Packs approach similar queries are grouped and represented as a tree-like structure. Common parts of these queries are executed once and their results are saved to complete the evaluation of the concept descriptors. Common Prefixes approach benefits from the properties of refinement operators of the concept discovery systems. Common Prefixes saves results of good concept descriptors to calculate values of their refinements.

3 CRIS and Tabular CRIS

In this section we present the ILP-based concept discovery system CRIS (Concept Rule Induction System) [16] and Tabular CRIS. CRIS is a hybrid concept discovery system which uses association rule mining techniques [12] to find frequent and strong concept descriptors.

CRIS loops through the following five steps until it finds concept descriptors that explain all target instances or no more concept descriptors can be found.

(1) **Generalization**: Most general two literal concept descriptors are generated. CRIS utilizes absorption operator of inverse resolution for generalization.
(2) **Specialization**: Concept descriptors of length l are specialized to form concept descriptors of length $l + 1$. CRIS utilizes Apriori-based specialization operator [17] to refine the concept descriptors.
(3) **Hypothesis space evaluation and pruning**: Concept descriptors are translated into SQL queries and these queries are run against the background data for support and confidence value calculations. Then, concept descriptors are pruned based on these values. Support value of a concept descriptor is about the number of target instances it explains, and confidence value is about the number of background instances that satisfies its body.
(4) **Filtering**: A search tree consisting of solution clauses that induce the uncovered target instances is constructed. Then, based on f-metric the system decides on which solution clause in the search tree represents a better concept description than others.
(5) **Covering**: Target instances explained by the solution clauses are removed from the target instance set.

Table 1 The *daughter* data set

Target instances	Background facts	
daughter(mary, james)	father(james, mary)	mother(helen, barbara)
daughter(patricia, robert)	father(david, linda)	female(maria)
daughter(linda, david)	father(paul, susan)	female(linda)
daughter(barbara, helen)	father(robert, patricia)	female(mary)
daughter(maria, sandra)	father(dennis, walter)	female(patricia)
daughter(susan, paul)	mother(sandra, maria)	female(barbara)
	mother(amanda, gary)	

Tabular CRIS employs memoization to improve *specialization* and *search space evaluation and pruning* steps of CRIS.

In CRIS, concept descriptors that form the search space should be unique. However, refinement of two distinct concept descriptors may result in the same refined concept descriptor. Tabular CRIS improves the specialization step by replacing the sequential search of CRIS with a hash based search to detect the repeating concept descriptors.

Although all concept descriptors of length l are different, their corresponding SQL statements may be identical. This is due the different renamings of the free body variables. In order to handle repeating queries problem Tabular CRIS maintains a hash table to store executed queries. Before a query is sent to database it is first searched in the hash table. If the search is a hit, the result is retrieved from the hash table, otherwise the query is executed and along with its result is inserted into the hash table. Tabular CRIS stores $< query, int >$ tuples, where query is the key and *int* is the mapped value indicating the result of the *SELECT COUNT* type of queries.

Tabular CRIS maintains the queries in two hash tables: one for support queries and the second one for confidence queries. The hash table that stores the support queries is cleaned each time some solution clauses are found, as support queries are about the number of target instances explained by the concept descriptors. The hash table for confidence queries is never cleaned as the background data is never altered.

4 The Proposed Approach: Tabular CRIS with Extended Features

The main shortage of Tabular CRIS arises from its incapability of handling regenerated support queries if some solution clauses are found meanwhile, i.e. at different epochs. Consider the *daughter* data set given in Table 1. Some of the most general two literal concept descriptors that CRIS will produce for the *daughter* data set in the generalization step are listed in Table 2.

Since all of these concept descriptors map to the same support query, Tabular CRIS will execute the support query for concept descriptor r_1, and will retrieve the

Table 2 Repeating queries

Concept rule	SQL query
r_1: daughter(A,B):-father(C,D)	SELECT COUNT(DISTINCT CONCAT(d.n1,'-',d.n2))
r_2: daughter(A,B):-mother(C,D)	FROM d WHERE d.covered = 0
r_3: daughter(A,B):-female(C)	

results of r_2 and r_3 from the hash table. Suppose that minimum support is set to 0.6, minimum confidence is set to 1.0, and maximum rule length is set to 2. With these setting, Tabular CRIS will output the following concept descriptor as a solution clause at the end of the first epoch:

daughter(X,Y):- father(Y,X), female(A)

This rule covers target instances: *daughter*(*mary, james*), *daughter*(*patricia, robert*), *daughter*(*linda, david*) and *daughter*(*susan, paul*).

In the second epoch, Tabular CRIS will reproduce r_1, r_2, and r_3 as the most general two literal concept descriptors. Tabular CRIS has to re-execute the support query for r_1 as the hash table has been cleaned after the solution clause is found. In order to avoid such situations Tabular CRIS is modified in the following ways:

(1) Query structure of support queries is changed from *SELECT COUNT* to regular *SELECT* statements. With this modification, queries now return a set of target instances that satisfy the concept descriptor, and hash table for support queries store $< query, resultset >$. Confidence queries still follow the *SELECT COUNT* structure.
(2) The covering algorithm is modified in such a way that it not only removes the explained examples from the target instance set but also removes them from the hash table for support queries. With this modification in the covering algorithm, hash table for support queries stores updated result sets after some solution clauses are found.

The modified version of the covering algorithm is given in Algorithm 1. The loop given in lines 4 through 6 is the modified part that removes the target instances from the hash table for support queries.

Similar to Common Prefixes and Query Packs approaches, Tabular CRIS wEF uses memoization techniques to improve search space evaluation. Although these systems aim to make use of the results of previously executed queries, their aims are quite different. Tabular CRIS wEF aims to match queries that are generated both at the same and at different epochs, while Common Prefixes and Query Packs approaches aim to match queries that are executed only within the same epoch. Another advantage of the proposed approach over the Query Packs and Common Prefixes approaches is that, it can retrieve support and confidence values of different concept descriptors if they map to the same SQL query. Indeed Query Packs approach also looks for common substructures in the concept descriptors but the proposed approach can match

Algorithm 1 Covering Algorithm

Require: E: target instances, BF: Background knowledge, H: Hypothesis
Ensure: E: target instances not covered, Memorized result sets updated
1: **for all** $e \in E$ **do**
2: **if** $BF \cup H \models e$ **then**
3: $E = E \setminus e$
4: **for** $i = 0$ to $i < SupData.size()$ **do**
5: $supData[i].resultSet = supData[i].resultSet \setminus e$
6: **end for**
7: **end if**
8: **end for**

Table 3 Properties of data sets

Data set	Num. pred.	Num. ins.	Min. sup.	Min. conf.
Dunur	9	224	0.3	0.7
Eastbound	12	196	0.1	0.1
Elti	9	224	0.3	0.7
Mesh	26	1749	0.1	0.7
Muta-Aggr	12	190	0.3	0.7
Muta	8	13541	0.1	0.7
PTE	27	23850	0.1	0.7
PTE-5	32	29267	0.1	0.7

two concept descriptors that are completely different; i.e. Query Packs approach will not be able to match (A,B):-$q(C,A)$,$r(D,E)$ to $p(A,B)$:-$r(C,D)$,$q(E,A)$ while the proposed approach will catch the common SQL query. When the proposed approach is compared with Query Packs and Common Prefixes approaches in terms of memory requirement, Tabular CRIS wEF will require less memory as the Common Prefixes and Query Packs approaches store a view, i.e. image of the database that corresponds to a particular query, while Tabular CRIS wEF will store only a subset of the target instance set.

5 Experimental Results

In this section, we present the experimental results of Tabular CRIS wEF in comparison to Tabular CRIS and Query Packs approaches. Table 3 lists the properties of the data sets used in the experiments and the experimental settings. Muta-Aggr is a subset of the Muta data set enhanced with aggregate predicates. Similarly, PTE-5 is modified version of PTE with aggregate predicates. For more information about data sets, the reader may referred to [8]. Maximum length of the concept descriptors is set to 3.

Table 4 Comparison of tabular CRIS wEF to tabular CRIS

Data set	Num. epochs	Speedup		Hit count		Memory consumption	
		T. CRIS wEF	T. CRIS	T. CRIS wEF	T. CRIS	T. CRIS wEF	T. CRIS
Dunur	1	13.13	15.38	2010	2010	46.5	28
Eastbound	1	435	457	9623	9623	2.1	2
Elti	1	9.85	16.50	943	943	61	41
Mesh	3	366	252	148695	141371	349	75
Muta-Aggr	2	16.90	16.38	1162	718	312	166
Muta	9	14.27	12.07	74586	43186	1273	60
PTE	1	7.24	5.90	13282	9347	2565	184
PTE 5	2	9.60	8.80	34194	25132	12427	1163

Table 4 compares achieved speedups, the number of hash table hits, and memory consumption of Tabular CRIS wEF to Tabular CRIS. The second column shows the number of the epochs Tabular CRIS wEF performed to find the solution set.

The reported speedup is over the running time of the original implementation of CRIS. As the experimental results show, for the data sets for which the solution clauses set is found in a single epoch there is a drop in the speedup. This is due to the extra computational cost of Tabular CRIS wEF introduced by searching and removing the covered target instances from the hash table. Although this takes very limited time, for the data sets where the solution clauses is found in seconds, it introduces drop in the speedup. For example, it takes around 1.52 s for Tabular CRIS to find the entire set of solution clauses for Eastbound data set and around 1.60 s for Tabular CRIS wEF. For the PTE-5 data set time required for the extra computations is around 3 s, which is almost ignorable compared to query execution time which is around 33 min.

In the fourth and fifth columns of the Table 4 we report the hash table hit counts. As the experimental results show, there is an increase in the number of hash table hits for the data set for which the entire solution clauses set is found in multiple epochs. For the other data sets, the hash table hit count achieved by Tabular CRIS is preserved by Tabular CRIS wEF as well.

The last two columns of Table 4 compare the memory consumption of Tabular CRIS and Tabular CRIS wEF. Compared to Tabular CRIS, Tabular CRIS wEF consumes more memory. This is due to the fact that, Tabular CRIS stores a single number for each query while Tabular CRIS wEF stores a list of tuples. Compared to other data sets, increase in memory requirement is fairly less for the Eastbound data set. This is due to the fact that these queries return result sets with at most 3 items.

As the experimental results show, Tabular CRIS wEF benefits from its added functionality when the entire solution set is found in multiple epochs. When the solution set is found in a single epoch, it behaves poorly in terms of memory requirement and speedup compared to Tabular CRIS.

When compared to state of the art memoization based concept discovery systems, Tabular CRIS wEF achieves greater speedups while consuming less memory for

memoization purposes. For example, for the Muta data set, Tabular CRIS wEF gains 14.25 speedup in cost of 1.2 MB, while Query Packs gains 2.51 speedup in cost of 1.5 MB.

6 Conclusion

In this work, we propose a method to improve hash table hit ratio of ILP-based concept discovery systems that aim to improve efficiency of such systems by employing memoization based techniques. We implemented the proposed approach on the ILP-based concept discovery system called Tabular CRIS. The resulting system is named Tabular CRIS wEF. The experimental results show that the proposed method considerably improves the time efficiency of Tabular CRIS when the solution clauses are found in multiple epochs under the cost of additional memory requirement. However, the maximum memory requirement is below the similar approaches in the literature and it is not a high amount for contemporary desktop machines.

References

1. Muggleton, S.: Inductive logic programming. In: Wilson, R.A., Keil, F.C. (eds.) The MIT Encyclopedia of the Cognitive Sciences (MITECS). MIT Press, Cambridge (1999)
2. Dzeroski, S.: Multi-relational data mining: an introduction. SIGKDD Explor. **5**(1), 1–16 (2003)
3. Bratko, I., King, R.D.: Applications of inductive logic programming. SIGART Bull. **5**(1), 43–49 (1994)
4. Blockeel, H., Dehaspe, L., Demoen, B., Janssens, G., Vandecasteele, H.: Improving the efficiency of inductive logic programming through the use of query packs. J. Artif. Intell. Res. **16**, 135–166 (2002)
5. Costa, V.S., Srinivasan, A., Camacho, R., Blockeel, H., Demoen, B., Janssens, G., Struyf, J., Vandecasteele, H., Laer, W.V.: Query transformations for improving the efficiency of ILP systems. J. Mach. Learn. Res. **4**, 465–491 (2003)
6. Mutlu, A., Berk, M.A., Senkul, P.: Improving the time efficiency of ilp-based multi-relational concept discovery with dynamic programming approach. In: ISCIS, pp. 43–50. (2010)
7. Struyf, J., Blockeel, H.: Query optimization in inductive logic programming by reordering literals. In: ILP, pp. 329–346. (2003)
8. Mutlu, A., Senkul, P., Kavurucu, Y.: Improving the scalability of ILP-based multi-relational concept discovery system through parallelization. Knowl. Based Syst. **24**, 352–368 (2012)
9. Rocha, R., Fonseca, N.A., Costa, V.S.: On applying tabling to inductive logic programming. In: ECML, pp. 707–714. (2005)
10. Quinlan, J.R.: Learning logical definitions from relations. Mach. Learn. **5**(3), 239–266 (1990)
11. Srinivasan, A.: The Aleph manual. http://www.comlab.ox.ac.uk/activities/machinelearning/Aleph/ (1999)
12. Dehaspe, L., De Raedt, L.: Mining association rules in multiple relations. In: ILP, pp. 125–132. (1997)
13. Sebag, M., Rouveirol, C.: Tractable induction and classification in first order logic via stochastic matching. In: IJCAI, pp. 888–893. (1997)
14. Srinivasan, A.: A study of two sampling methods for analyzing large datasets with ILP. Data Min. Knowl. Discov. **3**(1), 95–123 (1999)

15. Rocha, R., Silva, F., Costa, V.S.: YapTab: a tabling engine designed to support parallelism. In: TAPD, pp. 77–87. (2000)
16. Kavurucu, Y., Senkul, P., Toroslu, I.H.: Concept discovery on relational databases: new techniques for search space pruning and rule quality improvement. Knowl. Based Syst. 23(8), 743–756 (2010)
17. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 307–328. AAAI/MIT Press, Cambridge (1996)

Distributed Multivalued Consensus

Arta Babaee and Moez Draief

Abstract Motivated by the distributed binary consensus algorithm in [4] we propose a distributed algorithm for the *multivalued consensus* problem. In multivalued consensus problem, each node initially chooses from one of k available choices and the objective of all nodes is to find the choice which was initially chosen by the majority in a distributed fashion. Although the *voter model* (e.g. [1]) can be used to find a consensus on multiple choices, it only guarantees the consensus and not the consensus on the majority. We derive the time of convergence and an upper bound for the probability of error of our proposed algorithm which shows that, similar to [4], having an additional state would result in significant improvement of both convergence time and probability of error for complete graphs. We also show that our algorithm could be used in Erdos-Renyi and regular graphs by using simulations.

1 Introduction

With *distributed consensus* problem nodes use local interactions with their neighbors to reach an agreement on a choice from a set of k available choices. This would then be called a binary consensus if $k = 2$. The most famous distributed consensus algorithm is the voter model [1]. With the voter model at each time step each node contacts one of its neighbors and then changes its state to the state of that particular neighbor. Nodes are only required to be able to store and send one of the k states at any given time. However, the probability of error in the voter model is $\sum_i \frac{d_i}{2m}$, where m is the number of edges and d_is are the degrees of the nodes in minority.

A. Babaee (✉) · M. Draief
Intelligent Systems and Networks Group, Department of Electrical and Electronic Engineering, Imperial College London, London, UK
e-mail: ab3608@imperial.ac.uk

M. Draief
e-mail: mmd@imperial.ac.uk

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_28,

The work most closely related to this paper is the binary consensus algorithm in [4]. In [4], a binary consensus algorithm is introduced which uses three states for communication and memory. This is one state more than the voter model. It is then proved that by using this additional state the probability of error decreases exponentially with the number of nodes (N) and the convergence time becomes logarithmic in N for the case of complete graphs. In this paper, we use the same setup of [4] for consensus on k choices, which means adding only one state for both communication and memory. We prove that using this additional state, the convergence time becomes logarithmic in N for large N in the case of complete graphs. We also show that the upper bound on error probability decreases exponentially with a rate that depends on both N and the fraction of the two choices with highest number of votes. We then confirm our findings with simulations.

It is worth noting that while preparing the final version of this paper we found out about a similar algorithm for multivalued consensus in [2] as part of the proposed algorithm for addressing the *ranking* problem. The so called *plurality selection* algorithm uses $2k$ states of memory and signaling for the majority consensus problem in a complete graph. However, in our algorithm we use k+1 states for the communication and memory states as we are only addressing the majority consensus problem and not the ranking problem. Also, our results and proofs differ from those of [2].

2 Multivalued Consensus

We now introduce our algorithm which extends the binary consensus in [4] to a multivalued consensus on k choices.

2.1 Dynamics

Consider an asynchronous setup similar to [4] in which each node has a clock which ticks with Poisson rate 1 (equivalent of a global clock with Poisson rate N where N is the size of the network). At each time step, one of the nodes' (node z) clocks ticks. It then contacts a neighbor w with probability $p(z, w)$. If the graph is complete, and the probability distribution of interactions is uniform, the neighbor is chosen from the set of vertices (V) with probability $p(z, w) = \frac{1}{N}$. Node z will then change its state according to the message received from node w.

Let nodes initially choose one of the k available choices. With this type of consensus the goal for each node would then be to find the initial majority in a distributed fashion. With multivalued consensus nodes can have one of the states $1, \ldots, k$, and e at any time, where e is an additional state which shows that the node is undecided on the initial majority. In this case, if a node in state i ($i \in \{1, \ldots, k\}$) contacts a node in state i or e, it does not change its state and if it contacts a node in states $j, i \neq j$, it updates its value to e. Also, if a node in state e contacts a node in state i, it changes its state to i. In other words, nodes that have already decided about the initial majority would become undecided following a contact with a neighbor with a

different opinion. Also, any undecided node would simply accept the opinion of its neighbor.

Now let $X_i(z) = 1$ if a node z is in state i. Also, $X_l(z) = 0, \forall l, 1 \leq l \leq k$ if the node is in state e. Let $X_i = \sum_{z=1}^{N} X_i(z)$ and $p(z, w) = 1/N$ for all $z, w \in V$. This way X_i would be the number of nodes in state i. Also, define e_i as a vector of dimension k with all coordinates equal to 0 except the ith one which is 1. The rates for the Markov process would then be the following:

$$(X_1, \ldots, X_k) \rightarrow \begin{cases} (X_1 + 1, \ldots, X_k) : (N - X_1 - \cdots - X_k)X_1/N \\ (X_1 - 1, \ldots, X_k) : X_1(X_2 + \cdots + X_k)/N \\ \vdots \qquad\qquad\qquad\qquad \vdots \\ (X_1, \ldots, X_k + 1) : (N - X_1 - \cdots - X_k)X_k/N \\ (X_1, \ldots, X_k - 1) : X_k(X_1 + \cdots + X_{k-1})/N \end{cases}$$

Note that $\sum_{i=1}^{k} X_i \leq N$ and the Markov process terminates at one of the k states $(N, 0, \ldots, 0), \ldots, (0, \ldots, 0, N)$.

2.2 Probability of Error

To find the probability of error let X_i^t and U^t denote the number of nodes in state i and e at time t respectively. Also let the initial number of nodes in states $1, 2, \ldots, k$ be such that $X_1^0 < X_2^0 < \cdots < X_k^0$. This means that the number of initial votes for different choices are not the same and therefore a majority exists (the initial majority is choice k in this setting). Consider the following definition,

$$f(X_1, \ldots, X_k) = \mathbb{P}(X_1^t, \ldots, X_k^t) = (N, 0, \ldots, 0) \text{ for some } t \geq 0 \qquad (1)$$

$f(X_1, \ldots, X_k)$ is then the probability of all the nodes ending in state 1.

Note that if $k = 2$ the consensus would turn into a binary consensus and the probability of error would be equal to $f(X_1, X_2)$ (as the initial majority is choice 2). As we build our analysis based on the binary consensus we would denote this probability by a different notation $g(X_1, X_2)$ to avoid confusion. Therefore, the probability of error for the binary case would be:

$$g(X_1, X_2) = \mathbb{P}(X_1^t, X_2^t) = (N, 0) \text{ for some } t \geq 0 \qquad (2)$$

In [4] the probability of error for binary consensus has been derived.

For the sake of simplicity consider the following notations: $f(X) = f(X_1, \ldots, X_k)$ and $f_{X_l}(X) = f(X_l, X_1, \ldots, X_{l-1}, X_{l+1}, \ldots, X_k)$ where $1 \leq l \leq k$. Consequently $f_{X_l}(X)$ denotes the probability of all nodes ending in state l. Also, define $f(X + m \times e_i) = f(X_1, \ldots, X_{i-1}, X_i + m, X_{i+1}, \ldots, X_k)$. Using the first step analysis we can derive the following equation,

$$\left(\varepsilon\sum_{i=1}^{k}X_i+2\sum_{i,j=1,i\neq j}^{k}X_iX_j\right)f(X)=\varepsilon\sum_{i=1}^{k}X_if(X+e_i)+\sum_{j=1}^{k}X_j\sum_{i=1,i\neq j}^{k}X_if(X-e_j) \tag{3}$$

where $\varepsilon = N - \sum_{i=1}^{k} X_i$. Accordingly, $f(X_1, 0, \ldots, 0) = 1$ and $f(0, X_2, \ldots, X_k) = 0$. Here, if $X_k(0) > \cdots > X_1(0)$ the error occurs when the system hits any of the final states except $(0, \ldots, N)$. Also, $f(X_1, \ldots, X_1) = \frac{1}{k}$ because of the symmetry of the protocol. The following lemma is then true.

Lemma 1 *The solution to (3) where there is at least one node in each of the k states is given by*

$$f(X) = \frac{1}{k}\sum_{i=1}^{k} f(X - e_i) \tag{4}$$

The boundary conditions for k choices are defined by the ones for $k-1$ choices starting with the boundary conditions for two choices (equivalent of $g(X_1, X_2)$ as defined in [4]) where $f(X_1, 0) = 1$ and $f(0, X_1) = 0$.

The proof could be found in the appendix. Finding the probability of error using (4) is not straightforward. Instead, we try to use the result for $g(X_1, X_2)$ in [4] to find an upper bound for $f(X)$. We use the following Lemma.

Lemma 2 *For two different consensus algorithms (binary and multivalued) applied on the same set of nodes, the following relationship exists between the probability of ending in one of the two states in binary consensus (2) and the probability of ending in one of the k states in multivalued consensus (1).*

$$f(X) \leq g(X_1, X_i), i \neq 1 \tag{5}$$

where $\sum_{i=1}^{k} X_i = N$.

The proof is in the appendix. Lemma 2 means that the probability of ending up in any of the absorbing states decreases when we have more choices. Note that with k choices when $(X_1^0 < \cdots < X_k^0)$, error occurs when the consensus finishes in any of the absorbing states other than $(0, \ldots, N)$. This means that the probability of error (P_e) can be defined as follows:

$$P_e = \sum_{i=1}^{k-1} f_{Xi}(X) \tag{6}$$

Using the Lemma 2, we then would have the following bound on P_e

$$P_e < \sum_{i=1}^{k-1} g(X_i, X_k) \tag{7}$$

This is due to the fact that $f_{X_1}(X) = f(X_1, \ldots, X_k) < g(X_1, X_k)$, $f_{X_2}(X) < g(X_2, X_k)$, and so on. A looser bound would be as follows which only depends on the two largest sets (X_{k-1} and X_k).

$$P_e < (k-1)g(X_{k-1}, X_k) \tag{8}$$

This is because $f_{X_i}(X) < f_{X_{k-1}}(X), \forall i < k-1$, meaning that starting with less number of nodes deciding on a specific state would result in lower probability of ending in that particular state. Consider the case where $X_i^0 = \alpha_i N$, where $(\alpha_1 < \cdots < \alpha_k)$, and the result for the probability of error for two choices in [4]. In this case $X_{k-1}^0 + X_k^0 = (\alpha_{k-1} + \alpha_k)N$. Accordingly we would have the following theorem,

Theorem 1 *The bound of the probability of error for the multivalued consensus on k choices would be:*

$$\frac{1}{N}\log_2 P_e \leq -(\alpha_{k-1} + \alpha_k)\left[1 - H\left(\frac{\alpha_k}{\alpha_{k-1} + \alpha_k}\right)\right] + \frac{\log_2(k-1)}{N} \tag{9}$$

where $H(x) = -\log_2(x) - (1-x)\log_2(1-x)$ *for* $x \in [0, 1]$.

Equation (9) shows that the probability of error decreases exponentially. The rate of the decay depends on α_k, the portion of the majority state and α_{k-1} the state which has the highest number of votes among the other states. Therefore, adding an extra state improves the probability of error significantly compared with the voter model where $P_e = 1 - \alpha_k$ regardless of the size of the network.

2.3 Convergence Time

We now try to find the convergence time of the algorithm. We know that $\sum_{i=1}^{k} X_i^t + U^t = N, \forall t$. Now if we define states $x_{i,N}^t = X_i^t/N$, $1 \leq i \leq k$ and $u_N^t = U^t/N$, similar to [4], the Markov Process is a density dependent Markov Jump Process, and by the results in [3], $(x_{1,N}^t, \ldots, x_{k,N}^t, u_n^t)$ converges on any compact time interval to $(x_1^t, \ldots, x_k^t, u^t)$, given by the following series of differential equations:

$$\frac{dx_i^t}{dt} = x_i^t u^t - x_i^t \left(\sum_{j=1}^{i-1} x_j^t - \sum_{j=i+1}^{k} x_j^t\right)$$

We know that $u^t = 1 - \sum_{i=1}^{k} x_i^t$, as a result:

$$\frac{dx_i^t}{dt} = x_i^t \left(1 - x_i^t - 2 \sum_{j=1}^{i-1} x_j^t - 2 \sum_{j=i+1}^{k} x_j^t \right) \tag{10}$$

Theorem 2 *Considering (10), for $x_1^0 < \cdots < x_k^0$, the time t^N to reach $(x_1^t, \ldots, x_k^t)$ so that $x_1^{t^N} \sim 1 - 1/N$, $\sum_{i=2}^{k} x_i^{t^N} \sim 1/N$ of convergence is the following:*

$$t^N \sim \log N \tag{11}$$

The proof is in the appendix. This shows that even with the multivalued consensus, using an extra state results in a convergence time which is logarithmic in N.

3 Simulations and Heuristics for Other Graphs

We now present the results obtained by simulations. Figure 1a shows that the probability of error of the algorithm for complete graphs decays exponentially along with our derived upper bound and Fig. 1d shows that the convergence time grows logarithmically with N which confirms our result. Also, Fig. 1b, e, c, and f show the probability of error and convergence time of the algorithm for Erdos-Renyi and regular graphs respectively which is very similar to those of complete graphs suggesting that the algorithm could be used in those types of graphs as well.

It is worth mentioning that similar to the binary consensus in [4] using an extra state not only would not improve the probability of error in some cases like the *path* graphs, but it also would take longer than the voter model to converge to consensus. Consider a path graph where voters for each choice are grouped together at the start of the algorithm, i.e. nodes number 1 to $\alpha_1 N$ are voting for choice 1, nodes $\alpha_1 N$ to $(\alpha_1 + \alpha_2)N$ vote for choice 2 and so on. Here, the algorithm exactly develops as if we are using a voter model. The only difference is the that each node needs to contact its neighbors twice before it would accept their opinion. Consequently, the probability of error would be the same as the voter model but with longer convergence time.

4 Conclusions

We showed that using an extra state for consensus on multiple choices improves the performance for complete graphs both in terms of convergence time and probability of error compared with the voter model. We proved this analytically and

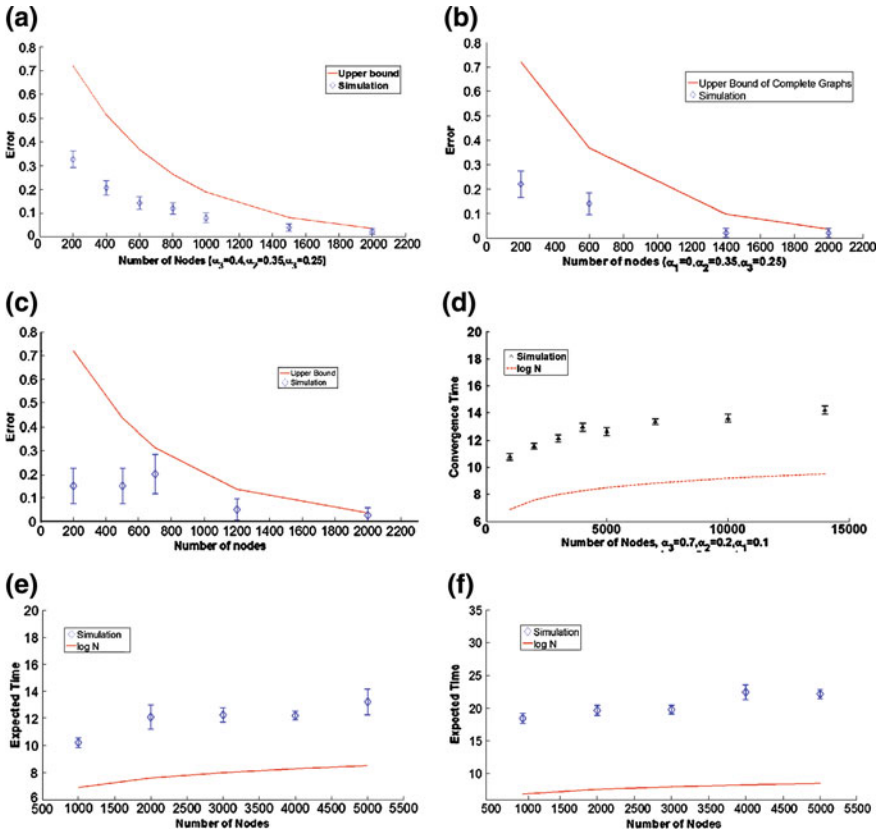


Fig. 1 Simulation results for $k = 3$ for complete, Erdos-Renyi, and regular graphs. For error plots (first row) $\alpha_1 = 0.4, \alpha_2 = 0.35, \alpha_3 = 0.25$ and for convergence time plots (second row) $\alpha_1 = 0.7, \alpha_2 = 0.2, \alpha_3 = 0.1$. The bars indicate 95 % confidence estimates. **a** Complete graph, **b** Erdos-Renyi graph, **c** regular graph, **d** complete graph, **e** Erdos-Renyi graph, **f** regular graph

confirmed it by running simulations for complete graphs. Furthermore, we showed that empirically using the algorithm is justified for other graphs such as Erdos-Renyi and regular. We mentioned that similar to the binary case using an additional state does not give any advantage over the simple voter model for special graphs like path.

Acknowledgments Moez Draief is supported by QNRF through NPRP grant number 09-1150-2-148.

Appendix

Proof of Lemma 1

Assume that $f(X)$ satisfies (4) for all $(X_1, \ldots, X_k)$. We show that this $f(X)$ also satisfies the recursion (3). First we show by induction that

$$\sum_{i=1}^{k} X_i f(X) = X_1 f(X + e_1 - e_k) + \sum_{i=2}^{k} X_i f(X - e_{i-1} + e_i) \tag{12}$$

Base case: Let $n = \sum_{i=1}^{k} X_i = k$. It is easy to check that for $(X_1, \ldots, X_k) = (1, \ldots, 1)$, both (12) and (4) yield $f_{1,\ldots,1} = \frac{1}{k}$.
Induction step: Assume that $f(X_1, \ldots, X_k)$ satisfies (12) for all $(X_1, \ldots, X_k)$ such that $\sum_{i=1}^{k} X_i \leq n - 1$. Now, for any $(X_1, \ldots, X_k)$ satisfying $\sum_{i=1}^{k} X_i = n$, the induction implies the following for $(X_1 - 1, X_2, \ldots, X_k)$,

$$\left(X_1 - 1 + \sum_{i=2}^{k} X_i\right) f(X - e_1) = (X_1 - 1) f(X - e_k) + X_2 f(X - 2e_1 + e_2) + \sum_{i=3}^{k} X_i f(X - e_1 - e_{i-1} + e_i) \tag{13}$$

If we rewrite (13) for the other $k - 1$ points ($(X_1, X_2 - 1, \ldots, X_k)$ and so on) we would have series of equations. If we sum up these equations and add $X_1 f(X - e_k) + \sum_{i=2}^{k} X_i f(X - e_{i-1})$, this would add up to (12). Next, the following can be proved using (4) for $(X_1 + 1, X_2, \ldots, X_k), \ldots, (X_1, \ldots, X_k + 1)$ and also using (12) and its variations.

$$\sum_{i=1}^{k} X_i f(X) = \sum_{i=1}^{k} X_i f(X + e_i) \tag{14}$$

The proof of the lemma follows by noting that (3) is a linear combination of (4) and (14) and therefore, (4) is a solution of (3).

Proof of Lemma 2

The proof of Lemma 2 needs an induction within induction. One for the number of nodes and one for the number of choices. The following proof is for three states. The logic is the same for $k > 3$.

Base Case: Let $n = 3$ and $X_1 + X_2 = 2$. By symmetry $f(X_1, X_2, X_3) = \frac{1}{3}$ which is less than $f(X_1, X_2) = \frac{1}{2}$.

Induction Step: Now assume for all (X_1, X_2, X_3) that $X_1 + X_2 + X_3 \leq N - 1$ satisfies (5). The following equations are then true:

$$f(X_1, X_2, X_3 - 1) \leq g(X_1, X_2),\ f(X_1, X_2 - 1, X_3) \leq g(X_1, X_2 - 1),$$
$$f(X_1 - 1, X_2, X_3) \leq g(X_1 - 1, X_2) \quad (15)$$

It is proved in [4] that $g(X_1, X_2) = 1/2g(X_1 - 1, X_2) + 1/2g(X_1, X_2 - 1)$. Using (15) and (4) would then prove (5) for $X_1 + X_2 + X_3 = N$.

Proof of Theorem 2

Here, for simplicity in showing notations we have assumed $X_1^0 > \cdots > X_k^0$. Also, $x_i = x_i^t$. Writing (10) for x_1, x_2, and x_3 it is not difficult to see that

$$d \log \left(x_1 (x_2 - x_3) \prod_{i=4}^{k} x_i \right) = \left(k - 1 - (2k - 3) \sum_{i=1}^{k} x_i \right) dt \quad (16)$$

$$d \log \prod_{i=1}^{k} x_i = \left(k - (2k - 1) \sum_{i=1}^{k} x_i \right) dt \quad (17)$$

Using (16) and (17) for the case where $x_1^{t^N} \sim 1 - 1/N$, $x_i^{t^N} \sim \alpha_i / N, \forall i > 1$, and $\sum_{i=2}^{k} x_i^{t^N} = 1/N$, the time of convergence would be the following

$$t^N \sim \log N$$

References

1. Hassin, Y., Peleg, D.: Distributed probabilistic polling and applications to proportionate agreement. Inf. Comput. **171**(2), 248–268 (2002)
2. Jung, K., Kim, B.Y., Vojnovic, M.: Distributed ranking in networks with limited memory and communication. http://research.microsoft.com (2011)
3. Kurtz, T.G.: Approximation of Population Processes. Society for Industrial and Applied Mathematics-Social Science, Philadelphia (1981)
4. Perron, E., Vasudevan, D., Vojnovic, M.: Using three states for binary consensus on complete graphs. In: Proceedings of IEEE INFOCOM, pp. 2527–2535. April (2009)

Optimization of Binary Interval Consensus

Arta Babaee and Moez Draief

Abstract Motivated by binary interval consensus algorithm in [1], the bounds for the time of convergence of this type of consensus [4], and using the optimization techniques for doubly stochastic matrices [2, 3], we introduce a distributed way to optimize binary interval consensus. With binary consensus problem, each node initially chooses one of the states 0 or 1 and the goal for the nodes is to agree on the state which was initially held by the majority. Binary interval consensus is a specific type of binary consensus which uses two intermediate states along with 0 and 1 to reduce the probability of error to zero. We show that if the probability of the nodes contacting each other is defined by a doubly stochastic matrix, the optimization of binary interval consensus can be done by reducing the second largest eigenvalue of the rate matrix Q.

1 Introduction

A problem of interest in the context of distributed algorithms is the *binary consensus problem* with which the nodes try to come to an agreement on one of two available choices based on the opinion of the majority (e.g. 0 and 1). The most famous binary consensus algorithm is the *voter model* which has been fully investigated in [6]. In the asynchronous voter model initially nodes choose one of the two states and then at each time step one of the nodes contacts one of its neighbors and simply accepts

A. Babaee (✉) · M. Draief
Intelligent Systems and Networks Group, Department of Electrical and Electronic Engineering, Imperial College London, London, UK
e-mail: ab3608@imperial.ac.uk

M. Draief
e-mail: mmd@imperial.ac.uk

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_29,

its opinion. It is proved that the probability of error depends on the degree of the nodes which were initially at the minority.

It was in [1] that a binary consensus algorithm was introduced which converged to the right result with almost sure probability. This was achieved by adding two additional states (compared with the voter model) and also using a two way communication between nodes. In [4], this algorithm was investigated in terms of convergence speed. In this paper we call this type of binary consensus *binary interval consensus* as it is called in [4].

One part of interest with any distributed algorithm is to optimize the convergence time. However, the optimization usually needs to be done in a distributed fashion to be most useful in the context of distributed algorithms. We try to optimize the binary interval consensus using the same techniques used for optimizing the averaging algorithm in [2]. These optimization techniques are usually applied to the probability matrix P which defines the probability of nodes contacting each other in the network. On the other hand, the bounds on the convergence time which have been found in [4] are dependent on the eigenvalues of some contact rate matrices related to the rate matrix Q that governs the consensus process. Therefore, we first find the relationship between Q and P and then try to optimize the eigenvalues of Q through optimizing the eigenvalues of P.

2 Binary Interval Consensus

We now give an overview of binary interval consensus algorithm in [1].

2.1 Algorithm

With binary interval consensus each node can have four states. In [4] these states are denoted by 0, 0.5^-, 0.5^+, and 1 where $0 < 0.5^- < 0.5^+ < 1$. Here, being in state 0 or 0.5^- means that a node believes the initial majority was 0 or equivalently the average values of the nodes is between 0.5 and 0.

Consider the setting where each node i has a clock which ticks at the Poisson rate of $\frac{1}{2}$. Every time this clock ticks, node i contacts one of its neighbors j with probability P_{ij}. Therefore, node j is contacted by i with Poisson rate $q_{i\to j} = \frac{1}{2}P_{ij}$. Similarly, $q_{j\to i} = \frac{1}{2}P_{ji}$. Note that here we consider P as a doubly stochastic matrix. In the setup of [4], each pair of nodes interact at instances of a Poisson rate $q_{i,j}$ (where $q_{i,j} = q_{j,i} \neq 0$ if $(i, j) \in E$ where E is the set of edges). Therefore $q_{ij} = q_{ji} = q_{i\to j} + q_{j\to i} = \frac{1}{2}(P_{ij} + P_{ji})$. Accordingly, the rate matrix Q in [4] is defined as follows,

$$Q(i,j) = \begin{cases} q_{ii} = -\sum_{l \in V} q_{il} & i = j \\ q_{ij} & i \neq j \end{cases} \tag{1}$$

where V is the set of vertices. Note that using this setup the contact rate of 1 is guaranteed for any node i $\left(\sum_j q_{ij} = \sum_j \frac{1}{2}(P_{ij} + P_{ji}) = 1\right)$. Furthermore, the following relationship exists between P and Q,

$$Q = \frac{1}{2}(P + P^T) - I_n \tag{2}$$

where I_n is the identity matrix of size n (number of nodes). We denote $(P + P^T)/2$ by P'. Now consider the interaction between any pair of nodes (i, j). At each contact of the two nodes i, j their states get updated using the following:

$$\begin{aligned} (0, 0.5^-) &\rightarrow (0.5^-, 0), (0, 0.5^+) \rightarrow (0.5^-, 0), (0, 1) \rightarrow (0.5^+, 0.5^-) \\ (0.5^-, 0.5^+) &\rightarrow (0.5^+, 0.5^-), (0.5^-, 1) \rightarrow (1, 0.5^+), (0.5^+, 1) \rightarrow (1, 0.5^+) \\ (s, s) &\rightarrow (s, s), \quad \text{for} \quad s = 0, 0.5^-, 0.5^+, 1 \end{aligned}$$

The number of nodes in both states 1 and 0 will decrease by 1 only when a node in state 1 interacts with a node in state 0. We denote the set of nodes in state i at time t by $S_i(t)$, also $|S_i| \equiv |S_i(0)|$ $(i = 0, 1)$. If nodes with state 0 are the majority, and α denotes the fraction of nodes in state 0 at the beginning of the process, $\frac{1}{2} < \alpha \leq 1$, and therefore $|S_0| = \alpha n$ and $|S_1| = (1 - \alpha)n$. As the number of nodes in state 1 and 0 decreases at the encounters between 0 and 1, finally there will be no nodes in state 1 left in the network. Also, the number of nodes in state 0 will become $|S_0| - |S_1|$ at the end of the process. There will be only nodes in state 0.5^+, 0.5^-, and 0 left. Next, the number of nodes in state 0.5^+ will decrease when they interact with nodes in state 0 and consequently after some time the nodes in state 0.5^+ will also disappear and only nodes with state 0 or 0.5^- will remain. At the end of this stage the algorithm reaches the consensus. This means that all the nodes agree that the average is in the interval [0,0.5) which indicates that nodes with state 0 initially had the majority.

In [4], the upper bounds for the expected time for each of these phases have been derived in terms of the eigenvalues of a set of matrices that depend on Q. If S is considered as a non-empty subset of V, Q_S is defined as:

$$Q_S(i,j) = \begin{cases} -\sum_{l \in V} q_{il} & i = j \\ q_{ij} & i \notin S, j \neq i \\ 0 & i \in S, j \neq i \end{cases} \tag{3}$$

The following lemma is then derived:

Lemma 1 *For any finite graph G, there exists $\delta(G, \alpha) > 0$ such that, for any non-empty subset of vertices S ($|S| < n$), if $\lambda_1(Q_S)$ is the largest eigenvalue of Q_S, then it satisfies*

$$\delta(G,\alpha) = \min_{S\subset V,\, \frac{|S|}{n}\in[2\alpha-1,\alpha]} |\lambda_1(Q_S)| = \min_{S\subset V, |S|=(2\alpha-1)n} |\lambda_1(Q_S)|$$

Note that using this definition, for all non-empty set S, $\delta(G,\alpha) > 0$ because $\lambda_1(Q_S) < 0$.

Theorem 1 *If T is considered as the time of convergence (i.e. the time it takes for nodes in states 1 and 0.5^+ to deplete), it will be bounded as follows,*

$$\mathbb{E}(T) \leq \frac{2}{\delta(G,\alpha)}(\log n + 1). \tag{4}$$

The convergence time directly depends on $\delta(G,\alpha)$. In Sect. 3 we show that the bounds of δ depend on the eigenvalues of Q and therefore $\delta(G,\alpha)$ could be increased by optimizing the eigenvalues of Q. Considering (4) increasing $\delta(G,\alpha)$ would then mean the decrease of the bound on the convergence time. Motivated by the use of Semi-Definite Programming (SDP) in optimizing the speed of distributed averaging algorithm in [2] we try to optimize the eigenvalues of Q using the same techniques.

3 Our Findings

Our main result is the following:

Theorem 2 *If $\delta(G,\alpha)$ is defined by Lemma 1 it would be bounded by the second largest eigenvalue of the rate matrix ($\lambda_2(Q)$) as follows,*

$$-\lambda_2(Q)/4 \leq \delta(G,\alpha) \leq -\lambda_2(Q)$$

Theorem 2 yields that minimizing $\lambda_2(Q)$ would lead to the increase of $\delta(G,\alpha)$. This would then decrease the convergence time and hence the problem of optimization of the convergence time can be reduced to the problem of optimizing the eigenvalues of Q. The proof of the theorem consists of the following lemmas.

Lemma 2 *The following relationship exists between the diagonal elements of the P matrix and $\delta(G,\alpha)$,*

$$\frac{1 - max\ P_{ii}}{2} \leq \delta(G,\alpha)$$

Proof Define $A = [a_{ij}]$ as an irreducible nonnegative matrix of size n. Furthermore, let ρ be the largest eigenvalue of A, i.e. $\rho(A) = \lambda_1(A)$ where $\lambda_1 > \cdots > \lambda_n$. Also, for $U \subset \langle n\rangle$ ($\langle n\rangle = \{1,\ldots,n\}$) let $A(U)$ be the principle submatrix of A whose rows and columns are in U and set $\rho_m(A) = \max_{U\subset\langle n\rangle, |U|=m}\rho(A(U))$. Considering the result in [5] regarding the eigenvalues of nonnegative matrices and applying it to P', it is not difficult to derive the following result:

$$\rho_s(P') \leq 1 - \frac{1}{2}(1 - \max_i P'_{ii})$$

for $s = 1, \ldots, n-1$. As $\lambda(Q) = \lambda(P') - 1$ and also considering the definition of $\delta(G, \alpha)$ in Lemma 1, this immediately results in the following,

$$\delta(G, \alpha) \geq \frac{1}{2}(1 - \max_i P_{ii})$$

where P_{ii}s are the diagonal elements of P matrix. Note that $P'_{ii} = P_{ii}$.

Lemma 3 *Consider the $\delta(G, \alpha)$ in Lemma 1 and $\lambda_2(Q)$, the second largest eigenvalue of the Q matrix. The following relationship is then true,*

$$\delta(G, \alpha) \leq -\lambda_2(Q)$$

Proof Writing the characteristic polynomial of Q_S would yield that $\lambda_1(Q_S)$ is maximum when the size of Q_S is maximum (i.e. $n-1$). Using the *Cauchy's Interlace Theorem* in [8], when $\lambda(Q_S)$ is maximum it is between $\lambda_2(Q)$ and $\lambda_1(Q)$. Considering the definition of $\delta(G, \alpha)$ this would conclude the proof.

Lemma 4 *Considering the Q matrix which denotes the rate of the interactions between nodes, the following is true.*

$$-\lambda_2(Q) \leq 2(1 - max\, P_{ii})$$

Proof Using *Gershgorin Theorem* in [7] for $\lambda_2(Q)$ would prove the result.

4 Optimizing the Convergence Time Using SDP

The following is the immediate result of (2),

$$\lambda_i(Q) = \lambda_i\left(P'\right) - 1 \tag{5}$$

This means that decreasing the eigenvalues of P' will decrease the eigenvalues of Q which changes the problem of optimization of Q to optimization of P' which is both stochastic and symmetric. It is known that the sum of any number of the largest eigenvalues of a symmetric matrix is a convex function of the matrix (e.g. $\lambda_1 + \lambda_2$ where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$) [3]. P' is a stochastic matrix so $\lambda_1(P') = 1$ and consequently $\lambda_2(P') = (\lambda_1(P') + \lambda_2(P')) - 1$ is a convex function and can be optimized using convex optimization techniques, one of which is SDP. Therefore, the convex optimization problem will be as follows:

$$\text{minimize } \lambda_2(P') \quad \text{subject to } P_{ij} \geq 0,\ P_{ij} = 0 \quad \text{if} \quad i, j \notin E$$
$$\text{and } \sum_j P_{ij} = 1, \forall i$$

SDP cannot be solved in a decentralized manner, however, we use a subgradient method (similar to [2]) using which nodes can perform SDP locally provided that they know their corresponding eigenvalue in the eigenvector associated with P'. In [2], this subgradient method is used to optimize the eigenvalues of a matrix which governs the averaging process. We use the same technique to optimize the eigenvalues of P' which results in optimizing the eigenvalues of Q. Different methods have been developed to calculate the eigenvalues of the graph in a decentralized manner (e.g. [9]). Similar to [2], we use the fact that a distributed algorithm to calculate the eigenvalues of the graph exists with a good precision.

Our arguments are very similar to [2, 3]. Therefore, we briefly mention our algorithm and skip the details. Let P_{ij}s be assigned as elements of vector p. Also, if l is the number assigned for a non-self loop edge (i, j), it will be shown by $l \sim (i, j)$ $(i < j, l = 1, \ldots, m$, where m is the total number of non-self loop edges) and then $p_l = P_{ij}$ and $p_{-l} = P_{ji}$. As the sum of the elements in each row of P is 1 we do not need to optimize P_{ii}s. P' can be then written as follows,

$$P' = I + \frac{1}{2}\sum_{l=1}^{m}(p_l B_l + p_{-l} B_{-l})$$

where B_l matrix for $l \sim (i, j)$ is set as, $B_{l_{ii}} = B_{l_{jj}} = -1$, $B_{l_{ij}} = B_{l_{ji}} = 1$ and 0 elsewhere. To use the subgradient method, the convex optimization problem can then be defined as the following:

$$\text{minimize } \lambda_2\left(I + \frac{1}{2}\sum_{l=1}^{m}(p_l B_l + p_{-l} B_{-l})\right) \quad \text{subject to } \mathbf{1}^T \boldsymbol{p_i} \leq 1, \forall i$$
$$\text{and } \quad p_l \geq 0, 1 \leq |l| \leq m$$

where $\boldsymbol{p_i}$ is the vector of nonzero elements in row i of the P matrix i.e. $\boldsymbol{p_i} = [P_{ij};\ (i, j) \in E]$. Then, if u is the eigenvector associated with $\lambda_2(P')$, the subgradient $f(p)$ and its components would be,

$$f(p) = \frac{1}{2}\left(u^T B_{-m} u, \ldots, u^T B_m u\right)$$
$$f_l(p) = \frac{1}{2}u^T B_l u = -\frac{1}{2}(u_i - u_j)^2, l \sim (i, j), |l| = 1, \ldots, m$$

Accordingly the steps in Algorithm 1 have been used for optimization at each step k, given a feasible p. Note that in Algorithm 1, the step size β_k satisfies the diminishing rule, $\beta_k \geq 0$, $\beta_k \to 0$, $\sum_k \beta_k = \infty$.

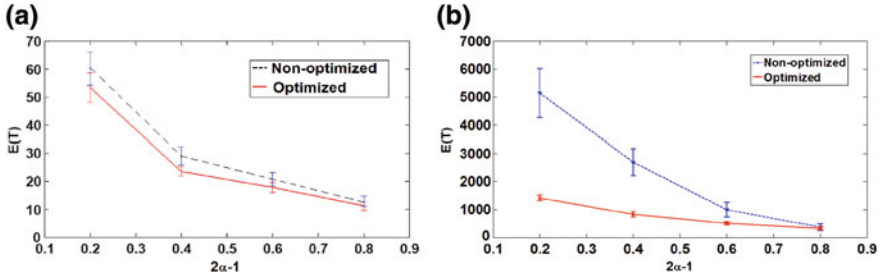


Fig. 1 Simulation results for optimized and non-optimized versions of binary interval consensus. *Left* expected time for Erdos-Renyi graph **a** of size 1024. *Right* expected time for Power law graph **b** of size 225. The bars indicate 95 % confidence estimates

Algorithm 1 Optimization

$k \leftarrow 1$
repeat
`// Subgradient Step`
Calculate $f^{(k)}$ and update p, $p \leftarrow p - \beta_k f^{(k)}$
`// Sequential Projection`
$p_l \leftarrow \max\{p_l, 0\}, |l| = 1, ..., m$
for each node $i = 1, ..., n$, $\mathscr{L}(i) = \{l |$ edge l connected to $i\ \}$ **do**
  while $\sum_{l \in \mathscr{L}(i)} p_l > 1$ **do**
    $\mathscr{L}(i) \leftarrow \{l | l \in \mathscr{L}(i), p_l > 0\}$
    $\delta \leftarrow \min\left\{\min_{l \in \mathscr{L}(i)} p_l, \left(\sum_{l \in \mathscr{L}(i)} p_l - 1\right) / |\mathscr{L}(i)|\right\}$
    $p_l \leftarrow p_l - \delta, l \in \mathscr{L}(i)$
  end while
end for
$k \leftarrow k + 1$

4.1 Decentralization

For decentralization, we use Algorithm 2[1] which is the Decentralized Orthogonal Iterations algorithm in [9]. This is the same algorithm used in [2] to decentralize the subgradient method for averaging.

Algorithm 2 Decentralization

Initialize the process with a random vector y_0
repeat
Set $y_k \leftarrow P' y_{k-1}$
$y_k \leftarrow y_k - \left(\sum_{i=1}^{n} \frac{1}{n} y_{k_i}\right) \mathbf{1}$ `// Orthogonalizing`
$y_k \leftarrow y_k / ||y_k||$ `// Scaling to unit norm`

[1] 1 is a vector of all ones.

Note that the first step can be performed in a decentralized fashion as P_{ij} is only nonzero when there is an edge between i, j. The second and third steps can be done by using a distributed averaging algorithm such as [10]. It is then proved in [2] that the subgradient method will converge even with considering the approximation error in calculating the eigenvalues in a distributed way. We use the same algorithm for decentralization. The difference here is the use of P' instead of the W matrix in [2] which governs the averaging process. The argument for proving the convergence of the subgradient method is also the same as [2]. Note that we have proved that the optimization should be done on $\lambda_2(Q)$. This can be done before the consensus process when the graph structure is fixed.

5 Simulations

We now give our simulations which confirm our findings. Figure 1a shows the simulation results for the Erdos-Renyi graphs. It can be seen that the optimization scheme has reduced the convergence time for different voting margins. Figure 1b shows the results for the Power-Law graphs. Here, the effect of optimization is clearer as the degree distribution is not homogeneous anymore and different nodes have higher difference in their degrees. However, as it can be seen the optimization works much better for lower voting margins where the number of votes are close. Note that for our simulations we are considering only the effect of optimization and not decentralization and therefore it is assumed that nodes know their corresponding eigenvalue in the eigenvector of P'.

6 Conclusion

We have found a distributed optimization technique for binary interval consensus process using SDP. For this we related the time of convergence to the second largest eigenvalue of the rate matrix governing the algorithm. Our results are for general graphs and we confirmed them by simulating our optimization scheme for Erdos-Renyi and Power-Law graphs.

Acknowledgments Moez Draief holds a Leverhulme Trust Research Fellowship RF/9/RFG/2010/02/08.

References

1. Benezit, F., Thiran, P., Vetterli, M.: Interval consensus: from quantized gossip to voting. In: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '09, pp. 3661–3664. IEEE Computer Society, Washington, DC (2009)
2. Boyd, S., Ghosh, A., Prabhakar, B., Shah, D.: Randomized gossip algorithms. IEEE Trans. Inf. Theory **52**(6), 2508–2530 (2006)

3. Boyd, S., Diaconis, P., Xiao, L.: Fastest mixing markov chain on a graph. SIAM Rev. **46**(4), 667–689 (2004)
4. Draief, M., Vojnović, M.: Convergence speed of binary interval consensus. SIAM J. Control Optim. **50**(3), 1087–1109 (2012)
5. Friedland, S., Nabben, R.: Fakultat Fur Mathematik: On the second real eigenvalue of nonnegative and z-matrices. Linear Algebra Appl. **255**, 303–313 (1997)
6. Hassin, Y., Peleg, D.: Distributed probabilistic polling and applications to proportionate agreement. Inf. Comput. **171**(2), 248–268 (2002)
7. Horn, R.G., Johnson, C.R.: Matrix Analysis. Cambridge University Press, London (1985)
8. Hwang, S.-G.: Cauchy's interlace theorem for eigenvalues of hermitian matrices. Am. Math. Mon. **111**(2), 157–159 (2004)
9. Kempe, D., McSherry, F.: A decentralized algorithm for spectral analysis. In: Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of computing, STOC '04, pp. 561–568. ACM, New York (2004)
10. Xiao, L., Boyd, S.: Fast linear iterations for distributed averaging. Syst. Control Lett. **53**(1), 65–78 (2004)

Team Formation in Social Networks

Meenal Chhabra, Sanmay Das and Boleslaw Szymanski

Abstract It is now recognized that the performance of an individual in a group depends not only on her own skills but also on her relationship with other members of the group. It may be possible to exploit such synergies by explicitly taking into account social network topology. We analyze team-formation in the context of a large organization that wants to form multiple teams composed of its members. Such organizations could range from intelligence services with many analysts to consulting companies with many consultants, all having different expertise. The organization must divide its members into teams, with each team having a specified list of interrelated tasks to complete, each of which is associated with a different reward. We characterize the skill level of a member for a particular task type by her probability of successfully completing that task. Members who are connected to each other in the social network provide a positive externality: they can help each other out on related tasks, boosting success probabilities. We propose a greedy approximation for the problem of allocating interrelated tasks to teams of members while taking social network structure into account. We demonstrate that the approximation is close to optimal on problems where the optimal allocation can be explicitly computed, and that it

This work was supported in part by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 and in part by an NSF CAREER Award (0952918). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies either expressed or implied of the Army Research Laboratory. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

M. Chhabra (✉) · S. Das · B. Szymanski
Department of Computer Science, Rensselaer Polytechnic Institute,
Troy, NY, USA
e-mail: chhabm@cs.rpi.edu

S. Das
e-mail: sanmay@cs.rpi.edu

B. Szymanski
e-mail: szymansk@cs.rpi.edu

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_30,

provides significant benefits over the optimal allocation that does not take the network structure into account in large networks. We also discuss the types of networks for which the social structure provides the greatest boost to overall performance.

1 Introduction

Good team-formation is one of the keys to success of any organization. Many researchers emphasize that the performance of an individual in a group depends not only on her skills but also on her relationships with other members of the group [3, 6]. Relationships between team members do not necessarily have to be friendship relations. The important question is whether members work together synergistically—how compatible they are in collaborative environments (in fact, there is empirical evidence, at least among MBA students, that friendship within teammates is *negatively* correlated with performance [1]). Increasingly, organizations are attempting to optimize team composition, for example by taking personality types into account when forming teams to work on tasks [3].

We are interested in the problem of optimal allocation of members to tasks within an organization, taking into account their network of relationships with others. We consider a model where there are multiple task types, and each member has a certain cognitive ability for each task type. Each member is therefore characterized completely by her ability to perform each of the different types of tasks. The organization has a set of projects to be performed, each project consisting of a set of tasks of different types. Accordingly, each task is characterized by its value, its type, and the project to which it belongs. The motivating idea is that the organization is a group of experts (consultants in a service company, or analysts in an intelligence office). Each expert has expertise in a particular task type regardless of the project to which this task belongs. For example an accounting consultant can analyze finances of different companies or transportation system analysts can analyze military movements in many regions or countries. Experts on different task types may have a synergistic effect on each others' performance when they are allocated to the same project by fruitfully sharing information that could be valuable for different task types.

We consider the problem of optimal allocation of members to tasks in this framework. We introduce a model that captures the elements described above, and then demonstrate that taking social network structure into account can have significant benefits in terms of the overall optimality of task allocation. We introduce a greedy algorithm for the problem of task allocation taking social network structure into account, and demonstrate experimentally that it is a good approximation to the optimal allocation on small graphs. Then, we show that it achieves significant benefits on large graphs compared with the optimal solution that does not take social network structure into account. Finally, we use this greedy algorithm to explore the properties of different kinds of social networks, and find that the most affected graphs are small world networks, followed by random graph networks, and the least affected graphs are preferential attachment networks. Our work is related to previous papers

that demonstrate the effect of underlying social graph structure on team performance [4, 5]. It is perhaps closest to the work of Lappas et al., who also consider the social graph structure of individuals while forming teams [6]. However, their focus is different; in their model, agents have binary skills, and each task is focused on the composition of appropriate skill sets. In contrast, our work focuses on optimal resource allocation in a utility-theoretic framework.

2 The Model

There are n experts $E_1, \ldots, E_n$ who work for an organization. The social network representing synergistic relationships is given by S. Two experts are connected in S if they boost each others' performance when they work on the same team assigned to a particular project.

There are z different task types. Each expert has a different skill level associated with each of these z task types. The skills of an expert E_i are thus represented by a vector $S_i = (s_{i1}, s_{i2}, \ldots, s_{iz})$ where s_{ij} is the probability that E_i can complete a task of type j successfully. This is a general measure; it may include not only technical skills but also interpersonal or organizational skills. We assume that the manager responsible for task allocation (M) knows the skills vector for each expert.

Time proceeds in discrete steps and at every T units of time, Manager M allocates work to the experts for the next T units; we call this period of T units a 'round'. Manager M allocates m different projects $R_1, \ldots, R_m$. Each project R_i has q tasks $T_1^1, \ldots, T_q^i$. The distribution of task types is the same for each project. Each task T_i of project R_j (represented by T_i^j) is associated with a value V_i^j which represents the gains received by the organization for successfully completing the task (this is a direct measure of utility). Therefore, each available task in the organization is characterized by three attributes: task type, project to which it belongs, and value. For convenience, we construct a vector of probabilities of successful completion of tasks in a given project R_k for each expert E_i, $P_i^k : (p_{i1}^k, \ldots, p_{iq}^k)$ from S_i, the vector of skills, as defined earlier.

We assume the manager re-allocates each of the q different tasks in all the m projects at the beginning of each round; therefore, overall there are mq tasks to be allocated to the n experts. The values of tasks do not remain the same at each round, so the allocation algorithm has to be run at the beginning of each round. The tasks are designed such that they can be finished in a round. An expert can only be assigned one task.

Network Effects: The social network S of experts is known to the manager. This is not an unrealistic assumption, because while working with these experts, the manager may have acquired this information. There are also certain personality tests, like Myers-Briggs, Kolbe Conative Index etc., which the manager could conduct to discover expert types which could be used to build a network based on compatibilities

between different expert types. Now we specify an explicit model of how experts can help each other out in performing their tasks.

Let W represent the friendship adjacency matrix created from the social graph S:

$$W_{ij} = \begin{cases} 1 & \text{if } E_i \ \& \ E_j \text{ are friends,} \\ 0 & \text{otherwise} \end{cases}$$

Let f_i^k represent the number of friends of E_i among experts assigned a task of project R_k. We model the network effect as a boost in the probability that an expert successfully completes a task. Let $B = 1 - e^{-f_i^k}$ denote a coefficient that represents the improvement in the performance of an expert resulting from collaborating on a task with friends; B defines the fraction of the performance gap $(1 - p_{ij}^k)$ that is covered by collaboration. By definition, $0 < B < 1$. Then, the boosted probability that expert i successfully accomplishes task T_j in the project R_k, denoted as p_{ij}^{k+} is defined as:

$$p_{ij}^{k+} = \frac{p_{ij}^k}{1 - B(1 - \max(p_{ij}^k, 1/2))} \tag{1}$$

so the higher the coefficient B, the greater the boost. The boost is naturally limited by the factor $1 - p_{ij}^k$, and we further cap it at covering the gap of $\frac{1}{2}$ for lower p_{ij}^k's. The denominator of the boosted probability expression is at most $\min(p_{ij}^k, 1/2)$ so the boosted probability is less than $\min(2p_{ij}^k, 1)$.

3 Algorithms for Task Allocation

The manager's goal is to maximize expected utility. Let us define indicator variables a_{ij}^k as follows:

$$a_{ij}^k = \begin{cases} 1 & \text{if } E_i \text{ is allocated task } T_j \text{ in project } R_k \\ 0 & \text{otherwise} \end{cases}$$

Allocation Ignoring Network Effects: If there is no network effect then the performance of any task depends on the selected experts' skill levels. Then the total expected utility is given by:

$$U = \sum_{i=1}^{n} \sum_{j=1}^{q} \sum_{k=1}^{m} a_{ij}^k p_{ij}^k V_j^k \tag{2}$$

The objective is to find allocation variables a_{ij}^k such that Eq. (2) is maximized.

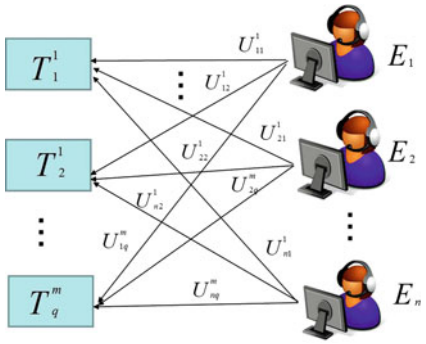


Fig. 1 The *left side* represents tasks and the *right side* represents experts. The weight U_{ij}^k on the edge between expert E_i and task T_j^k is the expected return value if E_i is assigned task T_j^k. The objective is to find the matching such that the total expected return value is maximized. It is easy to observe that the optimal allocation is the same as the maximum weight matching

Since the manager knows the skill level of all the experts, this problem reduces to maximum weight matching in a bipartite graph with experts on one side and tasks on the other, as shown in Fig. 1. The weight of the edge between expert E_i and task T_j^k is equal to the expected value if E_i is assigned to task T_j^k, i.e., the product of the probability of successful completion p_{ij}^k and the value associated with that task V_j^k. In our experiments we use the Hungarian algorithm to solve the maximum weight matching problem and find the optimal allocation.

Note that, even though the manager ignores network effects when deciding on the task allocation, the effects still come into play in terms of overall performance. Therefore, when we compare solutions that take network effects into account, we include the hidden network effect in the utility term after the allocation has been done without including that term when deciding allocation.

Taking Network Effects Into Account: The total expected utility can be calculated by updating the success probability in Eq. (2) to take into account network effects:

$$U_f = \sum_{i=1}^{n} \sum_{j=1}^{q} \sum_{k=1}^{m} a_{ij}^k p_{ij}^{k+} V_j^k \tag{3}$$

We need to assign a_{ij}^k's such that the total expected utility U_f is maximized. Let $\text{Opt} = \max_{a_{ij}^k}(U_f)$ represent the maximum expected utility that can possibly be achieved. This problem is computationally hard to solve optimally, so we focus on a greedy approximation algorithm.

A Greedy Approximation: We propose a greedy approximation algorithm that can be used by the organization for task allocation and works as follows. First, construct a weighted bipartite graph with experts on one side and tasks on the other (this is distinct from the graph representing social synergies among experts). As above, the

weight on an edge between expert E_i and task T_j^k is equal to the expected return value if the expert E_i is assigned to task T_j^k, i.e., the product of the probability that expert E_i can finish the task T_j^k and the return value associated with task T_j^k. Second, select a link with maximum weight and assign the task on the link to the corresponding expert. Update the success probabilities of all experts connected to this one in the graph of social synergies for all tasks that are on project k. Repeat this process until there are no more tasks or no more experts. The overall complexity of this algorithm is $O(\min(n, mq)nmq)$.

4 Experimental Results

Network Models: We consider three standard network models:

1. Random Graph Network (RGN): We use the $G(n, p)$ Erdos-Renyi model for random graph generation.
2. Small World Network (SWN): We use the β model of Watts and Strogatz [7]. The network is represented by a tuple (n, k, β), where n is the number of nodes in the network, k is the mean degree of each node, and β is a parameter such that $0 \leq \beta \leq 1$ which represents randomness.
3. Preferential Attachment Network (PAN): This network model captures the "rich get richer" phenomenon. We use the mechanism of Barabasi et al. [2] to generate these networks.

Testing the Greedy Algorithm: We compare the performance of the greedy algorithm described earlier with respect to the optimal allocation. We know of no efficient means of computing the optimal allocation when taking network structure into account. As validation, we consider graphs with a small number of experts, so that brute-force search is feasible for finding the optimal solution. We consider a network with 10 experts and two projects, each with five tasks. The average degree is 2. There is only one task type (equivalently, all experts are equally proficient in carrying out any task). The probability of success is 0.2 for any expert to successfully complete any task. The return value of each task is an i.i.d. sample from a Gaussian distribution with $\mu = 1$ and $\sigma = 0.05$.

The results are shown in Table 1. We observe that the greedy approximation yields close to optimal results. It is interesting to note the variation in performance with respect to network topology. There is a significant change in the optimal utility when the underlying network creation model is modified. For example, the order of optimal utility is SWN > RGN > PAN. This holds even though the average degree is kept constant across the different types of networks. We also observe that for small world networks (SWN), the changes in rewiring factor do not affect the optimal performance achievable; however, there is a slight increase in the performance of the greedy algorithm as the rewiring probability β decreases. One caveat is that these are small networks, so the approximation may be worse in larger networks.

Table 1 Comparison of greedy and optimal allocations

Network type	(k)	β	Pr (Success)	Utility w/o network	Opt utility	Greedy
SWN	2	0.25	0.2	2.75	3.30	3.20
SWN	2	0.10	0.2	2.76	3.30	3.24
SWN	2	0.00	0.2	2.77	3.29	3.29
PAN	2	–	0.2	2.64	3.06	2.72
RGN	2	–	0.2	2.66	3.11	2.98
RGN	2	–	$\mathcal{N}(0.2, 0.05^2)$	2.70	3.16	3.04

k denotes mean degree, 'Utility w/o network' is the utility when allocations are made without considering the social network, i.e., using the maximum weight matching approach described in Sect. 3, but the utility is calculated using Eq. (3); 'Opt Utility' is calculated by finding the allocation which maximizes Eq. (3); Greedy allocates tasks using the algorithm described earlier (utility is again calculated using Eq. (3)). Two key observations are that: (1) Not considering the network structure in allocation is significantly suboptimal; (2) The greedy allocation yields almost optimal performance

Table 2 For medium-size organizations: Experimental results when 480 experts are assigned to 48 projects, each with 10 tasks

Network type	(k)	β(Randomness)	Pr (Success)	Utility w/o network	Greedy	UR
SWN	96	0.25	0.2	154.6	192.0	1.24
SWN	96	0.00	0.2	155.0	191.9	1.23
PAN	96	–	0.2	152.0	185.8	1.22
RGN	96	–	0.2	155.1	189.8	1.22

"UR" represents utility ratio

Understanding the approximation properties of this algorithm is an interesting open research question.

This experiment also demonstrates that there is a significant advantage to considering network effects during team formation. Although, we cannot calculate the optimal utility for larger networks due to computational costs, if we can show that the utility attainable using the greedy algorithm is significantly higher than that attained when network effects are not taken into account, this is a lower bound on the gains that could be achievable. We thus turn our attention to exploring utility differences between the greedy allocation and the optimal allocation that ignores network effects in larger networks.

Real World Networks: For the rest of this paper we use the term *Utility Ratio* (UR) as the ratio of the utility achieved using the greedy algorithm and the utility achieved by optimizing the allocation without considering the network effect (although of course the network effect is taken into account in computing the actual utility). Table 2 shows that the possible gains from smarter task allocation strategies really become evident when we look at larger organizations. Again note that while the percentage gain from considering network structure in allocation is roughly equivalent for the three types of networks, the overall utility tends to be higher for small world networks especially.

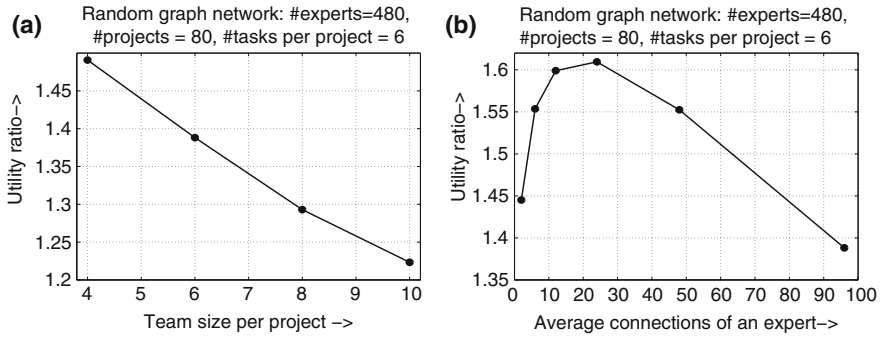


Fig. 2 As the team size increases, the likelihood of an expert having a neighbor in his/her team also increases in any allocation; therefore, the utility ratio decreases with the increase in team size. As the average connectivity increases, initially the utility ratio increases because there are more experts available who can boost a given expert's productivity; however, it later decreases. Again, this is because the likelihood of having an expert's neighbors working on the same project by chance also increases in any allocation strategy

Effects of Team Size and Connectivity: Figure 2 shows benefits achieved by considering network effects as a function of team size and of connectivity for random graph networks. The benefit diminishes with increasing team size if the connectivity of the social network remains constant. This is because the chances of having a socially synergistic expert in a large team even with a random allocation (or one that doesn't take network structure into account) is higher than it would be if teams were small. For connectivity, the utility ratio initially increases because of the increase in socially synergistic experts assigned to the same project, but later decreases as connectivity increases. Again, this decrease is because as connectivity increases, socially synergistic experts are more likely to work on the same project even if network structure is not explicitly a factor in the allocation. We experimented with other types of networks as well but the observations were similar, so we do not report them here for the sake of brevity.

5 Conclusions

Our results demonstrate the value of considering social network structure in allocation of tasks in networks of experts. We have also characterized situations in terms of graph structure, connectivity, and team size, in which organizations may find it particularly valuable to explicitly take social network structure into account in determining the allocation of experts to tasks.

References

1. Baldwin, T., Bedell, M., Johnson, J.: The social fabric of a team-based mba program: network effects on student satisfaction and performance. Acad. Manag. J. **40**(6), 1369–1397 (1997)
2. Barabási, A., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999)
3. Chen, S., Lin, L.: Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering. IEEE Trans. Eng. Manage. **51**(2), 111–124 (2004)
4. Gaston, M., desJardins M.: Agent-organized networks for dynamic team formation. In : Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 230–237. ACM, Utrecht (2005)
5. Gaston, M., Simmons, J., DesJardins, M.: Adapting network structure for efficient team formation. In: Proceedings of the AAAI 2004 Fall Symposium on Artificial Multi-agent Learning, (2004)
6. Lappas, T., Liu, K., Terzi, E.: Finding a team of experts in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 467–476. ACM (2009)
7. Watts, D., Strogatz, S.: Collective dynamics of small-world networks. Nature **393**(6684), 440–442 (1998)

Part VIII
Computer Vision II

Semi-Automatic Semantic Video Annotation Tool

Merve Aydınlılar and Adnan Yazıcı

Abstract Video management systems require semantic annotation of video for indexing and retrieval tasks. Currently, it is not possible to extract all high-level semantic information automatically. Also, automatic content based retrieval systems use high-level semantic annotations as ground truth data. We present a semi-automatic semantic video annotation tool that assists user to generate annotations to describe video in fine detail. Annotation process is partly automated to reduce annotation time. Generated annotations are in MPEG-7 metadata format for interoperability.

Keywords Video annotation · MPEG-7 · Semi-automatic annotation

1 Introduction

Advances in technology and price drop of digital cameras and broadband Internet, give rise to number of videos publicly accessible. Also, professionally created content is digitally available. However, by its very nature, querying of video data is much more complex than textual data. Video content analysis systems are successful for specific objects and concepts but their results are not satisfactory for general domains, yet. This brings importance of manual annotation for video management systems.

This work is supported in part by a research grant from TUBITAK EEEAG with grant number 109E014.

M. Aydınlılar (✉) · A. Yazıcı
Department of Computer Engineering, Middle East Technical University,
06800 Ankara, Turkey
e-mail: merve@ceng.metu.edu.tr

A. Yazıcı
e-mail: yazici@ceng.metu.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_31,

Annotations are also necessary for content based retrieval systems as training data and ground truth.

Video annotation tools provide necessary environment for annotators to describe video content. These descriptions can be low-level and high-level. While low-level descriptions are related to features like color and texture, high-level descriptions are semantic descriptions like objects, concepts etc. Low-level descriptions are mostly used for query-by-example applications and high-level descriptions are used for semantic queries. Low-level descriptions can be obtained automatically, however semantic annotations cannot be generated fully automatically. For semantic annotations, annotator can use free text, keywords, predefined vocabulary of application domain or ontology. Interoperability is an important issue for video annotation tools, since generated output will be used as input by other systems. In order to have fine detail descriptions, video must be decomposed into structural units, spatially and temporally. Manually generating these descriptions are time consuming and prone to errors. To reduce annotation time, image processing and pattern recognition techniques can be employed, but results of these techniques should be represented to annotator clearly and should be editable to ensure the consistency of the annotations.

A comprehensive semi-automatic semantic video annotation tool is presented. The annotation tool generates both low-level and high-level annotations. Video is decomposed into shots, key frames and events temporally, and spatio-temporally to still and moving regions. Temporal decompositions are obtained automatically by shot boundary detection and key frame extraction. It is possible to annotate each component of the video semantically. Low-level descriptions can be extracted both from key-frames and spatial decompositions of key-frames. Generated annotations are in MPEG-7 [1] format for interoperability. However, it is possible to generate different MPEG-7 annotations for the same video with the same semantic descriptions. To ensure interoperability further, Detailed Audio Visual Profile (DAVP) [2], is used. To reduce annotation time, object redetection, face detection and object tracking utilities are included. User defined Web Ontology Language (OWL) ontologies are supported to make semantic annotations faster and to define semantic relations between concepts.

2 Background and Related Work

There are many existing video annotation tools to generate descriptions from video content. These annotation tools can be examined in different aspects like annotation vocabulary, metadata format, content type, granularity, etc. as in [3]. In this scope, they can be roughly partitioned into two groups; fully manual ones and partly automated ones.

VIA[1] is a semantic video annotation tool that supports both low-level and high-level descriptions. Low-level features can be extracted both for frames and still

[1] http://mklab.iti.gr/via/

regions. Annotation level of the tool is entire video, shots, frames, still and moving regions. It supports live video annotation and frame by frame annotation. Generated annotations are in XML format which may cause interoperability issues. OWL ontology for representing domain knowledge and free text annotation is supported. The annotations are done manually, no automation utility is included.

Ontolog [4] supports high-level descriptions. Low-level descriptions are not included. Annotation level is entire video and its temporal decompositions, video, shot and frames. Ontolog produces Resource Description Framework (RDF) annotations. It supports both semantic concepts and semantic relations between these concepts. Annotation vocabulary of Ontolog is Resource Description Framework Schema (RDFS) ontologies. No automation is supported by this annotation tool.

VideoAnnEx [5] is a widely used semantic video annotation tool. It supports high-level descriptions, low-level feature extraction is not supported. Annotation level of the tool is video, video, shot, frame, and still region. VideoAnnEx provides a lexicon as annotation vocabulary. This lexicon is editable by user interface and also custom lexicons in MPEG-7 format can be loaded. Besides keywords from lexicon free text annotations are also supported. VideoAnnEx generates annotations in MPEG-7 format. It supports automatic shot detection and key frame extraction. Once shot boundaries are detected and key frames identified this information is printed to a separate XML file to be used for later annotations. VideoAnnEx also supports detection of similar frames to the annotated frame in order to decrease annotation time.

SVAS [6] is a semi-automatic semantic video annotation tool that consists of two separate components. The first component is Media Analyzer and the second one is Semantic Video Annotation Tool (SVAT). Media Analyzer takes video file as input and without annotator's intervention it analyzes video file. Shot boundary detection, key frame extraction and extraction of low-level features are done by Media Analyzer and saved as a MPEG-7 document and conformed to DAVP. Then, SVAT takes this file and video as input and generates semantic annotations. All of the descriptions are saved in a single MPEG-7 document. Annotation level of SVAS is entire video, shots, key frames and still regions. Still regions of a frame is determined by automatic segmentation of free hand drawing. It is not possible to extract low-level features from these regions. SVAS supports redetection of objects by using SIFT [7] features. After an object is selected, it is searched in entire video and similar matches are listed to annotator by bounding boxes. However, these bounding boxes are not editable, annotator can just delete the wrong results. Shot redetection is also include, if a shot occurs more than one in a video it can be detected.

3 System Overview

Basically, the annotation tool takes a video file and optionally an ontology file as input and with the assistance of annotator an MPEG-7 video description file is generated. The annotation tool has a built-in video player to watch the video to be annotated.

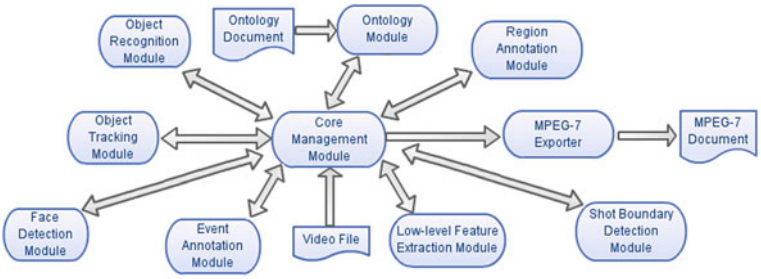


Fig. 1 Module interaction diagram of the annotation tool

Key frames, other video frames and regions from frames can be saved as separate files. Annotation tool consists of ten modules, Fig. 1 shows these modules and their interactions. Core management module takes annotator's request, sends necessary information to the related module, and after the process is finished and annotation results are obtained, these results sent back to core module and displayed to the annotator. The annotator may also interact with other modules, but modules interact with each other through the core module.

The annotation process starts with loading a video file. Then, shot boundary detection process is started. After detecting start and end frames for shots, these frames and I-frames between these frames are saved as key frames. The first frames of the shots and key frames are listed to the annotator. Once key frames are saved, SURF [8] features are extracted from these frames. At this point, annotator may load an OWL ontology that contains domain knowledge. This ontology file is parsed and displayed to the annotator in a hierarchical structure. Annotator can add new classes/instances or edit them from the ontology display. After that annotator can choose which annotations should be generated. If event annotation is chosen, then the event annotation window, shown in Fig. 2, is displayed. Annotator can define events and subevents from scratch or using defined events from the subclass of events class of ontology. Start and stop times of the events can be marked from event annotation. Also temporal relations between events can be defined. After annotations of events are finished, results are send back to the core module and events are listed in the main window. For facial annotations, face detection is included to the annotation tool. When annotator sends facial annotation request, face detection is executed for all key frames and potential facial regions marked and displayed to the annotator. Annotator can edit or delete these regions or can add new regions and mark them as face. These regions can be saved as a separate file and low-level features can be extracted from these regions. These features are important for applications like face recognition. If annotator needs to annotate still regions, the frame with the interest object is displayed in a separate window and annotator mark the object with its bounding box. Semantic information related to this object can be supplied as free text or by instances or concepts from the ontology. This annotated object can be searched through the all keyframes and with object recognition similar objects are

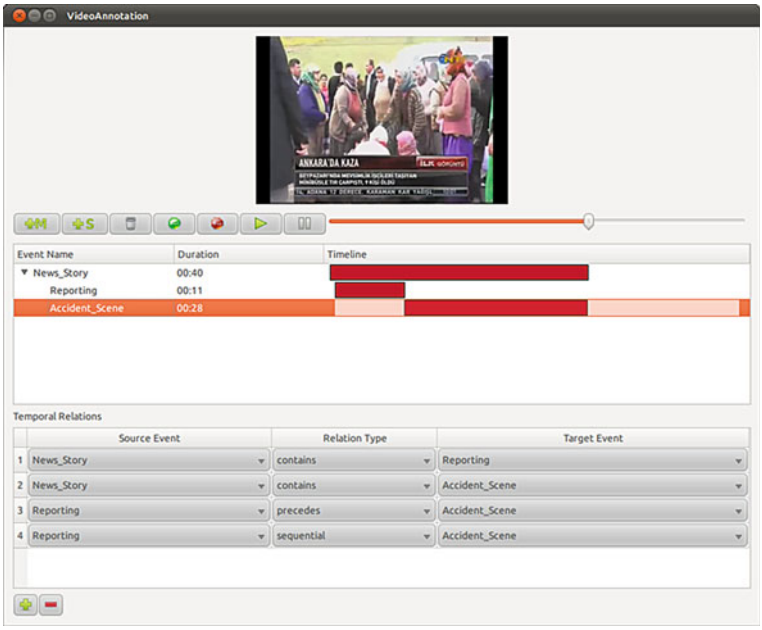


Fig. 2 Event annotation window

listed to the annotator. This property reduces annotation time. Like face detection, annotator can edit the results of object recognition. After annotator's verification, these results are saved as still regions and in the final document have links to the originally annotated object. Low-level features can be extracted from entire frames or still regions. These features are extracted with XM Reference Software [9]. Annotator can choose which features to be extracted. Low-level features are included in order to generate complete descriptions. Moving regions can be annotated from shots. Annotator selects the object from the key frame and with object tracking, the selected object is tracked through the shot and information about object motion is saved. After annotation process is finished, all information related to the annotations are sent to the exporter. Exporter processes this information and generates a single MPEG-7 document that is conformed to DAVP and lastly this document is saved to the file system. Following sections describe the modules that automize annotation process and how they reduce annotation time.

3.1 Shot Detection

Shot boundary detection is a fundamental step for video applications. Unlike scene and event detection, shot boundary detection can be done fully automatically. The annotation tool uses Edge Change Ratio [10] algorithm to detect shot

boundaries. Basically the algorithm calculates edge pixel ratios for consecutive frames. Calculated by the following equations:

$$\rho = max(\rho_{in}, \rho_{out}) \tag{1}$$

$$\rho_{out} = 1 - \frac{\sum E[x, y] \cdot \overline{E}'[x, y]}{\sum E[x, y]} \tag{2}$$

$$\rho_{in} = 1 - \frac{\sum \overline{E}[x, y] \cdot E'[x, y]}{\sum E[x, y]} \tag{3}$$

where E and E' are binary edge images of consecutive frames. $\overline{E}$ and $\overline{E'}$ are diamond shape dilated versions of these binary edge images. ρ_{out} denotes exiting edge pixels while ρ_{in} denotes entering edge pixels. In shots these ratios are relatively stable but at shot boundaries, they get high values and these abrupt changes indicate shot boundaries. The video is temporally decomposed with shot boundary detection and shots are indicated in the video description file. After shot boundaries are detected, key frames are extracted from these shots.

3.2 Object Tracking

Object tracking is used for moving region annotation. Once an object is selected, the application tracks it through the shot, and its position on different frames are obtained automatically. CAMSHIFT [11] algorithm is used for object tracking. This algorithm requires initial search window size and position, which is given by the annotator for moving region annotation. CAMSHIFT calculates a search window that has the same center with the initial search window. After that dominant mode is calculated and search window's center is updated, until it converges. The algorithm changes search window size dynamically to accurately locate the moving object, because moving object's size changes while it gets closer to the camera. CAMSHIFT algorithm is very fast, it can track object in real time. In order to automate this process further, without the selection of the moving object by the annotator, object detection module employs this approach: for still background a moving object can be easily detected by using motion vectors [12]. Since motion vectors are already encoded in the video, this approach is very fast compared to background modelling techniques. But motion vectors can be very noisy, for dynamic backgrounds, motion vectors do not give any useful information about the moving object.

3.3 Face Detection

Faces are always important for video annotation and if they are detected beforehand and presented to the annotator, this will reduce annotation time. The annotation tool employs widely used face detection method presented in [13]. The algorithm works in

real time. It uses Haar-like features and a modified version of AdaBoost algorithm to eliminate features and train classifiers. Since features are treated as weak classifiers, a large number of classifiers are obtained. With cascade of classifiers are built. It aims to reject nonfacial regions with computably cheap features and only make more calculations for promising regions. In order to achieve this, low level classifiers in the cascade are simple to compute and computationally more complex classifiers are at higher levels of the cascade. The cascade of classifiers highly reduces computation time yet preserves high detection rates. The annotation tool uses trained data from [14], which is obtained from a large dataset and gives good results as reported.

3.4 Object Recognition

Object recognition is used by the annotation tool to find similar objects to the annotated objects and link these semantic annotations to the recognized objects. In order to accomplish this task in a fast and effective way, annotation tool uses SURF features. After shot boundaries are detected and keyframes are extracted, SURF features are extracted from each key frame. When an object is annotated and similar objects are requested, SURF features of the annotated object extracted and SURF features of the object and frames are matched. For the number of matches that above a threshold, the matching object is located approximately and results are displayed to the annotator. Annotator can remove or edit the resulting objects. In order to use SURF descriptors, interest points of the image are detected. Then SURF descriptors are extracted from these points. These descriptors are scale and rotation invariant and have low dimensionality. Dimension of the feature vector is important since all key frames are searched for a single object. SURF provides a fast and robust method for object recognition.

4 Conclusion

A semi-automatic semantic annotation tool is presented. The annotation tool generates detailed MPEG-7 descriptions with annotator assistance. Annotation time is reduced by using image processing and pattern recognition techniques. Results are presented to the annotator and the annotator can edit or delete these results. Used MPEG-7 profile Detailed Audio Visual Profile (DAVP) ensures interoperability of the descriptions. Low-level feature extraction from frames and still regions, and relating these features to high-level concepts provide valuable information for content based retrieval applications. The annotation tool generates descriptions for visual data and it can be extended to operate on audiovisual data.

References

1. MPEG-7: ISO/IEC 15938. Multimedia Content Description Interface (2001)
2. Bailer, W., Schallauer, P.: The detailed audiovisual profile: enabling interoperability between MPEG-7 based systems. In: Proceedings of the 12th International MultiMedia Modelling Conference, pp. 217–224 (2006)
3. Dasiopoulou, S., Giannakidou, E., Litos, G., Malasioti, P., Kompatsiaris, Y.: A survey of semantic image and video annotation tools. In: Knowledge-Driven Multimedia Information Extraction and Ontology Evolution, pp. 196–239 (2011)
4. Heggland, J.: Ontolog: temporal annotation using ad hoc ontologies and application profiles. In: Research and Advanced Technology for Digital Libraries, pp. 5–17 (2002)
5. Lin, C., Tseng, B., Smith, J.: VideoAnnEx: IBM MPEG-7 annotation tool for multimedia indexing and concept learning. In: IEEE International Conference on Multimedia and Expo (2003)
6. Schallauer, P., Ober, S., Neuschmied, H.: Efficient semantic video annotation by object and shot re-detection. In: Posters and Demos Session, 2nd International Conference on Semantic and Digital Media Technologies (SAMT). Koblenz, Germany (2008)
7. Lowe D.: Object recognition from local scale-invariant features. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) The Proceedings of the Seventh International Conference on Computer Vision, vol. 2, pp. 1150–1157. IEEE, Los Alamitos (1999)
8. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Computer Vision-ECCV, pp. 404–417 (2006)
9. M. R. Group: Reference software. In: ISO/IEC JTC1/SC29 15938–6 (2003)
10. Zabih, R., Miller, J., Mai, K.: A feature-based algorithm for detecting and classifying production effects. Multimedia Syst. **7**(2), 119–128 (1999)
11. Bradski, G.: Computer vision face tracking for use in a perceptual user interface. Intel Technol. J. **Q2**, 1–15 (1998)
12. Zen, H., Hasegawa, T., Ozawa, S.: Moving object detection from MPEG coded picture. In: Proceedings of International Conference on Image Processing (ICIP), vol. 4, pp. 25–29. IEEE, Los Alamitos (1999)
13. Viola, P., Jones, M.: Robust real-time face detection. Int. J. Comput. Vis. **57**(2), 137–154 (2004)
14. Lienhart, R., Kuranov, A., Pisarevsky, V.: Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In: DAGM 25th Pattern Recognition, Symposium, pp. 297–304 (2003)

Space-Filling Curve for Image Dynamical Indexing

Giap Nguyen, Patrick Franco and Jean-Marc Ogier

Abstract In image retrieval, high-dimensional features lead often to good results, however, their uses in indexing and searching are time-consuming. The space-filling curve that reduces the number of dimensions to one while preserving the neighborhood relation can be used in this context. A new fast technique for image indexing is developed which enables rapid insertions of new images without changing existing data. The retrieving is accelerated by avoiding the distance computing because images are ordered on 1-D data structure. Hilbert curve, the most neighborhood preserving space-filling curve, is used in the experimentation. A proposal of fast mapping facilitates the computing of 1-D Hilbert indexes from high dimensional features.

1 Introduction

In content-based image retrieval, image is usually described by high-dimensional features, such as, histograms, SIFT features and whose number of dimensions could be very big, 128 for SIFT and attain thousands for histograms, and this may cause the time-consuming when we index or search images in a large image collection. In fact, the general origin of time-consuming is distance calculation in multi-dimensional space, the number of operations increases n^2 time when the number of dimension increases n time.

To overcome this problem, many research focus on the dimension reduction, such as feature selection, PCA or neural networks. However, these techniques themselves are time-consuming.

In this paper, the space-filling curve is used to map high-dimensional image features with one-dimensional indexes while preserving the locality of features.

G. Nguyen (✉) · P. Franco · R. Mullot · J.-M. Ogier
L3i Laboratory, University of La Rochelle, La Rochelle, France
e-mail: giap.nguyen@univ-lr.fr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_32,

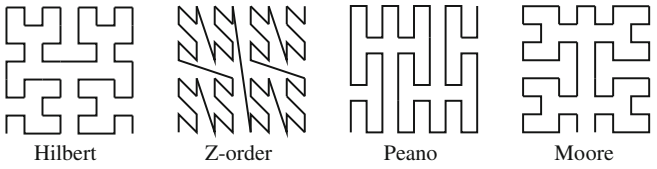


Fig. 1 Space-filling curves

Consequently, the features can be ordered linearly while the closeness on multi-dimensional space is preserved, i.e. similar images of an image are its previous and next images, thus, there is no distance calculation needed to find similar images. Unlike other dimension reduction techniques, index are invariable, the index of an image stay independent from other image indexes, hence, there is no change on existing images when new images are inserted and the time for insertion is linearly increasing because there is no supplementary indexing process.

Hilbert curve is used for its neighborhood preserving. However, the problem cannot be resolved by applying the original curves because it is a 2-D curve, a multi-dimensional extension is required. There is not an efficacy method, we propose an algorithm to map n-D features to 1-D indexes. This algorithm is then applied in a color image retrieval system that experiments the curve performance.

2 Space-Filling Curve

A space-filling curve [10] is a way to visit every point of a square, a cube or more generally, a n-D hypercube. Therefore, space-filling curve can be seen as a way to order the multi-dimensional data and to reduce the number of dimension to one. This 1-D feature, which is the position of point on curve, is called index. SFCs are self-similar: their subdivisions are similar to the whole curve.

The first space-filling curve is presented by Peano in 1989. After that, many others curve are presented, such as, Hilbert curve or Lebesgue curve. There are also alternative of these curves, for instance, Z-order curve,[1] Moore curve or Wunderlich curves [10]. Figure 1 shows the most famous space-filling curves.

The important property of the space-filling curve is locality-preserving: two close points are ordered nearly on curve. Thanks to this property, close points of a multi-dimensional point can be rapidly found by taking the points corresponding to previous indexes and next indexes of the input point index.

Many research try to measure the locality-preserving of space-filling curve which show that different curves preserve differently the locality. Note that the locality is not preserved totally because a 1-D index have only two neighbors while a n-D features have $2n$ neighbors (when von Neumann neighborhood is used). In almost of measure, the Hilbert curve is recognized to be the most locality-preserving curve

[1] This curve is based on the Lebegue curve and introduced by Morton [8].

[3, 4, 7]. That could explain why it is largely used in applications [5, 6, 9]. Thus, we use the Hilbert curve in the below experimentation.

3 Hilbert Curve: A Proposal of Algorithm for Fast High Dimensional Mapping

The original Hilbert curve is proposed in 2-D and its arbitrary dimension extension is not evident. There are only two effective mapping algorithms. Bially propose the first mapping technique, this is a table-based approach [1]. The drawback of this approach is the processing time and the memory occupation for the table, so it is not a good choice for high-dimensional mapping.

Butz algorithm [2] avoids this problem, there is no table used, the transformations are code by a series of binary operations. However, Butz proposes only the mapping from indexes (1-D) to their points (n-D) while in general, we need an inverse mapping, which does the dimension reduction.

There is no efficient n-D to 1-D mapping, applications are limited by low dimension. We propose below an algorithm to map n-D points to their indexes on curve in using steps of Butz algorithm. This approach works well in the high-dimensional case and inherits the rapidity of Butz algorithm.

Butz algorithm maps an index $\rho_1^1\rho_2^1\ldots\rho_n^1\rho_1^2\rho_2^2\ldots\rho_n^2\ldots\rho_1^m\rho_2^m\ldots\rho_n^m$ ($\rho_j^i \in \{0,1\}$) to a n-D point $(a_1, a_2, \ldots, a_n)$. The notation ρ^i is used to stand for $\rho_1^i\rho_2^i\ldots\rho_n^i$. The "principal position" of ρ^i is the last position in ρ^i such that $\rho_j^i \neq \rho_n^i$ (if all bits of ρ^i are equal, the principal position is the nth). The remaining entities necessary to define the algorithm are given in Table 1.

If α^i has the binary representation $\alpha_1^i\alpha_2^i\ldots\alpha_n^i$ then a_j has the binary representation $\alpha_j^1\alpha_j^2\ldots\alpha_j^m$. It is easy to recognize that (a_j)s are the rows of the matrix which has columns are (α_i)s.

Our proposal: *n-D to 1-D mapping* gets index of multi-dimensional point. With given point $(a_1, a_2, \ldots, a_n)$, we can easily get back the (α_i)s because they are columns of the matrix which has rows (a_j)s.

Table 2 shows two useful properties of the binary EXCLUSIVE-OR operation. From these properties and $\alpha^i = \omega^i \oplus \tilde{\sigma}^i$, we have: $\tilde{\sigma}^i = \alpha^i \oplus \omega^i$. The calculation of ρ_i is given in Table 3.

4 Application to Color Image Retrieving in Large Database

We apply now our mapping method in a image retrieval test. The color histogram is used as image feature. HSV (hue, lightness and saturation) color space, which is suitable for perception [11], is used. B-tree is used to order the images by their

Table 1 Entities used in Butz algorithm

J_i: An integer between 1 and n equal to the subscript of the principal position of ρ^i. I the following four examples of ρ^i for the case $n = 5$, the values of J_i are 5, 2, 4, and 5, respectively (the principal positions are circled):

$$\begin{array}{ccccc} 1 & 1 & 1 & 1 & \textcircled{1} \\ 1 & \textcircled{0} & 1 & 1 & 1 \\ 0 & 0 & 1 & \textcircled{1} & 0 \\ 0 & 0 & 0 & 0 & \textcircled{0} \end{array}$$

σ^i: A byte of n bits, such that $\sigma_1^i = \rho_1^i, \sigma_2^i = \rho_2^i \oplus \rho_1^i, \sigma_3^i = \rho_3^i \oplus \rho_2^i, ..., \sigma_n^i = \rho_n^i \oplus \rho_{n-1}^i$, where $\oplus$ stands for EXCLUSIVE-OR operation.

τ^i: A byte of n bits obtained by complementing σ^i in the nth position and then, if and only if the resulting byte is of odd parity, complementing in the principal position. Hence, τ^i is always of even parity. Note that the parity of σ^i is given by the bit ρ_n^i and that a mask for performing the second complementation may be set up in the same process which calculates J_i.

$\tilde{\sigma}^i$: A byte of n bits obtained by shifting σ^i right circular a number of positions equal to

$$(J_1 - 1) + (J_2 - 1) + ... + (J_{i-1} - 1)$$

There is no shift in σ^1

$\tilde{\tau}^i$: A byte of n bits obtained by shifting τ^i in exactly the same way.

ω^i: A byte of n bits where

$$\omega^i = \omega^{i-1} \oplus \tilde{\tau}^{i-1}, \quad \omega^1 = 00...00$$

and where $\oplus$ indicates the EXCLUSIVE-OR operation on corresponding bits.

α^i: A byte of n bits where $\alpha^i = \omega^i \oplus \tilde{\sigma}^i$.

Table 2 Some properties of EXCLUSIVE-OR

Definition			Properties
$\oplus$	0	1	$x \oplus y = y \oplus x$
0	0	1	if $x \oplus y = z$ then $x \oplus z = y$
1	1	0	

Hilbert indexes. This famous data structure minimizes the index search time and makes clear the space-filling curve performance.

4.1 Image Retrieving Procedure

In the following experimentation, the image retrieving include two phases:

Database constructing Color histogram of each image is computed and mapped with its Hilbert index. Images are then incrementally added to a B-tree where increasing order of indexes is maintained.

Table 3 Inversion of Butz algorithm

σ^1	: $\sigma^1 = \alpha^1$ because:
	$\alpha^1 = \omega^1 \oplus \tilde{\sigma}^1 \Rightarrow \tilde{\sigma}^1 = \alpha^1 \oplus \omega^1$
	$\omega^1 = 00\ldots00 \Rightarrow \tilde{\sigma}^1 = \alpha^1$
	There is no shift in $\sigma^1 \Rightarrow \sigma^1 = \tilde{\sigma}^1$
ρ^i	: A byte of n bits, such that $\rho^i_1 = \sigma^i_1, \rho^i_2 = \sigma^i_2 \oplus \rho^i_1, \rho^i_3 = \sigma^i_3 \oplus \rho^i_2, \ldots, \rho^i_n = \sigma^i_n \oplus \rho^i_{n-1}$
J_i	: see Table 1
τ^i	: see Table 1
$\tilde{\tau}^i$	: see Table 1
ω^{i+1}	: see Table 1
$\tilde{\sigma}^{i+1}$	: A byte of n bits where $\tilde{\sigma}^{i+1} = \alpha^{i+1} \oplus \omega^{i+1}$
σ^{i+1}	: A byte of n bits obtained by shifting σ^i left circular a number of positions equal to $(J_1 - 1) + (J_2 - 1) + \cdots + (J_i - 1)$

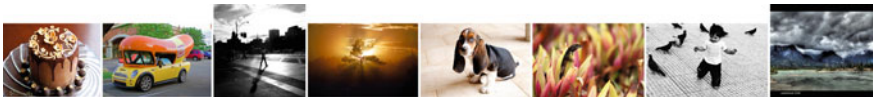


Fig. 2 Some images of the MIRFLICKR-25,000 collection

Image searching is then resumed by index searching in the before-said B-tree, the request index corresponding to the given input image. Images having closest indexes are issued as result.

Because the Hilbert curve preserves neighborhoods, images ordered successively have much probability to be similar and the images having indexes close to input images index have also much probability to be similar to input images. Thereto, we can find out images very fast since they are kept in a B-tree (Fig. 2).

4.2 Experimentation Condition

We use the MIRFLICKR-25,000 collection (http://press.liacs.nl/mirflickr/) as image database. This standard collection contains 25,000 color images under the Creative Commons license collected from the famous photo archive http://www.flickr.com/. The collection is offered with its annotations which assign images with 38 different categories, such as animal, baby, car, sky. Image size is normalized such that the larger dimension is 500 pixels long. The following images resume test database:

Test images are separated in 3 partitions for 3 purposes: 20,000 first images are used in database building test, dynamical insertion test uses 4,000 others images. 1,000 left images are used in the search test.

To computed the color histogram, HSV color must be quantized into bins. The quantization scheme in [11] is used in our experimentation with 13 hues, 3 saturations,

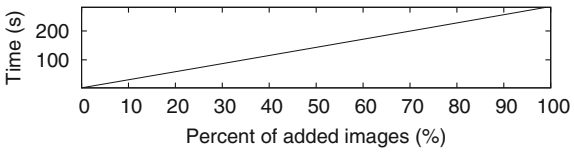


Fig. 3 Cumulative adding time

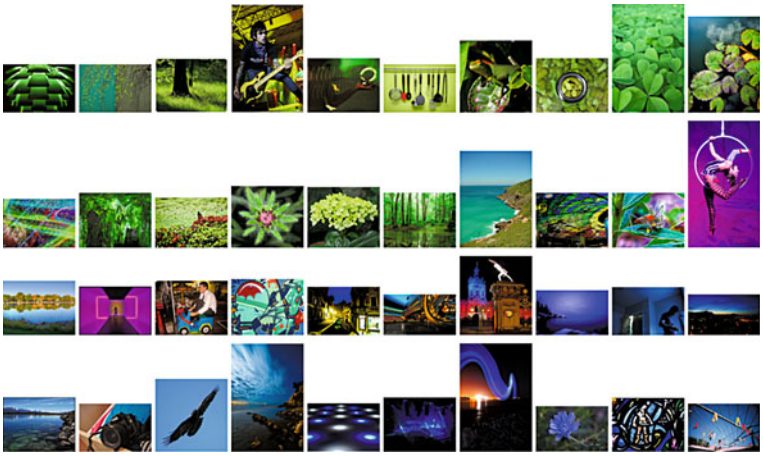


Fig. 4 Image ordering

3 values and 4 grays. These parameters give 121 colors corresponding to 121 bins histogram used as 121-dimensional images feature.

The tests is realized on a standard computer:

Processor	Intel Core 2 Duo T9800 2.93 GHz × 2
Memory	3.9 GB
OS	Ubuntu 11.10 64-bit

4.3 Image Indexing

Color histogram of each image (from 20,000 images) is computed and mapped with its Hilbert index. The image information (name, histogram and index) is then inserted into the B-tree (Fig. 3).

Image ordering We can recognize that similar images are usually ordered closely. For example, Fig. 4 shows images having order from 2501 to 2540. The head images

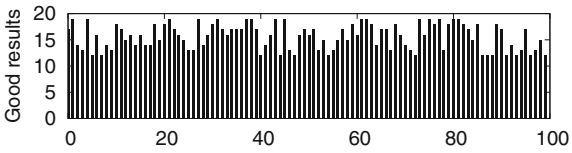


Fig. 5 Search accuracy

have greener color while the tail images are bluer but the change is little by little. This reflects the relation between the colors and the indexes: successive indexes usually correspond to similar colors. The average size of image series containing similar color is 3.62.

Fast indexing The database building time, including time for histogram computing, mapping them with Hilbert indexes and insertion into the B-tree, on 20,000 images is 282.18 s (0.0141 s/image). The cumulative adding time (via percentage of images added) is shown in the following graph:

We can see a regular linear increase of added time, there is no abnormal change.

Dynamical indexing The databases are dynamics, we can add new images without modifying existing entities. The time for adding 4,000 new images into database is only 57.83 s. The average adding time for each image is very small (0.0145 s). Therefore, the database updating is simple (Fig. 5).

4.4 Image Searching

For each image in 1,000 test images, the histogram are mapped with the its index, the search of this index will issue 20 closest images. For example, Fig. 6 shows search results of 3 inputs.

Besides many good results, we can recognize some results which haven't the same colors as the input. The inaccuracy can be explained by the lack of similar images but it is also from the neighbor missing of the curve. The original space, which of histogram, is multi-dimensional but the points on Hilbert curve are only one dimensional. The map can be considered as a dimensional reduction which causes inevitably the information missing, in this case, it is the neighbor missing. In our test, the average of number of good results are 15.68. In more details, the following graph shows the accuracy over 100 first test searches (Fig. 7).

In return for the neighbor missing, the images can be increasingly ordered by Hilbert indexes, therefore, the search is much accelerated. In our test, the search time is only $O(\log_N(n))$ where n is database size and N is capacity of B-tree node. In our test, the total search time for 1,000 search tests is 14.82 s and the average search time is 0.014 s/image. The following graph shows search time details.

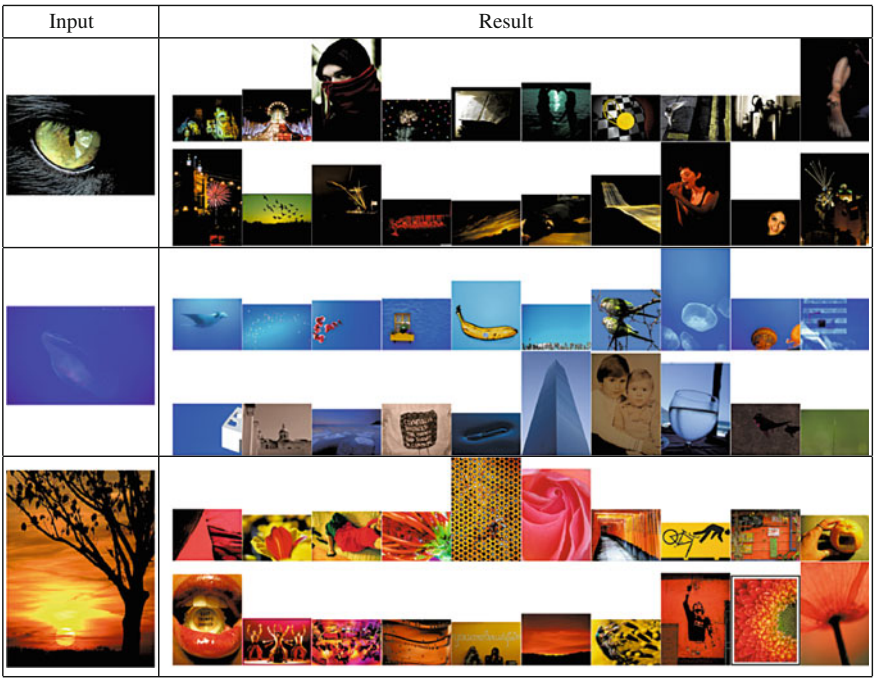


Fig. 6 Search results

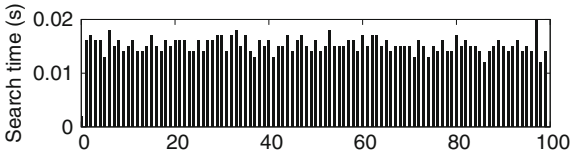


Fig. 7 Search time

5 Conclusion

In images processing applications, multi-dimensional features are frequently used and spatial relation between these features decides the data essence. Space-filling curve fits well with these applications by reducing the feature dimension to 1 and preserving the spatial relation of features. The use of space-filling curve in multi-dimensional feature applications eliminates the distance calculation time and reduces the search time once features are linearly ordered. Besides, organization of images is dynamical, images are independent, there is no computation on existing images when new images are inserted while in usual image retrieval system, a new training process is needed.

The new fast and light on memory usage n-D to 1-D mapping can be efficiently use in applications which need of multi-dimensional feature distance calculation, including application which limited by low-dimension space-filling curves. The integration of this mechanism in a high accurate system, like CBIRs, could be interesting.

Our test application shows the good performance of Hilbert curve in image ordering and search time. Of course, in a preliminary test, the search results are far from a high accuracy image search system. This simple application suggest a complete automatic image organization—instant search system, which improves the found results, for example, by combining and refining the results. Application of space-filling curve is not limited to global image features, we can also apply space-filling curves for local features.

References

1. Bially, T.: Space-filling curves: their generation and their application to bandwidth reduction. IEEE Trans. Inf. Theory **15**(6), 658–664 (1969)
2. Butz, A.: Alternative algorithm for Hilbert's space-filling curve. IEEE Trans. Comput. **20**(4), 424–426 (1971)
3. Faloutsos, C., Roseman, S.: Fractals for secondary key retrieval. In: Proceedings of the Eighth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS '89, pp 247–252. ACM, New York (1989)
4. Gotsman, C., Lindenbaum, M.: On the metric properties of discrete space-filling curves. IEEE Trans. Image Process. **5**(5), 794–797 (1996)
5. Lawder, J., King, P.: Using space-filling curves for multi-dimensional indexing. Adv. Databases **1832**, 20–35 (2000)
6. Liang, J., Chen, C., Huang, C., Liu, L.: Lossless compression of medical images using Hilbert space-filling curves abstract. Comput. Med. Imaging Graph. **32**(3), 174–182 (2008)
7. Moon, B., Jagadish, H., Faloutsos, C., Saltz, J.: Analysis of the clustering properties of the Hilbert space-filling curve. IEEE Trans. Knowl. Data Eng. **13**(1), 124–141 (2001)
8. Morton, G.: A Computer Oriented Geodetic Data Base and a New Technique in File Sequencing. IBM, Ottawa (1966)
9. Muelder, C., Ma, K.: Rapid graph layout using space filling curves. IEEE Trans. Visual. Comput. Graph. **14**, 1301–1308 (2008)
10. Sagan, H.: Space-Filling Curves, vol. 2. Springer, New York (1994)
11. Smith, J., Chang, S.: Tools and techniques for color image retrieval. Storage Retr. Image Video Databases IV **2670**, 426–437 (1996)

Person Independent Facial Expression Recognition Using 3D Facial Feature Positions

Kamil Yurtkan and Hasan Demirel

Abstract Facial expressions contain a lot of information about the feelings of a human. They play an important role in human–computer interaction. In this paper, we propose a person independent facial expression recognition algorithm based on 3-Dimensional (3D) geometrical facial feature positions to classify the six basic expressions of the face: Anger, disgust, fear, happiness, sadness and surprise. The algorithm is tested on BU-3DFE database and provides encouraging recognition rates.

Keywords Facial expression analysis · Facial expression recognition · Facial feature selection · Face biometrics

1 Introduction

Recent improvements in computer graphics and image processing fields made human–computer interaction applicable. Human face contains most of the information about the feelings of a human and human–computer interaction highly depends on facial analysis.

Ekman [2] is one of the most important researchers in the earlier studies of facial expressions in 1970s. Ekman's studies were about the classification of the human facial expressions in seven basic classes: Anger, disgust, fear, happiness, sadness,

K. Yurtkan (✉)
Computer Engineering Department, Cyprus International University,
Mersin 10, Lefkosa, Turkey
e-mail: kamil.yurtkan@ciu.edu.tr; kyurtkan@ciu.edu.tr

H. Demirel
Electrical and Electronic Engineering Department,
Eastern Mediterranean University, Mersin 10, Gazimağusa, Turkey
e-mail: hasan.demirel@emu.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_33,

surprise and neutral. Then, Ekman and Friesen have proposed the Facial Action Coding System (FACS) to code the facial expressions with the action units defining the facial movements [3]. In 1999, MPEG-4 standard introduced a neutral face model including 83 feature points. MPEG-4 standard also defined 68 Facial Animation Parameters (FAPs) used to animate the face by the movements of the feature points. MPEG-4 FAPs are still used in most of the research labs in facial expression synthesis and analysis studies [1].

Current research studies in facial expression recognition systems focused on automatic facial expression recognition and achieved acceptable recognition rates. Unfortunately, most of the systems developed have several limitations such as pose, lighting, resolution and orientation of the face images. Therefore, automatic recognition of facial expressions is still under study by the researchers.

Facial expression recognition process depends on facial features used to define expressions. Hence, selection of the discriminative features determines the overall classification performance. Several facial features can be extracted from face images depending on the classification problem. We consider 83 3D geometrical facial feature positions on the face defined in MPEG-4 Facial Definition Parameters (FDPs) in our recognition system.

In this study, we focus on the problem of person independent facial expression recognition. Our approach is based on 3D geometrical facial feature positions on the face. We develop a Support Vector Machine (SVM) classifier system to recognize facial expressions with the input geometrical feature positions.

The paper proposes a novel algorithm for the person independent recognition of six basic facial expressions and feature selection procedure for expression classification. The system details are given in Sect. 2 about the classifier system. Then, Sect. 3 describes feature selection process for expression classification problem. The performance of the proposed system is reported in Sect. 4 on BU-3DFE database [8].

2 Support Vector Machine Classifier System

Facial expression recognition is considered as a classification problem. We define each face as a row vector of geometrical feature point positions. Our approach is based on Support Vector Machine (SVM) classifiers. System considers geometrical facial feature point locations in 3D as input. The facial feature points are selected from the MPEG-4 feature set. The feature selection procedure is described in Sect. 3.

System includes 15 SVM classifiers in total. Each classifier employs a kernel function that maps the training data into kernel space. Linear function (or dot product) is selected as the kernel function for all the classifiers.

Classifiers are designed for two classes including all the combinations of six expression classes. The classifiers with expression couples are anger-disgust, anger-fear, anger-happiness, anger-sadness, anger-surprise, disgust-fear, disgust-happiness, disgust-sadness, disgust-surprise, fear-happiness, fear-sadness, fear-surprise,

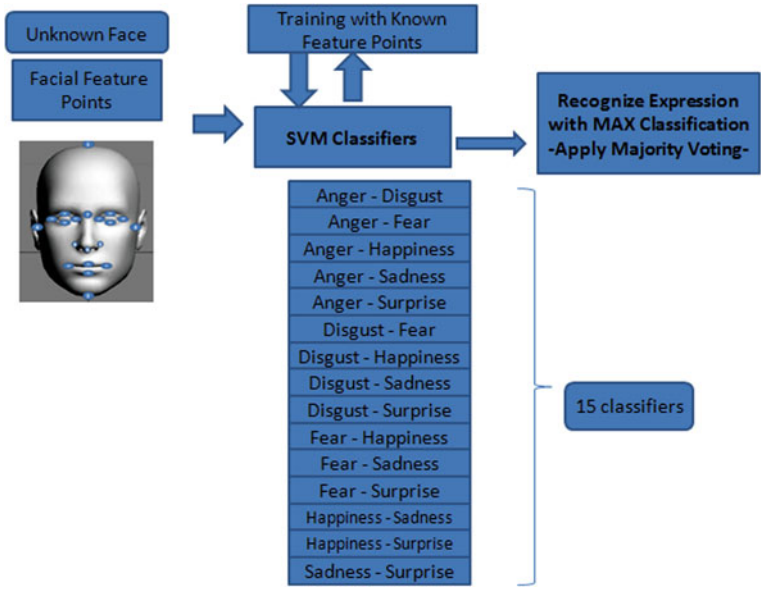


Fig. 1 SVM classifier system proposed for facial expression recognition

happiness-sadness, happiness-surprise and sadness-surprise. System has 15 classifiers in total for all the combinations of expression couples.

Each classifier is trained separately and does not affect each other. 85 % of the faces is reserved for the training of classifiers, and 15 % is used in the testing part. The details about the training and testing phases of the classifiers are described in Sect. 4 by reporting the performances on BU-3DFE database [8]. After classifying each feature set in 15 classifiers, we employ majority voting among six expressions. The expression with the maximum number of classifications is selected as the recognized expression. The SVM classifier system is illustrated in Fig. 1.

The input vector consists of 3D facial feature positions describing a face. All selected facial feature positions are arranged as a row vector for each face. Consider a row vector definition of a facial feature point as shown in Eq. 1, V_m being a 3D vector definition for a facial feature point.

$$V_{m,i} = \left[V_{m,i_x} \quad V_{m,i_y} \quad V_{m,i_z} \right] \tag{1}$$

Each facial feature position is appended as a row vector FV shown in Eq. 2 where k stands for number of facial feature points selected to represent a face. After combining all row vectors for all faces, a matrix FM is formed as shown in Eq. 3. The number of face vectors is represented by n.

$$FV_i = \left[V_{m,1} \quad V_{m,2} \quad V_{m,3} \ldots V_{m,k} \right] \tag{2}$$

We use 100 samples from BU-3DFE database with 6 different expressions resulting in 600 face vectors in total to construct matrix *FM*. BU-3DFE database includes 4 intensity levels of expressions. We have used maximum intensity level, which is level 4, in our tests. The training and test sets are derived from the subdivision of the matrix *FM* into two subsets.

$$FVM = \begin{bmatrix} FV_1 \\ FV_2 \\ \cdots\cdots \\ FV_n \end{bmatrix} \tag{3}$$

3 Feature Selection for Expression Classification

Accurate description of a face is important to represent the facial characteristics of the expressions to solve the expression classification problem. MPEG-4 defines Facial Definition Parameters (FDPs) and Facial Animation Parameters (FAPs) on a generic face model used to represent facial expressions and animations [1]. Facial expressions can be modelled with deformations on the neutral face by using MPEG-4 FAPs [1]. Therefore, we consider the MPEG-4 FDP set for the representation of a face with geometrical feature positions.

Our first experiments show that MPEG-4 FDP set defines facial expression deformations on the face well and achieves acceptable recognition rates as presented in Table 1.

FAPs represent a complete set of basic facial actions including head motion, tongue, eye and mouth control. Our previous studies about facial expression modelling and synthesis [9, 10] showed that the action of the FAPs on the whole 3D generic face model is obtained by displacement functions for each vertex. The displacement functions $\Delta i\ (x, y, z)$ of a vertex are computed according to the position of the vertex in the influence area, to the intensity of the FAP in the related feature point(s) and an additional weight for design issues as shown in Eq. 4 [1].

$$\begin{bmatrix} \Delta i_x \\ \Delta i_y \\ \Delta i_z \end{bmatrix} = \begin{bmatrix} Wi_x \\ Wi_y \\ Wi_z \end{bmatrix} * W'j * FAP_{x,y,z} \tag{4}$$

The weight *Wi* is based on the distance of the vertex from the feature point and the weight spreads decreasingly from the centre of the influence area. If we want that vertex not to be affected by the FAP, $W'j = 0$ may be chosen.

It is seen from MPEG-4 FAP implementation that although most of the FDPs are affected among 6 basic expressions, some FDPs are weakly influenced from facial expression deformations. Therefore, we select some subset of facial feature points

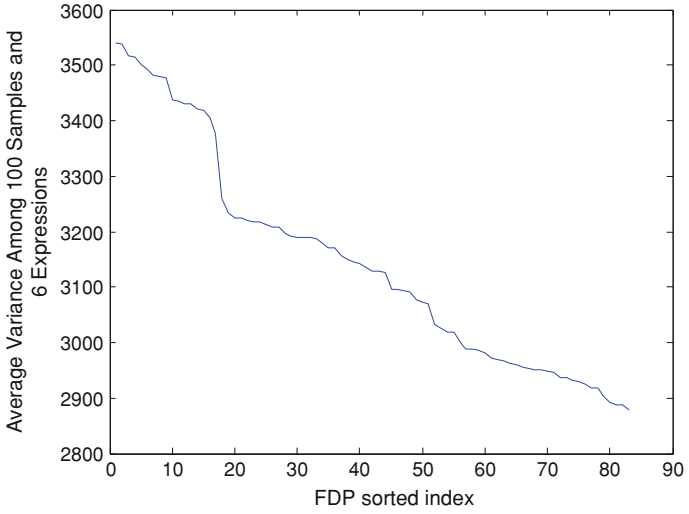


Fig. 2 Variance analysis of MPEG-4 FDPs for 100 samples among neutral face and 6 basic expressions

from MPEG-4 FDP set which are highly affected by expression deformations and include most of the information.

To select the facial feature points highly affected by facial expression deformations, we measured the variances of each feature point in 3D for the 6 basic expressions and a neutral face, and analyzed the overall variances of the MPEG-4 FDP set as shown in Eq. 5, where X is a 3D feature point position and μ is the mean value for the related feature point among 6 basic expressions and neutral face. For 85 samples selected for training in BU-3DFE database, we measure the variances of each FDP among all facial expressions and neutral face. Then, the average variances are calculated for each FDP.

Figure 2 illustrates variance analysis graph of MPEG-4 facial features on our training set. It is seen from Fig. 2 that after sorting the variances in descending order, there are feature points having higher variances according to others when face is deformed from neutral to anyone of the expressions. Thus, we eliminate low variance feature points in face definition vectors for our recognition tests and formed *FM* matrix shown in Eq. 3 accordingly. We sort feature point variances in descending order and obtain breaking points from sudden decreases in the variance. These breaking points are after first 9, 17, 27, 36, 44, 48, 55 and 71 feature points. Then, a brute force search is applied to find the facial features selected according to breaking points which maximizes recognition rate. Thus, we employ first 71 feature points with high variance from MPEG-4 FDPs because of having a clear decrease in variance value and maximizing recognition rate.

$$Var(x) = E[(X - \mu)^2] \tag{5}$$

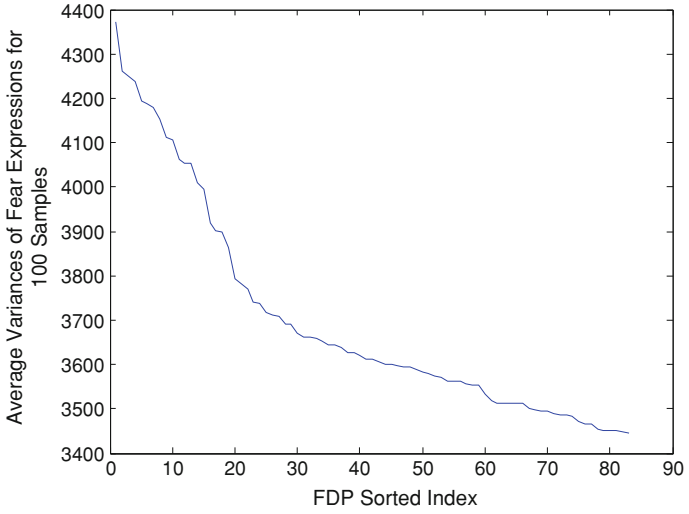


Fig. 3 Variance analysis of MPEG-4 FDPs for 100 samples among neutral face and fear expression

Our performance results showed that the features selected improved recognition performance of basic facial expressions compared to all MPEG-4 FDP set. However, fear expression is recognized with average recognition rates with both MPEG-4 FDP set and our proposed selected feature point set. Hence, we consider fear expression separately in our study and perform variance analysis specifically for fear expression. The variance analysis is shown in Fig. 3. Following the same procedure for feature selection, we select 37 feature points from the MPEG-4 set with higher variance among fear expressions of 85 training samples and perform experiments accordingly for fear expression test sets.

4 Performance Analysis

The performance of proposed facial expression recognition system is measured on BU-3DFE database [8] and recognition rates are reported among six basic facial expressions which are anger, disgust, fear, happiness, sadness and surprise.

FM matrix is constructed as described in Sect. 2 including 100 samples with 6 basic expressions. In total, matrix includes 600 row vectors describing 600 faces. We have selected maximum intensity expressions from available 4 levels of intensities given in BU-3DFE database.

15 2-class SVM classifiers are trained separately with 85 % of the row vectors of *FM* matrix and 15 % of row vectors are used in testing. As described in Sect. 2, we classify expressions using 15 2-class SVM classifiers and apply majority voting

Table 1 Recognition rates for proposed SVM classifier using 83 FDPs

Expression	Recognition rate (%)
Anger	93.3
Disgust	86.7
Fear	60
Happiness	93.3
Sadness	80
Surprise	86.7
Overall	**83.33**

Table 2 Recognition rates using 71 selected FDPs with (2nd column) and without (1st column) fear expression specific recognition using 37 selected FDPs

Expression/ selected features	Recognition rates (%)	
	71 Features	71 Features combined with fear specific 37 features' performance
Anger	100	100
Disgust	86.7	86.7
Fear	60	86.7
Happiness	93.3	93.3
Sadness	80	80
Surprise	93.3	93.3
Overall	**85.55**	**90**

among 6 expression classes. Table 1 presents recognition rates of basic expressions using proposed system with 83 MPEG-4 FDPs. In Table 2, improvements in the recognition rates after applying our proposed feature selection procedure can be observed. Overall recognition rate is improved to 85.55 %.

It is seen from Tables 1 and 2 that fear expression has average recognition rate around 60 % which is not acceptable in expression recognition systems. The main reason for this average recognition rate is that fear and sadness expressions are highly related and their geometric feature positions are close to each other. In order to improve recognition of fear expressions, we employ a separate feature selection procedure specific to fear expression as described in Sect. 3. The variances of 83 MPEG-4 FDPs among neutral and fear expressions of 85 training samples are measured. The corresponding variance analysis is illustrated in Fig. 3. Together with specific feature selection procedure to fear expression, our proposed system achieves 90 % overall recognition rates.

Currently, we are working on separate feature selection procedures to identify different selections of feature points for the recognition of each expression separately.

The comparison of the proposed system with the current systems in the literature tested on BU-3DFE database is given in Table 3. The other algorithms we compare employs 90 % training samples, whereas our approach employs 85 % training samples from the same database. We see from Table 3 that our proposed system achieves

Table 3 Performance comparison of proposed recognition systems

Expression	Recognition rates (%)				
	Soyel et al. [5]	Wang et al. [7]	Mpiperis et al. [4]	Tang et al. [6]	Proposed
Anger	85.9	80	83.6	86.7	100.0
Disgust	87.4	80.4	100.0	84.2	86.7
Fear	85.3	75.0	97.9	74.2	86.7
Happiness	93.5	95.0	99.2	95.8	93.3
Sadness	82.9	80.4	62.4	82.5	80
Surprise	94.7	90.8	100.0	99.2	93.3
Overall	**88.3**	**83.6**	**90.5**	**87.1**	**90**

competitive recognition rates compared to the current systems in the literature and open for further improvements.

5 Conclusion

In this paper, a person independent facial expression recognition system is proposed based on 2-class SVM classifiers and 3D geometric facial feature positions. A novel feature selection procedure is proposed to improve recognition performance of the system. Performance results show that our proposed system achieves encouraging recognition rates and it is open to further improvements considering expression specific feature selection procedures.

References

1. Abrantes, G., Pereira, F.: MPEG-4 facial animation technology: survey, implementation and results. IEEE Trans. Circ. Syst. Video Technol. **9**(2), 290–305 (1999)
2. Ekman, P., Friesen, W.: Pictures of Facial Affect. Consulting Psychologist Press, Palo Alto (1976)
3. Ekman, P., Friesen, W.: The Facial Action Coding System: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press, San Francisco (1978)
4. Mpiperis, I., Malassiotis, S., Strintzis, M.G.: Bilinear models for 3D face and facial expression recognition. IEEE Trans. Inf. Forensics Secur. **3**, 498–511 (2008)
5. Soyel, H., Tekguc, U., Demirel, H.: Application of NSGA-II to feature selection for facial expression recognition. Comput. Electr. Eng. (Elsevier) **37**(6), 1232–1240 (2011)
6. Tang, H., Huang, T.S.: 3D facial expression recognition based on properties of line segments connecting facial feature points. IEEE International Conference on Automatic Face and Gesture Recognition (2008)
7. Wang, J., Yin, L., Wei, X., Sun, Y.: 3D facial expression recognition based on primitive surface feature distribution. Comput. Vis. Pattern Recognit. **2**, 1399–1406 (2006)

8. Yin, L., Wei, X., Sun, Y., Wang, J., Rosato, M.: A 3d facial expression database for facial behavior research. In: Proceedings of International Conference on FGR, pp. 211–216, Southampton (2006)
9. Yurtkan, K., Demirel, H.: Facial expression synthesis from single frontal face image. In: Proceedigns of the 6th International Symposium on Electrical and Computer Systems, Eueropean University of Lefke, Gemikonagi, TRNC, 25–26 November 2010
10. Yurtkan, K., Soyel, H., Demirel, H., Özkaramanli, H., Uyguroglu, M., Varoglu, E.: Face modeling and adaptive texture mapping for model based video coding. LNCS **3691**, 498–505 (2005)

Part IX
Network Science II

Performance Evaluation of Different CRL Distribution Schemes Embedded in WMN Authentication

Ahmet Onur Durahim, İsmail Fatih Yıldırım, Erkay Savaş and Albert Levi

Abstract Wireless Mesh Networks (WMNs) have emerged as a promising technology to provide low cost and scalable solutions for high speed Internet access and additional services. In hybrid WMNs, where mesh clients also act as relaying agents and form a mesh client network, it is important to provide users with an efficient anonymous and accountable authentication scheme. Accountability is required for the malicious users that are to be identified and revoked from the network access and related services. Promising revocation schemes are based on Certification Revocation Lists (CRLs). Since in hybrid WMNs mesh clients also authenticate other clients, distribution of these CRLs is an important task. In this paper, we propose and examine the performance of different distribution schemes of CRLs and analyze authentication performance in two scenarios: in one scenario all mesh routers and mesh clients obtain CRLs and in the second one, CRLs are held only by the mesh routers and mesh clients acting as relaying agents require CRL checking to be performed from the router in authenticating another client.

Erkay Savaş, and Ahmet Onur Durahim are supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under Project number 105E089. Albert Levi and İsmail Fatih Yıldırım are supported by Turk Telekom under Grant Number 3014-01.

A. O. Durahim · İ. F. Yıldırım · E. Savaş · A. Levi (✉)
Sabanci University, Istanbul, Turkey
e-mail: levi@sabanciuniv.edu

A. O. Durahim
e-mail: durahim@sabanciuniv.edu

İ. F. Yıldırım
e-mail: ismailfatih@sabanciuniv.edu

E. Savaş
e-mail: erkays@sabanciuniv.edu

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_34,

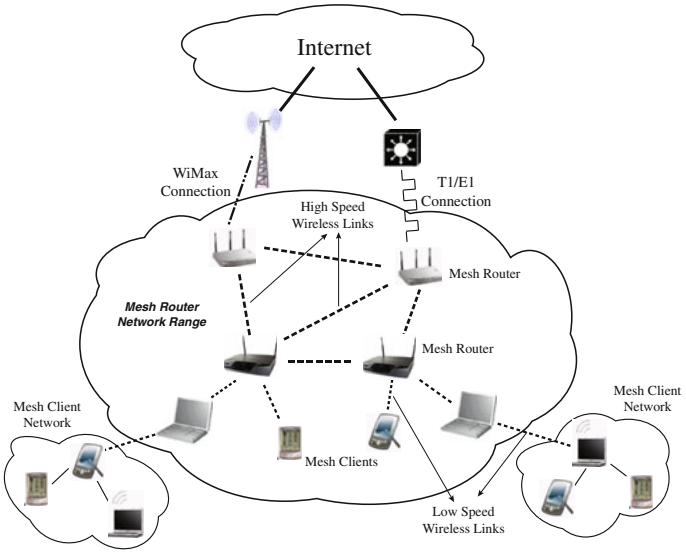


Fig. 1 Hybrid WMN architecture

1 Introduction

Recently, using mobile devices and wireless networks become a convenient and inexpensive way to connect to Internet. In this respect, hybrid WMNs are proposed as a solution where mesh clients and routers collaboratively form a well-connected network. Generally, WMNs are comprised of mesh routers and mesh clients (network users), whereby mesh routers are in charge of providing coverage and routing services for mesh clients which connect to the network using laptops, PDAs, smartphones, etc. Hybrid architectures [1] (*cf.* Fig. 1) are the most popular since in addition to mesh routers, mesh users may also perform routing and configuration functionalities for other users to help improve the connectivity and coverage of the network.Ubiquity and invasiveness of WMNs, however, pose serious challenges for security and privacy of individuals who cherish their benefits. Being connected via a smart mobile device may necessitate entrusting one's privacy to some—not necessarily trustworthy—third parties to varying extents. In many cases, privacy is simply ignored. As in the case of security, initial authentication of a user to the network is a key point for privacy protection. On the other hand, uncontrolled anonymity encourages some users with ill intentions to act maliciously, since they would not be identified or tracked due to their anonymous access to the network.

Therefore, anonymous authentication frameworks to be proposed for the hybrid WMNs should both satisfy necessary privacy and accountability requirements at the same time. Revocation mechanisms play a crucial role in providing accountability by identifying and revoking a malicious user. Most promising revocation mechanisms

are based on Certification Revocation Lists (CRLs), where an identifier of a network user is added to the list in order to prevent a revoked user from future access to the network. Thus, required check on deciding whether an authenticating user is revoked or legitimate, can only be performed by the entity who holds the CRL. Furthermore, this check must be accomplished with an up-to-date list. So, it is important to determine where to keep the CRLs and how to update them and where to perform the revocation check.

There are two alternative CRL distribution solutions are proposed and examined in this paper: First, CRLs are held both by mesh routers and mesh clients acting as relaying agents. In the second alternative, CRLs are held only by the mesh routers and revocation check is performed by the mesh routers on behalf of the relaying agents authenticating an another mesh clients.

In order to examine these alternatives, A^2-MAKE framework [2] is chosen as a base authentication platform where users can connect to the network in an anonymous and accountable manner and revocation mechanism in A^2-MAKE is based on the CRLs.[1]

2 A^2-MAKE

A^2-MAKE framework is a collection of protocols that provides anonymous mutual authentication to its users whereby legitimate users can connect to network from anywhere without being identified or tracked unwillingly. No single party (or authority, network operator, etc.) can violate the privacy of a user. User accountability is implemented via user identification and revocation protocols where revocation is performed using CRLs.

In order to connect to the network in A^2-MAKE, network users authenticate themselves to the mesh routers if there is one in communication range. Otherwise, they are connected to the network by mesh clients acting as relaying agents if they find one in their communication range. If the authentication is performed by the mesh routers, routers provide their authentication payload using conventional digital signature algorithms since routers does not require privacy protection. On the other hand, relaying agents who are also mesh clients that require privacy protection provide authentication payload using anonymous authentication scheme. In both connection attempts, authenticating mesh client performs anonymous authentication procedures.

In order to provide accountability, user identification and revocation procedures are proposed, whereby an identifier is added to the UserRL to revoke a user. So, authenticating agent checks this list in order to determine whether a network user is a legitimate or a revoked one.

[1] This list is named as UserRL in A^2-MAKE.

3 CRL Distribution Scenarios

We propose two different CRL distribution scenarios, based on where the list is held which are implemented over A^2-MAKE framework.

In the first scenario, it is assumed that CRL is held by mesh clients in addition to the mesh routers. Therefore mesh clients can perform revocation list check by themselves with the CRL obtained from the router it is connected when the updated list is broadcast by the Network Operator to the network through mesh routers. Important problem to be considered here is the possible use of obsolete CRL by the mesh clients acting as relaying agents in revocation list checking.

On the other hand, in the second scenario, CRL is only held by the mesh routers. A relaying mesh client asks the router it is connected, to perform UserRL checking for another client which she assists to connect to the network. As a result, all revocation list checkings are made by the mesh routers with the up-to-date CRL.

In both of these scenarios, it is important to examine the authentication times and the number of successful connections made. In the first scenario, differing from the second one, analysis of the number of true positive authentications made by the relaying mesh clients is required. True positive authentication is the ratio of the number of authentications accomplished by the relaying mesh clients with the up-to-date CRL to the total successful authentications made by her throughout the lifetime of the network including the authentications made with obsolete CRL.

4 Performance of Two Different CRL Distribution Scenarios

In order to evaluate the performance of different CRL distribution schemes, we conducted experiments on ns-3 (version 13) [3], on Ubuntu 10.04 platform.

In all our simulations, the simulated nodes are placed in a 4000 m $\times$ 4000 m square shape area. The number of mesh clients varies between 50 and 300 by 50 increments. Furthermore, the number of routers is taken as 121. The routers are placed at fixed positions on a grid in the network simulation area. The mesh clients start their movements at random points within the area and do random movements within it. The randomness for the users' movements is obtained by the random path generation algorithm provided in ns-3.13. Packet queue size of mesh routers and relaying mesh clients is assumed to be constant, which is set to 10 packets in our simulations, meaning that some of the packets will be dropped if the queue is full. Therefore, increased number of packets causes an increase in the rate of dropped packets (Table 1).

In our simulations, 30 % of the users are assumed to act as routers, i.e. relaying network users (or agents). Relaying users in this network are not assumed to be a part of the network backbone. Unlike the network operator and mesh routers, they have to authenticate with a router first in order to connect to the network and then perform the relaying activity.

Table 1 Timings for the protocol steps performed by the parties for 80- and 128-bit security

Protocol step	Party	Time (ms) 80-bit (128-bit)
Verification of an anonymous signature	Mesh router	401.8 (811.9)
	Relaying agent	1109 (2.241)
Verification of a conventional signature and anonymous signature generation	Mesh client	229.9 (583.1)
Verification of an anonymous signature and anonymous signature generation	Mesh client	1319 (2774)

All routers are assumed to be informed instantly by the network administrator of the up-to-date CRL using the established network. On the other hand, mesh clients that are acting as relaying agents obtain this updated list from a router only if they are connected to the network. These updates are assumed to be broadcast to corresponding receivers at three different time intervals; 60, 180, and 300 s. Furthermore, in every 30 s, routers broadcast their public parameters together with a signature, the beacon, to all users in vicinity. In addition, if there are any relaying users connected to the routers, they also broadcast their public parameters along with an anonymous group signature in every 30 s. All of the simulations were performed for one-day of simulated time.

In these simulations, it is assumed that mesh clients, either relaying agent or a normal user, are running the protocol steps on a processor with 800 MHz clock frequency (i.e. timings are taken for the platform with Atom™ Processor Z500). On the other hand, mesh routers are assumed to be running on a processor similar to the one used in protocol implementations, a dual core 2.26 GHz processor. Timings used in simulations are computed from the results given in Tables 4 and 5 of [2] (cf. Chap. 6) for the 80- and 128-bit security levels, respectively.

4.1 Scenario 1: UserRL is Held Both at Mesh Routers and Mesh Clients

In this section, results of the simulations performed considering the three different UserRL broadcast time intervals are analyzed. In this current scenario, where mesh clients hold UserRL locally, time intervals are assumed to be 60, 180, and 300 s between each UserRL broadcast.

Figure 2 shows the average authentication time of the mesh clients with respect to the number of the mesh clients within the network for both 80- and 128-bit security levels. Average time of the authentications made by mesh routers and relaying mesh clients are shown separately together with a weighted average of them. The average of all timings obtained from three different simulations corresponding to the three different UserRL broadcast time intervals are given as the authentication time. Weighted average is calculated by dividing the total time spent on all successful

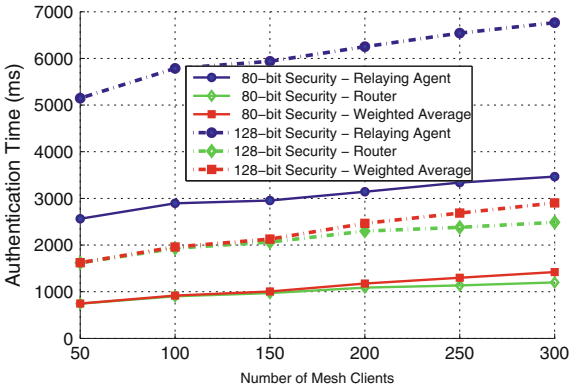


Fig. 2 Authentication times for 80- and 128-bit security levels

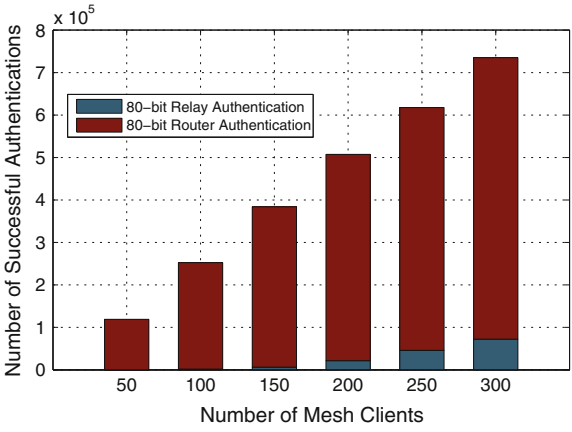


Fig. 3 Number of successful authentications by routers and relaying agents

authentications performed by both parties by the total number of successful authentications.

As seen from Fig. 2, ceteris paribus, average authentication time increases linearly with the increasing number of mesh clients. However, average authentication time increases very slowly as the number of mesh clients increases.Weighted average authentication time increases approximately 85, and 75 % at most for 80- and 128-bit security levels, respectively, with respect to six-fold increase in number of mesh clients.

Number of successful authentications made by relaying mesh clients and routers for 80-bit security level is given in Fig. 3. The results are similar for 128-bit security level. These numbers are used in the calculation of the weighted authentication time and explain why the weighted authentication times in Fig. 2 is nearly the same as the average authentication times resulting from the operation performed by the mesh routers. The latter is due to the fact that, on the average, approximately the 95 % of all the authentications are accomplished by the mesh routers. Furthermore, the

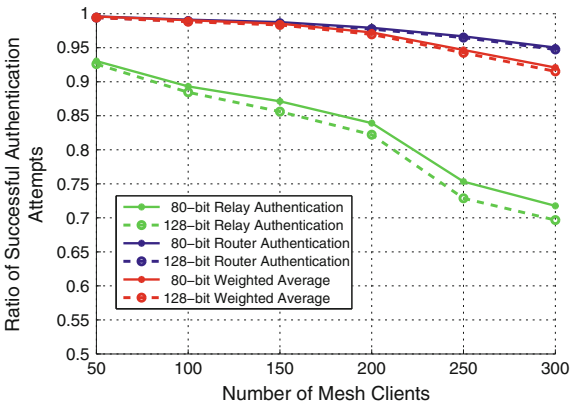


Fig. 4 Ratio of successful authentication attempts (with weighted averages)

total number of successful authentications made increases linearly with respect to increasing number of mesh clients as expected.

Another important metric is the ratio of successful authentication attempts. This metric is calculated as ratio of the number of successful authentications to the number of authentication requests made. Figure 4 demonstrates the ratio of successful authentication attempts made to the mesh routers and relaying mesh clients separately together with the ratio of the weighted average of these successful authentication attempts for 80- and 128-bit security levels. This ratio decreases with the increasing number of mesh clients. This is expected, since the number of packets throughout the network increases with the increasing number of mesh clients, whereas the number of mesh routers stays constant. Furthermore, each mesh router and relaying mesh client can handle only limited number of packets. As it is seen from Fig. 4, ratio drops from nearly 0.92 to 0.70 for the authentication attempts made to the relaying agents as number of mesh clients increases from 50 to 300. On the other hand, a decrease in the ratio is also observed for the authentication attempts made to the mesh routers while it is not as steep. Authentication of mesh clients are performed by the mesh routers and relaying agents where all these authenticators perform UserRL checking locally. Although the mesh routers are informed instantly by the network administrator for the updated UserRL, relaying agents are not able to obtain the updated list if they are not connected to the network during UserRL broadcast. As a result, it is possible for a relaying mesh client to perform authentication with an obsolete UserRL. We call the authentications made by relaying mesh clients with the up-to-date UserRL as true positive authentications. In Fig. 5, ratio of the true positive authentications made by the relaying agents to the total number of authentications is given. As seen from Fig. 5, generally true positive ratio decreases with the increasing UserRL broadcast time interval. However, this behavior becomes less conspicuous with the increasing number of mesh clients within the network. Moreover, security level does not seem to have a meaningful impact on this ratio.

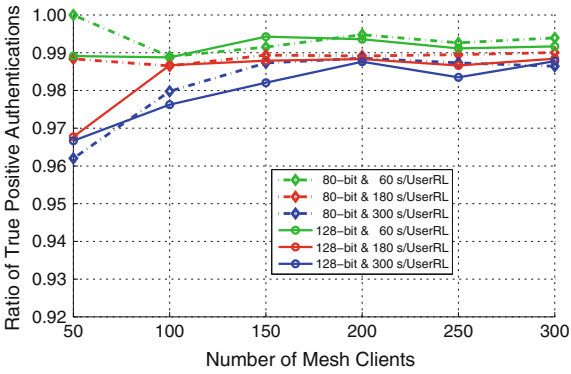


Fig. 5 True positive authentications made by relaying mesh clients

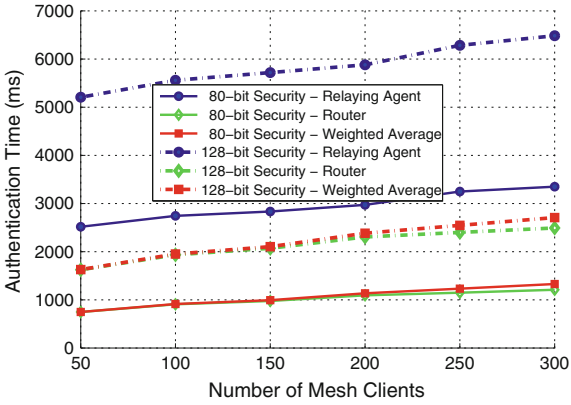


Fig. 6 Authentication times for 80- and 128-bit security levels

4.2 Scenario 2: UserRL is Held Only at Mesh Routers

In this scenario, it is assumed that UserRL is held only at mesh routers and relaying mesh clients do not have access to them. As a result, in order to authenticate another mesh client, relaying agent sends data values used in UserRL checking to the mesh router it is already connected to, and asks this router to perform UserRL checking. In simulations, it is assumed that there are 10 clients in the list throughout the simulated time. Therefore, it is assumed that the mesh routers perform UserRL checking in 0.02026, and 0.04909 s for 80- and 128-bit security levels, respectively.

Figure 6 shows the authentication time of the mesh clients for 80- and 128-bit security levels. Similar to the results obtained from the simulations performed for the first scenario, average authentication time increases linearly with the increasing number of mesh clients. It increases very slowly as the number of mesh clients increases. Weighted average authentication time increases approximately 75, and 65 % at most for 80- and 128-bit security levels, respectively, with respect to a six fold increase in the number of mesh clients. Related figure is the number of successful

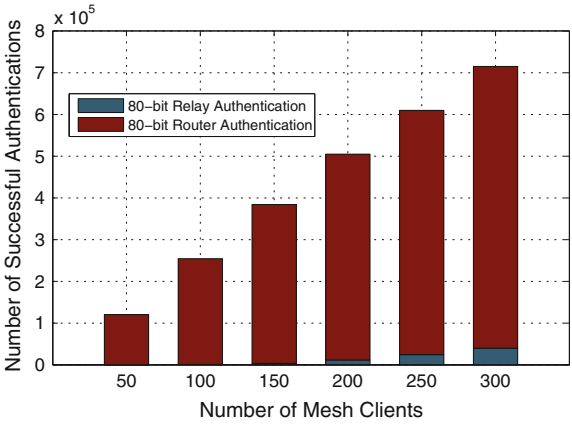


Fig. 7 Number of successful authentications by routers and relaying agents

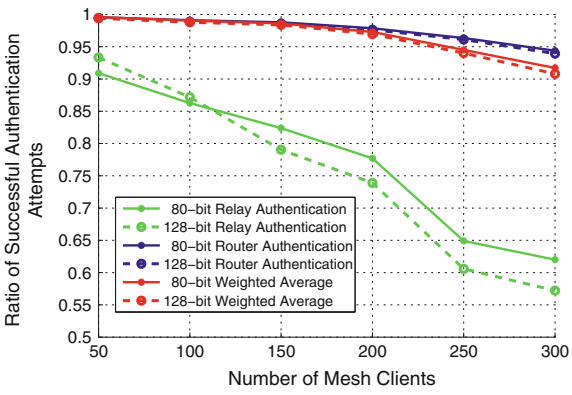


Fig. 8 Ratio of successful authentication attempts (with weighted averages)

authentications made by relaying mesh clients and router. Figure 7 shows the corresponding results for 80-bit security level. The results are similar for 128-bit security level. The ratio of number of successful authentication attempts to the number of authentication attempts made for the second scenario is given in Fig. 8. Figure 8 demonstrates the corresponding ratio for the successful authentication attempts made to the relaying mesh clients and mesh routers separately together with the weighted average of them. Comparing Fig. 8 with corresponding Fig. 4, it is seen that the ratio of the successful authentication attempts is lower for the second scenario where the UserRL checking is performed only by the mesh routers. This difference is notable in authentications made by the relaying mesh clients. This may be due to the increased packet drops throughout the network and increased response time of the mesh routers to the UserRL checking requests.

As a result, authentication times obtained from the simulations performed for this scenario are mostly lower than the ones obtained in the first scenario. This may occur since the authentications that require more time are possibly dropped,

either at the router due to the packet queue being full or within the network, leaving successful attempts having comparatively lower authentication times. This possibly compensates the expected increase in authentication times due to relaying agents waiting acknowledgments for the UserRL checking requests.

Lastly, ratio of true positive authentications is 1.0 in this scenario. This is due to the fact that relaying mesh clients always delegate UserRL checking to mesh routers that possess the up-to-date UserRL.

5 Conclusion

In this work we conducted simulations on A^2-MAKE anonymous authentication framework in order to address the issue of whether checking CRL in authentication is feasible on relaying agents on time (first scenario) or in a lazy manner by mesh routers only (second scenario) since this may become a serious concern as the number of revoked users increases.

To conclude, although the authentication times for both distribution mechanisms show similar behavior, higher ratio of the successful authentication attempts with respect to the second CRL distribution scenario in addition to the higher levels of true positive authentication favors the first scenario to be accepted as the CRL distribution scheme.

References

1. Akyildiz, I.F., Wang, X., Wang, W.: Wireless mesh networks: a survey. Comput. Netw. **47**(4), 445–487 (2005)
2. Durahim, A.O., Savaş, E.: A^2-MAKE: an efficient anonymous and accountable mutual authentication and key agreement protocol for WMNs. Ad Hoc Netw. **9**(7), 1202–1220 (2011)
3. The ns-3 Network Simulator. http://www.nsnam.org. Accessed 02 March 2012

On the Feasibility of Automated Semantic Attacks in the Cloud

Ryan Heartfield and George Loukas

Abstract While existing security mechanisms often work well against most known attack types, they are typically incapable of addressing semantic attacks. Such attacks bypass technical protection systems by exploiting the emotional response of the users in unusual technical configurations rather than by focussing on specific technical vulnerabilities. We show that semantic attacks can easily be performed in a cloud environment, where applications that would traditionally be run locally may now require interaction with an online system shared by several users. We illustrate the feasibility of an automated semantic attack in a popular cloud storage environment, evaluate its impact and provide recommendations for defending against such attacks.

1 Introduction

Cyber criminals are often adept at manipulating people into giving them access to information they need [1]. In an information security context, one can achieve deception by exploiting stereotypical thinking, processing ability, inexperience, truth bias and other semantic attack vectors [2]. We explore the applicability of such a semantic attack in cloud computing, with a proof of concept prototype worm which utilises variants of semantic attack vectors to propagate from one system to another. Our work illustrates the relative simplicity of automating such an attack effectively and without considerable complexity, time or expertise. Our particular case study is on the increasingly popular service of cloud-based storage.

R. Heartfield · G. Loukas (✉)
University of Greenwich, London, United Kingdom
e-mail: g.loukas@greenwich.ac.uk

R. Heartfield
e-mail: hr811@greenwich.ac.uk

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_35,

2 Related Work

Chen and Katz have recently argued that several of the usual underlying information security issues and challenges, such as phishing, downtime, password weaknesses and compromised hosts, not only remain but are often amplified in a cloud environment [3]. Cloud storage services provide applications that allow users and organisations to store information in local directories, synchronised and backed up in the cloud, available to access via web browsers or by installing particular applications on other machines. Chen and Katz have highlighted that this type of shared resource environment constitutes a security issue that is specific to cloud computing [3]. Mulazzani et al. recently showed how such cloud shared storage can be used as an attack platform, identifying in particular an exploitation of the popular Dropbox service [4]. During installation, Dropbox uses a unique host ID to authenticate a device to a user's account. The ID can be stolen via a social engineering guise, such as a spoofed email with a link to a rogue website. This compromises the Dropbox account and gives the attacker full access to all its content. The service itself is not directly attacked, but becomes the deception platform of the attack. Our aim is to illustrate that such social engineering attacks are not only applicable in the cloud, but can even be automated if combined with a worm and a complementary deception infrastructure, without considerable expertise or effort.

3 Worm-Based Semantic Attack

Our implementation assumes that the targets are the users of a popular cloud-based storage service in its usual default set-up in a Microsoft Windows operating system, but makes no assumption as to the technical competency of these users. We have chosen DropBox and SugarSync as the cloud services for our case-study due exclusively to their popularity and not any technical vulnerability, as we target the human rather than the technical aspects of the system. Such services provide simple software applications and web access portals that synchronise local folders to the cloud storage, essentially storing the files on a dedicated storage location that is available from anywhere, at any time, backed up and protected with up-to-date technical security systems. DropBox has an estimated 10 million users and SugarSync is used in almost every country in the world. An automated semantic attack using both would provide a wealth of prospective victims. To achieve such an attack, we developed a novel worm and a complementary deception infrastructure, including spoofed/phishing websites and scareware (Fig. 1).

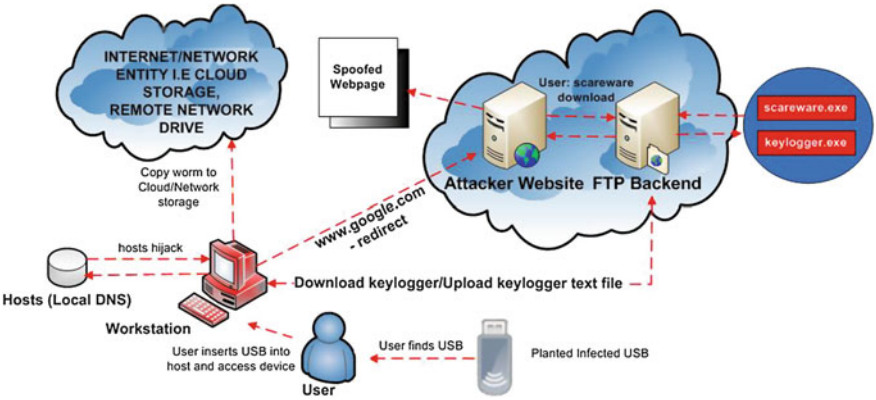


Fig. 1 Attack anatomy

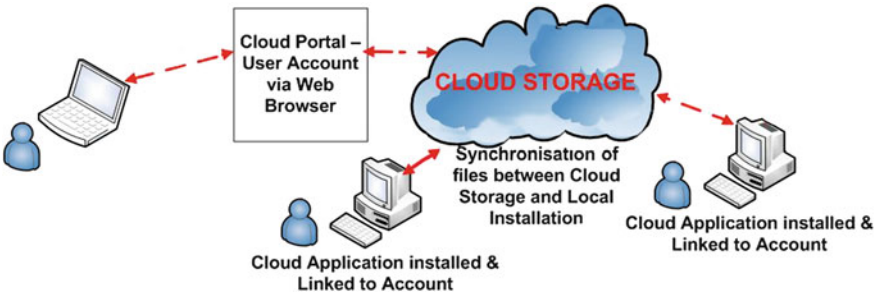


Fig. 2 Architecture of the cloud environment

3.1 The Test-Bed

Due to obvious ethical considerations, the case study worm was contained during development and experimental testing in a sandbox virtual network environment, physically disconnected from the Internet. For this purpose, we implemented the Hyper-V sandbox hypervisor test environment within Microsoft Windows Server 2008 R2. This enables the provisioning of virtual machines within a contained virtual network, without access to external resources outside of the virtual abstraction layer within Hyper-V. This internal network allows communication between virtual machines only and no communication between virtual machines and the hypervisor host or external entities. Internal network communication is performed through a virtual switch in the abstraction layer which transports data between synthetic virtual network interfaces assigned to each virtual machine, essentially producing a workgroup network environment with files replicated and accessible by several users on multiple systems (Fig. 2).

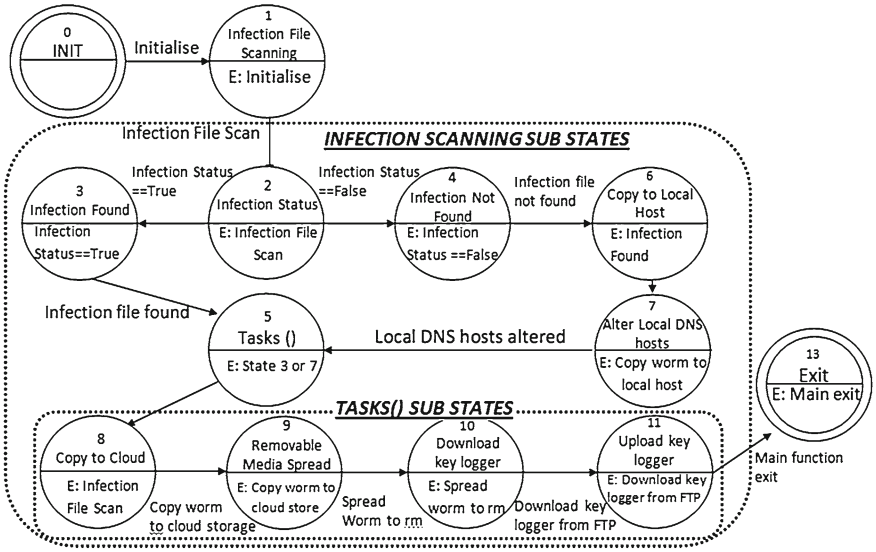


Fig. 3 The worm's finite state machine

3.2 The Worm's Function and Payload

The case study worm relies on manipulation of truth bias [5], essentially deceiving the user to assume a file to be another legitimate one, using a file masquerading technique. The original file is deleted and the malformed file takes its place. Figure 3 shows the finite state machine diagram for the worm.

By seeing the expected filename and extension, the user is likely to believe at first impression that this is the original file, especially as there is no other file with similar name and the user themselves have not removed the file. Should the user notice that the file has changed in any way, such as its icon or description, they are now likely to be curious as to why this has happened when the filename appears correct. Both these emotive reactions (truth bias and curiosity) entice the user to open the file and by doing so execute the worm program. At this point the worm will run without visual execution or further interaction with the user, who may now believe that the file is corrupted. The worm has hidden within the user's profile directory and on next login will execute the rest of its payload, including more file masquerading deceptions, a keylogger etc.

The attack may also involve a USB media semantic attack vector, with the worm running as a hidden executable autorun file. The semantic vector here is curiosity and desire to explore the USB memory device. Current trends in worms demonstrate operational relationships with hierarchical overlay systems (Zeus, Storm, Conficker), where the worm can be defined as an agent of a parent system to deliver or direct a

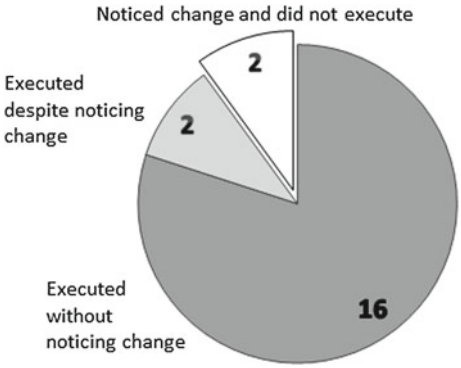


Fig. 4 Experimental results of worm execution

specific attack. To depict this, we introduce a typical malicious backend architecture consisting of a spoofed website (Google Search Engine) and scareware application (fictitious Google anti-virus), which illustrates how the worm can be used for fraud, data theft and other exploitations.

4 Experimental Evaluation

Before evaluating the impact of our automated semantic attack we first tested the deception element. Twenty students, randomly selected from our university's computing programmes for a fictitious survey, were given a brief overview of the use of Cloud storage and were asked to create a number of files of different types within the Dropbox/SugarSync folder, containing personal data. The system was reset and the user was asked to locate each file and alter its content. Figure 4 shows that only two out of the 20 students refrained from opening the file. Out of the remaining 18 who did execute the worm, 16 never noticed a change in the file, while two spotted inconsistencies but ran it nevertheless.

Although a sample of 20 students may not be large, these participants were highly technical individuals, enrolled in computing programmes, ranging from information systems to security and forensics. Yet, in their majority they were deceived into executing the worm. From this point on, for ethical reasons, the rest of the experimental process was carried out in an isolated environment without further participant input, with the assumption that the worm had been executed. Tables 1, 2, 3, 4, 5 show the impact observed in different experimental scenarios. We used two popular protection systems, McAfee VirusScan Enterprise 8.7.0i and Microsoft Security Essentials with the latest updates installed. None of the two alerted the user of malicious activity or hindered the success of any of the attacks described here.

Table 1 Scenario 1. Windows XP SP3 Blackbox test—no anti-virus (no USB infection phase)

Step	Test action	Impact
1	Copy worm to local host user profile directory	'csrss.exe' stored in local directory and relevant registry entry is created
2	Alter 'hosts' file under system directory for local DNS	'hosts' file contains IP—domain name mappings to redirect Google to phishing site
3	Worm will be executed on XP virtual machine build in current clean state	SugarSync and DropBox cloud applications storage directories should be infected with worm via file masquerading exploit
4	Copy worm to DropBox and SugarSync Cloud application directories	Worm impersonates existing file in cloud directory and removes the original
5	Worm spreads via removable and remote/network media	Worm impersonates existing file n network storage directory and removes the original
6	Worm downloads the key logger payload executable from the attacker FTP server	Keylogger stored locally and registry value for execution on login is created under the current user key space
7	Worm uploads key logger text file with captured keys to attacker FTP server	Upload will fail as key logger will not have run and no text file exists

Table 2 Scenario 2. Windows XP SP3 WhiteBox test—no anti-virus (no USB spread phase)

Step	Test action	Impact
1	Execute worm on laptop with USB media attached	Worm and autorun file copied to removable media

Table 3 Scenario 3. Windows XP SP3 BlackBox test—anti-virus installed

Step	Test action	Impact
1	Worm will be executed on XP virtual machine build in current clean state, with anti-virus installed	Anti-virus does not identify worm running on host
2	Checking worm functions success	Worm successfully completes all operations without anti-virus intervention or alert
3	Worm and Keylogger execution on user login	Keylogger and worm successfully run without anti-virus intervention or alert

5 Defence Recommendations and Future Work

Proactive and pre-emptive solutions against social engineering are still at the stage of best-practice suggestions and have not been agreed as standards [6, 7]. The *Signing Seal* technology employed by Yahoo [8] and web application and data security used in RAPPORT [9] help to determine site legitimacy and identity assurance and can be combined to provide a baseline technological solution. Data tagging schemes and

Table 4 Scenario 4. Windows 7 BlackBox Test—anti-virus installed

Step	Test action	Impact
1	Worm will be executed on Windows 7 virtual machine build in current clean state, with anti-virus installed	Anti-virus does not identify worm running on host
2	Checking worm functions success	Worm successfully completes all operations without anti-virus intervention or alert
3	Worm and Keylogger execution on user login	Keylogger and worm successfully run without anti-virus intervention or alert

Table 5 Scenario 5. Phishing site and Scareware testing— (XP build anti-virus installed)

Step	Test action	Impact
1	Instance of Internet Explorer opened and URL www.google.com browsed to	Hosts file redirects DNS mapping to phishing site IP address
2	Free download of Google Guard anti-virus initiated by hyperlink	Zip file containing scare ware prompts for download
3	Attempt to use Google search functions and tools	Redirection to phishing site error page
4	Scareware application executed and application virus check and cleaning functions used	Scareware displays infections, enable cleaning options. On selection redirect to an activation 'attacker' web page prompting for email credentials from the user for activation
5	User has entered credentials and received activation code, enters wrong code into worm.	Activation unsuccessful and prompts for correct code
6	User enters correct code	Activation confirmation, complete clean enable
7	User selects completed clean	Clean confirmation, all options disabled and hosts file cleaned

enforcement techniques have been proposed as the basis for end-to-end application security in the cloud [10], but they do not take into consideration the human element. Even if a cloud application prevents an external process from accessing the synchronised directory in the local machine, it still does not prevent the user from uploading/downloading a malformed file to/from the cloud application.

What we are missing is a solution that takes into account not only the technical system configurations, but also the users themselves as individuals with, for example, different levels of curiosity or risk aversion in different situations. Protection against semantic attacks may require a hybrid approach, combining technical access control with user conformity, education and training. Current controls recommend "least user rights" and comprehensive security training and education to build up a user's security profile [11], which may be feasible in a business environment, but a home

user would be unlikely to offer themselves least user rights or build their own security training policy. For home users, we believe that an online initiative that would involve cloud service providers rewarding users who build their security profile in compliance with best practices, would be beneficial.

6 Conclusions

The aim of this paper was to demonstrate the feasibility of an automated semantic attack in popular cloud storage services. The case study worm exploits the cloud service's reliance on the interface provided to the user by the user's own operating system. It merely requires the user to open the file for the exploit to be complete. Then, the file in the Cloud storage is replicated all over the Cloud infrastructure for that user's account. Since no specific technical vulnerability of the system is targeted and all processes performed in the technical manner they were supposed to, such an automated semantic attack cannot be detected by technical security software. The point demonstrated with this work is that semantic attacks cannot only adapt, but even be automated in the cloud, because they exploit core market drivers behind cloud computing, such as convenience, reduced ownership and technical simplicity.

References

1. Schneier, B.: Inside risks: semantic network attacks. Commun. ACM **43**(12), 168 (2000)
2. Tiantian Qi.: An investigation of heuristics of human judgement in detecting deception and potential implications in countering social engineering. In: IEEE International Conference on Intelligence and Security Informatics, pp. 152–159, New Brunswick, USA, May 2007
3. Chen, Y., Katz, R.H.: Glimpses of the Brave New World for Cloud Security. Feature Article. HPC in the Cloud, 22 Feb 2011
4. Mulazzani, M., Schrittwieser, S., Leithner, M., Huber, M.: Dark clouds on the horizon: using cloud storage as attack vector and online slack space. In: Proceedings of the 20th USENIX Conference on Security, CA, USA, 10–12 Aug 2011
5. Levine, T.R., Kim, R.K., Park, H.S., Hughes, M.: Deception detection accuracy is a predictable linear function of message veracity base-rate: a formal test of Park and Levine probability model. Commun. Monogr. **73**, 243–260 (2006)
6. Hinson, G.: Social engineering techniques, risks and controls. The EDP Audit, Control, and Security. Newsletter 37, 32–45 (2008)
7. Latze, C., Ultes-Nitsche, U.: How to Protect even Naive User against Phishing, Pharming and MITM Attacks. Communication Systems, Networks, and Applications, pp. 111–116, Beijing, China, Oct 2007
8. Yahoo Inc. What is a sign-in seal? Retrieved from Yahoo Security Center. http://security.yahoo.com/article.html?aid=2006102507 (2012)
9. Trusteer. Rapport Overview. Retrieved from Trusteer Building Trust Online. http://www.trusteer.com/product/trusteer-rapport (2011)
10. Bacon, J., Evans, D., Eyers, D.M., Migliavacca, M., Pietzuch, P., Shand, B.: Enforcing end-to-end application security in the cloud. In: 11th International Middleware Conference, Bangalore, India, Nov 2010

11. Hasan, M.I., Prajapati, N.B.: An attack vector for deception through persuasion used by Hackers and Crackers. Networks and Communications, pp. 254–258, ISBN 978-1-4244-5364-1, 27-29 Dec 2009
12. Mitnick, K., Simon, W.L.: The Art of Deception: Controlling the Human Element of Security. Wiley, Indianapolis (2002), ISBN 978-0471237129
13. Jordan, M., Goudey, H.: The signs, and semiotics of the successful semantic attack. In: 14th Annual EICAR Conference, Malta, pp. 344–364. 2005

Topic Tracking Using Chronological Term Ranking

Bilge Acun, Alper Başpınar, Ekin Oğuz, M. İlker Saraç and Fazlı Can

Abstract Topic tracking (TT) is an important component of topic detection and tracking (TDT) applications. TT algorithms aim to determine all subsequent stories of a certain topic based on a small number of initial sample stories. We propose an alternative similarity measure based on chronological term ranking (CTR) concept to quantify the relatedness among news articles for topic tracking. The CTR approach is based on the fact that in general important issues are presented at the beginning of news articles. By following this observation we modify the traditional Okapi BM25 similarity measure using the CTR concept. Using a large standard test collection we show that our method provides a statistically significantly improvement with respect to the Okapi BM25 measure. The highly successful performance indicates that the approach can be used in real applications.

1 Introduction

News portal web sites deal with huge amount of data from different sources. As the number of sources and events increase, news-consumers are overwhelmed with too much information. Different organizational techniques have been employed for more effective, efficient, and enjoyable browsing [1]. Studies on new event detection and topic tracking aim to organize news with respect to events or topics. In this work, we study topic tracking (TT) which aims to find all news articles that follow an initial event/topic.

In topic detection and tracking (TDT) studies, an event is defined as something that happens at a given "place and time, along with all the necessary preconditions and unavoidable consequences," like a car accident. Topic is a connected series of events

B. Acun · A. Başınar · E. Oğuz · M. İ. Saraç (✉) · F. Can
Computer Engineering Department, Bilkent Information Retrieval Group,
Bilkent University, 06800 Ankara, Turkey
e-mail: ilker1486@gmail.com

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_36,

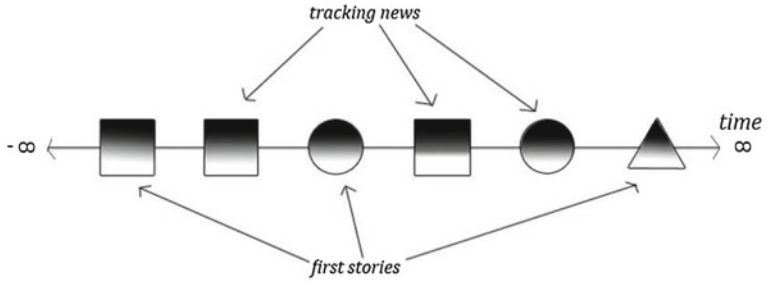


Fig. 1 Illustration of topic tracking (TT) and chronological term ranking (CTR), in the figure different *shapes* indicate stories related to different topics. *Darker gray* initial positions in each story indicate positions of more important words. CTR is based on the inverted pyramid metaphor that implies that most newsworthy information is provided at the beginning and as we go down to lower portions of a story importance of words gradually decreases

that have a common focus or purpose [1]. It is not a broad concept like "accidents," it is limited to a specific accident. In this study, we investigate the use of chronological term ranking (CTR) in TT to identify tracking stories of an event, i.e. we aim to find all tracking news articles of a topic based on a few number of sample initial news articles. To the best of our knowledge, prior to our work CTR approach has not been used for TT. This lack in literature, i.e., not having a CTR-based TT study in literature, is surprising since CTR is a natural fit to the TT problem domain. We use the Okapi BM25 (in short Okapi) similarity function in TT. It is a commonly used similarity function that gives good results in information retrieval and works well when paired with CTR [2]. Figure 1 illustrates the TT process and the CTR concept.

The contributions of this study are the following. (1) Our work extends the previous studies on TT by introducing the CTR approach to similarity calculation for identifying tracking stories in TT, (2) We experimentally show that term weighting component of Okapi can be altered by term position information (as suggested in [2]) such that its performance in TT can be statistically significantly improved, and (3) Successful results we obtain with the CTR approach show that the approach provides a performance that is compatible with those of previous studies [1] and can be used in practical applications with a high user satisfaction.

2 Related Work

Topic Detection and Tracking (TDT) studies were initiated in the second half of the 1990s by researchers from DARPA, Carnegie Mellon University, Dragon Systems and University of Massachusetts Amherst. Later some other groups also joined to the research initiative [3]. In TDT there are five tasks and they are New Event Detection (NED), Topic Tracking, Story Segmentation, Topic Detection (Cluster Detection), and Story Link Detection. Two of them, NED and TT, are more frequently studied in

literature and are also studied in Turkish in our previous research [1]. In NED, one common approach is keeping a sliding time-window due to performance concerns [1, 4]. In this approach, each incoming news article is compared with the previous articles stored in the window. For each article pair, if similarity value of the new article is below a similarity threshold value (usually obtained by training) for all other articles then it is identified as new. In TT, initial story (or a few numbers of initial stories) is provided then all incoming subsequent stories are compared with that initial story on the basis of a threshold value. If similarity value is above the threshold this article is flagged as a tracking article (follower). All tasks were understood as detection tasks and evaluated using miss and false alarm error rates [3]. A Detection Error Tradeoff (DET) plot [5] is the primary tool for describing the tracking errors.

This study follows our earlier studies on new event detection and topic tracking in Turkish [1] and information retrieval on Turkish texts [6]. In the experiments we use a large standard test collection BilCol2005 from [1]. In order to have objective initial idf (inverse document frequency, explained later in Sect. 4) values, a retrospective corpus need to be used [4], for that purpose we have another data collection, *Milliyet* Collection from [6].

CTR concept is previously only used to enhance relevance scoring between documents in information retrieval [2]. This idea fits perfectly to news stories since news reporters write articles using inverse pyramid style which consists of writing most important words in the initial sentences. The work reported in [2] shows that the CTR approach works well when paired with Okapi; however, as indicated earlier has not been used in topic tracking applications.

3 Test Collection and Topic Tracking Algorithm

We use two test collections which are BilCol2005 [1] and *Milliyet* Collection [6] in our TT system. BilCol2005 is a large TDT collection that contains 209,305 news stories and 80 topics spanning through 12 months in 2005. In our experiments the topics and stories of the first 8 months (141,910 articles containing several topics 50 of them have been annotated) in the collection are used for training and the remaining 4 months (67,395 articles in 30 topics) are used for testing. Details about the BilCol2005 collection can be found in [1]. *Milliyet* Collection contains 408,305 news articles from *Milliyet Gazetesi* between 2001 and 2004. We use *Milliyet* Collection to calculate idf values. Using these idf values, we start our experiments in an independent unbiased environment for BilCol2005. The details about the *Milliyet* Collection can be found in [6].

The TT algorithm uses a few number of sample stories about a given topic and aims to find the tracking stories for that particular topic in a news stream. In the literature, usually between 1 and 4 documents are used as sample stories [1].

We employ the traditional TT algorithm [1]. In the algorithm a similarity function is used to determine if an article of the news stream that follows the sample story is a tracking story or not. If the similarity value is higher than the threshold value obtained

during training it is classified as a tracking story. During training, for each training topic we calculate miss and false alarm rates using a threshold sweep within a range of similarity values (in our case it is between 0 and 121.1). The threshold values for the individual training topics that make the corresponding C_{det} value minimum are determined; where C_{det} signifies TT cost and calculated as a function of miss and false alarm rates [7] (defined in Sect. 5). The average of these threshold values are used during testing to obtain the performance. During the sweep, as in [2] we use 20 threshold values and go with the increments of 6.055 (which is equal to 121.1/20).

4 Experimental Design

News articles should be preprocessed by extraction, tokenizing and stemming before the similarity calculation process. Extraction includes getting the news articles from the collection, elimination of punctuation marks and stopwords. Stopwords are the words which are meaningless on their own but essential within the language (typical examples for English include words such as "the," "is," "at," "which," "on"). We eliminate all punctuation marks and use 217 stopwords which gives the best results in Turkish TDT experiments [8]. We use a Turkish NLP library, Zemberek [9] as a lemmatizer-based stemmer to eliminate the suffixes and prefixes and turn the words into their roots.

We perform TT experiments using the pure Okapi similarity function which we take as a baseline and CTR-based Okapi similarity functions in two different forms (additive and multiplicative).

4.1 Okapi Similarity Function

Okapi is a term frequency-inverse document frequency (tf-idf) based similarity function which calculates the similarity measure between two vectors (d, q) with the following formula [2].

$$sim(d, q) = \sum_{t \in d,q} w_{tf,d} \cdot w_{tf,q} \cdot w_{idf}$$

In Okapi, idf calculation is as follows.

$$w_{idf} = \text{In } \frac{N - df + 0.5}{df + 0.5}$$

Where df = number of documents that includes term t, N = total number of documents in document collection.

In Okapi, tf calculation is as follows.

$$w_{tf} = \frac{(k+1) \cdot tf}{k\left[(1-b) + b.\frac{dl}{avdl}\right] + tf}$$

Where b = 0.75, k = 1.2, dl = document length in terms of number of words (tokens), avdl = average document length.

4.2 Okapi Similarity Function with CTR

We add the term rank component additively and multiplicatively into the Okapi similarity function. They are defined as follows.

Additive CTR

$$sim(d,q) = \sum_{t \in d,q} (w_{tf,d} + R_{t,d}) \cdot (w_{tf,q} + R_{t,q}) \cdot w_{idf}$$

Multiplicative CTR

$$sim(d,q) = \sum_{t \in d,q} (w_{tf,d} \cdot R_{t,d}) \cdot (w_{tf,q} \cdot R_{t,q}) \cdot w_{idf}$$

The rank coefficient R ($R_{t,q}$. $R_{t,d}$) can be calculated as an inverted absolute rank: C/tr or as a percentage rank: $C \cdot tr/dl$, where C is a constant generally between 0 and 1 giving the best experimental results for term rank *tr*. The C values used in the experiments are adopted from [8] (see Appendix E). In [8] C values are determined for NED; however, since NED and TT are the two sides of the same coin it makes sense to use the C values obtained for NED for TT. All function combinations for the rank coefficient R in both additive and multiplicative CTR are adopted from [2]. In total 21 different formulas are evaluated.

5 Evaluation Methodology

In TT the most common evaluation measures are false alarm (FA) and miss rate (MR), more specifically their probabilities P_{FA} and P_{MR}. They are defined as follows.

- $P_{FA} = \frac{FA}{Number\ of\ non-tracking\ stories}$
- $P_{Miss} = \frac{MR}{Number\ of\ tracking\ stories}$

where

- FA = number of non-tracking stories labeled as tracking stories,
- MR = number of tracking stories labeled as non-tracking stories.

From the combination of these FA and MR values a detection cost formula is formed as a single metric for measuring the effectiveness.

$$C_{det} = C_{Miss} \cdot P_{Miss} \cdot P_{Target} + C_{FA} \cdot P_{FA} \cdot (1 - P_{Target})$$

where

- $C_{Miss} = 1$ and $C_{FA} = 0.1$ are the prespecified costs of a missed detection and a false alarm
- $P_{Target} = 0.02$, the *a priori* probability of finding a target as specified by the application [7].

In all calculations we use normalized C_{det} because in the given formula for C_{det} has a dynamic range of values which is difficult for relative comparison. In normalized C_{det}, C_{det} is divided by the minimum expected cost [7].

$$(C_{det})_{Norm} = \frac{C_{det}}{Minimum\{C_{Miss} \cdot P_{Target}, C_{FA} \cdot (1 - P_{Target})\}}$$

- Improvements of the functions with respect to the Okapi baseline are calculated as follows.

$$Improvement(\%) = \frac{(C_{det})_{okapi} - (C_{det})_{compared}}{(C_{det})_{okapi}} \cdot 100$$

6 Experimental Results and a Real Life Application

As illustrated in Table 1 the experimental results show that CTR-based approach is highly effective and in most of the cases does improve the performance of TT with respect to the baseline Okapi performance. Note that due to rounding of the C_{det} values Improvement (%) and C_{det} values do not match. In this table, for repeatability, the threshold values obtained by training are also provided. (The details for the other 19 formulas are not shown due to space limitation.) The CTR-based Okapi function that gives the highest improvement, which is 18 %, is the additive function *ai*. The p values are obtained by two-tailed t-tests when the results of a particular CTR-based Okapi measure is compared with those of the baseline. The results (several of them are not listed due to space limitation as we indicated above) show that in most of the cases the difference, in this case improvement, provided by the CTR approaches is either statistically significant or very strongly statistically significant. Furthermore,

Table 1 Test results of the experiment (Among 21 different formulas [2, 8], only the best additive (ai) and only the best multiplicative (mdbl) formulas are provided.)

	Training		Testing				
	$(C_{det})_{min}$	Threshold	P_{miss}	P_{FA}	C_{det}	Improvement (%)	P value
Okapi	0.188	48.440	0.075	0.009	0.120	N/A	N/A
ai	0.170	54.495	**0.064**	**0.007**	**0.098**	**17.934**	0.001**
mdbl	0.163	60.550	0.063	0.008	0.101	**15.870**	0.002*

* significant; ** very strongly significant

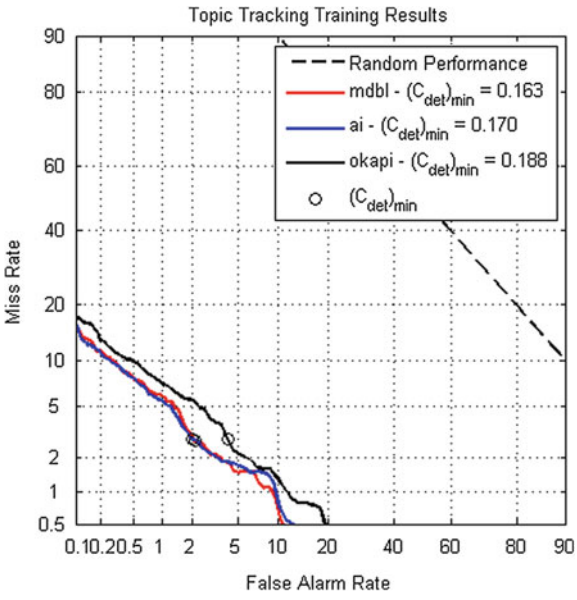


Fig. 2 TT training performance with mdbl-Okapi, ai-Okapi, and Okapi in terms of DET plots

ai gives the most desirable performance as indicated by its C_{det} value. The two-tailed t-test results also show that the improvement of C_{det} with *ai* with respect to the baseline Okapi function is very strongly significant ($p = 0.001$).

We also use Detection Error Tradeoff curve (DET) for representing the system performance. DET is a curve to see the tradeoff between miss and false alarm rate as shown in Fig. 2. It is obtained during training by using the threshold sweep method [1] and reveals what to expect during testing. DET curves are plotted on a Gaussian (normal) deviate scale which has advantages with respect to linear scales since it expands the high performance region [1, 7, p. 24]. The minimum C_{det} values of the functions are shown in circles. If the circle (or line) is closer to the origin, it means C_{det} value of corresponding point (or line) is smaller. We plotted baseline Okapi similarity function and functions with the greatest improvement in additive CTR and multiplicative CTR, which are *ai* and *mdbl*. These *ai* and *mdbl* functions mostly overlapped in Fig. 2; both their curve and min C_{det} points. As seen from the figure and from C_{det} values, *ai* and *mdbl* functions are significantly better than Okapi

similarity function. The curve of baseline Okapi function is above the other curves and its min C_{det} point is more separated from the origin than those of *ai* and *mdbl*.

As shown in Table 1, in the testing phase *ai* function provides miss and false alarm rates of 6.40 and 0.70 %, respectively. This means that out of 100 tracking stories we would miss approximately 6 of them and only one of the stories would be an incorrect choice. These results are compatible with those presented in [1]. In that work refer to Table 1 and look at the best static TT performance using the stand alone cosine similarity measure that provides miss and false alarm rates of 4.94 and 0.71 %, respectively.

Based on experimental results, we also developed an Android application [10] for end users. It aims to show news stories to the users according to their choice for tracking. On the server side our application continuously fetches news stories from news sources and makes the decision of tracking on the fly and saves them in our server.

7 Conclusions and Future Work

Topic tracking (TT) is an important component of TDT systems in various information aggregation applications like news and blog portals. We investigate the TT problem within the framework of the Okapi measure by employing the concept of chronological term ranking (CTR). We propose an alternative method for TT using the CTR concept. For this purpose, we extend the Okapi similarity measure with a CTR component in various ways. The experimental results and statistical tests show that in the majority of the cases CTR significantly improves the performance obtained by the original Okapi similarity measure and is highly successful and can be used in real life applications.

In future work, the cosine similarity coefficient or other similarity functions can be extended with the CTR approach. The results of the CTR-based Okapi approaches and other approaches can be combined to improve the effectiveness to an even higher level [1]. Furthermore, the CTR-based approach can be used in the implementation of various information aggregators for topic tracking and also for story link detection or information filtering.

Acknowledgments This work is partially supported by the Scientific and Technical Research Council of Turkey (TÜBİTAK) under the grant number 111E030. We thank Süleyman Kardaş of Sabancı University for his helps in performance evaluation.

References

1. Can, F., Kocberber, S., Baglioglu, O., Kardas, S., Ocalan, H.C., Uyar, E.: New event detection and topic tracking in Turkish. J. Am. Soc. Inf. Sci. Technol. **61**(4), 802–819 (2010)
2. Troy, A.D., Zhang, G.: Enhancing relevance scoring with chronological term rank. In: Proceedings of the ACM SIGIR'07 Conference, pp. 599–606 (2007)

3. Allan, J.: Introduction to topic detection and tracking. In: Allan, J. (ed.) Topic Detection and Tracking: Event-based Information Organization, pp. 1–16. Kluwer Academic Publishers, Norwell (2002)
4. Yang, Y., Pierce, T., Carbonell, J.: A study on retrospective and on-line event detection. In: Proceedings of the ACM SIGIR'98 Conference, pp. 28–36 (1998)
5. Topic Detection and Tracking Evaluation: NIST Information Access Division. DET-curve plotting software tool. http://www.itl.nist.gov/iad/mig//tests/tdt/ (2007). Accessed 14 April 2012
6. Can, F., Kocberber, S., Balcik, E., Kaynak, C., Ocalan, H.C., Vursavas, O.M.: Information retrieval on Turkish texts. J. Am. Soc. Inf. Sci. Technol. **59**(3), 407–421 (2008)
7. Fiscus, J.G., Doddington, G.R.: Topic detection and tracking evaluation overview. In: Allan, J. (ed.) Topic Detection and Tracking: Event-based Information Organization, pp. 17–31. Kluwer Academic Publisher, Norwell (2002)
8. Baglioglu, O.: New event detection using chronological term ranking. Master thesis, Computer Engineering Department, Bilkent University, Ankara, Turkey (2009). http://www.cs.bilkent.edu.tr/canf/bilir_web/theses/ozgurBaglioglu Thesis.pdf
9. Zemberek, open source NLP library for Turkic languages. http://code.google.com/p/zemberek/. Accessed 5 Jan 2012
10. BilTracker Android Application Beta Demo Extended. http://youtu.be/MnyTO8bendU. Accessed 5 May 2012

Clustering Frequent Navigation Patterns from Website Logs by Using Ontology and Temporal Information

Sefa Kilic, Pinar Senkul and Ismail Hakki Toroslu

Abstract In this work, clustering algorithms are used in order to group similar frequent sequences of Web page visits. A new sequence is compared with all clusters and it is assigned to the most similar one. This work can be used for predicting and prefetching the next page user will visit or for helping the navigation of user in the website. They can also be used to improve the structure of website for easier navigation. In this study the effect of time spent on each web page during the session is also analyzed.

Keywords Frequent navigation patterns · Clustering · Ontology · Web page recommendation · Semantic similarity

1 Introduction

A useful application area of web mining is the dynamic web page recommendation. Based on user navigation, next web page that is likely to be requested by user is recommended to him/her. For example, in a web site, if users usually access page B from page A, B can be recommended to a user who is on page A. Similarly, an important usage is prefetching and caching [2]. In the previous example, when the user browses page A, next page is likely to be B. While user is browsing A, B is fetched and cached. If user requests B, the copy in the cache is directly displayed

S. Kilic · P. Senkul (✉) · I. H. Toroslu
METU Computer Engineering Department, 06800 Ankara, Turkey
e-mail: senkul@ceng.metu.edu.tr

S. Kilic
e-mail: sefe.kilic@ceng.metu.edu.tr

I. H. Toroslu
e-mail: toroslu@ceng.metu.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_37,

without waiting the server for the content. The aim is to make browsing faster for the user.

Hyperlinks can dynamically be inserted into web pages. Consider the following example. If there is a significant web navigation pattern of pages $A \rightarrow B \rightarrow C \rightarrow D$ and time spent on B and C are considerably small, it can be interpreted that users follow this navigation path to access page D from A. In this case, inserting a link from A to D may make the browsing easier for users. As well as dynamic hyperlink generation, the navigation patterns can also be used for evaluation of quality of the website [2].

In this study, collected web navigation paths are clustered based on semantic similarity between paths.Contributions of this study can be summarized as follows: combining concept-based sequence clustering with time-spent information, proposing a new concept similarity metric and applying web usage mining on a non-commercial web domain; previous studies are usually on commercial web sites [8].

2 General Procedure for Clustering Navigation Patterns

In this study, web navigation information of users is integrated with the set of concepts defining web pages. Each session is a sequence of web pages and each web page is represented with a set of concepts from the taxonomy defined. Sessions are clustered to find meaningful partition with the aim of maximizing intra-cluster similarity while minimizing inter-cluster similarity.

In navigation clustering, the first step is the collection of web server logs and preprocessing of them. In preprocessing phase, sessions are constructed from logs and using manually defined taxonomy, each web page is mapped to a set of concepts from the taxonomy.

After preprocessing step, a series of similarity measures are needed to cluster sessions. To measure similarity of two session $S_i = \langle P_1^{(i)}, \ldots, P_{n_i}^{(i)} \rangle$ and $S_j = \langle P_1^{(j)}, \ldots, P_{n_j}^{(j)} \rangle$, the similarity among web pages of these two sessions is measured. The similarity between two web pages $P_a^{(i)}$ and $P_b^{(j)}$ for all pairs of a and b such that $1 \leq a \leq n_i$ and $1 \leq b \leq n_j$ has to be determined. It is measured using the similarity between two sets of concepts $C_a^{(i)}$ and $C_b^{(j)}$ defining $P_a^{(i)}$ and $P_b^{(j)}$, respectively. To measure similarity between sets of concepts, we need to define the similarity between concepts. After definition of similarity measures, web sessions are partitioned into clusters. To assign a new instance to one of the clusters, it can be compared with all clusters and the closest cluster should be selected.

To train and test our system, we used access logs of Middle East Technical University (METU) Computer Engineering Department website.[1] In this study, we use simple heuristics to process data which are briefly explained below.

[1] http://www.ceng.metu.edu.tr.

Table 1 Sample from server access log

IP	Time	URL
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/csToolbar.html
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/csMain.html
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/img/smLogoBlack.gif
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/img/tbAnnoun.gif
1.2.3.4	[13/Feb/2011:20:10:27 +0200]	/courses/ceng436/lectures/index.html

Only IP address, time and requested URL fields are given. IP address is changed for privacy protection

Data Cleaning. From web logs, we remove log items that are not useful for extraction of navigation patterns. We do not process multimedia files, archive files and external documents. Therefore, we remove logs of requests to these items. To identify these items, we use suffixes of filenames requested. For detection and removal of logs belong to web crawlers, we use a simple heuristic. We identify all IPs accessed to `robots.txt` as web crawlers and remove all access logs belonging to these IPs from the dataset.

Page View Identification. Multiple files of text and graphics can be loaded in the same view. After cleaning graphics, these text files are combined as a page view. All page requests from a single IP address within the same second are considered as members of the same page view. Sample from server log of www.ceng.metu.edu.tr is given in Table 1. The page `courses/ceng436` consists of two frames: `csToolbar.html` and `csMain.html`. `csToolbar.html` has two GIF image files and `csMain.html` has no image files embedded into it. When the client `1.2.3.4` requests `/courses/ceng436`, two `HTML` files and six GIF image files are requested too. Assuming GIF image files are removed in data cleaning step, there would be three log items with same time fields: `/course/ceng436`, `/course/ceng436/csToolbar.html` and `/course/ceng436/csMain.html`. Since time fields of all three of them are same, they are considered in the same page view. The time field of the last URL is different, therefore it belongs to a different page view.

User Identification and Session Construction. We apply a simple heuristic to identify users. We use IP address and agent fields of logs together and assign two different logs to the same user if their IP address and agent fields are same. For session construction, we adopt time-oriented heuristic. We consider two consecutive visits of a user in the same session if access time difference between them is not more than some threshold t. In our study we use $t = 30$ min.

Table 2 Some concepts from the ontology and keyword sets describing them

Concept	Associated keyword sets
`ResearchLaboratory`	{research, laboratory}
`Bioinformatics`	{bioinformatics}
`ImageProcessing`	{image, processing}, {pattern, recognition}
`OperatingSystems`	{operating, system}, {process}, {thread}, {deadlock}, {memory, management}
`UndergraduateStudent`	{undergraduate, student}
`GraduateStudent`	{graduate, student}

3 Building Taxonomy

Instead of using the content (usually in text format) of web pages directly, each web page is described with a set of keywords. To assess similarity of web pages, similarity of these keywords are used. Taxonomy of keywords (concepts) is built to model properties of these concepts and relationships among them. ISA ("is a") hierarchy (taxonomy) of the example computer science department ontology from Simple HTML Ontology Extensions (SHOE) project [3].[2]

For measure of similarity between two web pages, we map each web page to a set of concepts in the ontology defined. Each concept in the taxonomy is associated with some keywords. A set of concepts and keywords describing them are given in Table 2. Each web page is represented as a bag of words. In other words, it is represented as unordered collection of words. A concept is in the mapping of a web page if all keywords in one of associated keyword sets appear in the web page. For example, to label a web page P with concept `ImageProcessing`, P should contain both words "image" and "processing", or it should contain both "pattern" and "recognition".

4 Similarity Calculation

In order to measure similarity of two web pages (two sets of concepts), we need pairwise similarity/distance measure for similarity/distance of two concepts. All of the used concepts are represented in a taxonomy and taxonomic relations of concepts are used to measure similarity/distance between them.

To improve the similarity score between two web pages, the importance component is introduced and used together with the similarity component [1]. The importance component computes how important the similarity between two web pages and how much it should contribute to the overall similarity between two sessions containing these two web pages. It uses the fraction of time spent at these

[2] The original ontology is available at http://www.cs.umd.edu/projects/plus/SHOE/cs.html and he modified taxonomy used in this study is also available at http://www.ceng.metu.edu.tr/~sefa/msthesis/ontology.dat.

Table 3 A sample cluster of sessions

	Step_1	Step_2	Step_3
S_1	o_1, o_2	o_1, o_2, o_3	
S_2	o_2, o_3, o_5	o_4, o_5, o_8	o_1, o_2, o_3
S_3	o_1, o_2	o_1, o_3	

pages. Given two web pages P_i and P_j from sessions S_i and S_j respectively, let the similarity component be denoted by $\mathcal{S}'$ which is the similarity between two concept sets describing two web pages P_i and P_j. The importance component $\mathcal{S}''$ is given by

$$\mathcal{S}'' = \left(\frac{T(P_i)}{T(S_i)} \times \frac{T(P_j)}{T(S_j)} \right)^{1/2} \tag{1}$$

where $T(P_i)$ is the time spent on page P_i and $T(S_i)$ is the total time spent on session S_i. The total similarity between two web pages $P_i \in S_i$ and $P_j \in S_j$ is given by

$$\mathcal{S}(P_i, P_j) = \mathcal{S}' \times \mathcal{S}'' \tag{2}$$

The motivation for the importance component $\mathcal{S}''$ is that if two pages are semantically close to each other but fraction of time spent on these pages are small in overall sequences, then these two pages should not contribute to overall similarity so much. Small $T(P_i)/T(S_i)$ means page P_i is not an important element in session S_i. Maybe, user just visited page P_i to access another page where page P_i has a link to it. On the other hand large value of $T(P_i)/T(S_i)$ means page P_i is important in session S_i and should be used in the measurement of similarity of two sessions S_i and S_j.

5 Clustering Navigations

After defining the measure of similarity between two sessions, we apply clustering [5, 7, 8] to get meaningful partitions of user sessions. The most widely used method for clustering is k-means algorithm. In k-means algorithm, clusters are represented with centroids. However, finding a centroid for sequence data is difficult. In [8], in order to find cluster centroids, objects in sessions are aggregated. Sequences are aligned and some objects are selected for centroid sequence at each step. A sample cluster containing 3 sequences is given in Table 3. The objects in step_1 have support values of o_1: 66 %, o_2: 100 %, o_3: 33 % and o_5: 33 %. With support threshold 50 %, objects selected for step_1 of centroid sequence are o_1 and o_3. For step_2 and step_3, by calculating support values and selecting ones with support values greater than 50 %, the centroid sequence for that cluster would be $\{o_1, o_2\} \rightarrow \{o_1, o_3\} \rightarrow \{o_1, o_2, o_3\}$.

We used CLUTO clustering software [4] which is freely available.[3] There are several algorithms available in CLUTO package [9]. We used *Direct k-way clustering* algorithm for extracting navigation patterns. It is similar to traditional k means clustering algorithm. As in k means, it iteratively refines the clustering until no change.

[3] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview.

6 Experimental Evaluation

The dataset used in this study is the server logs of Middle East Technical University, Department of Computer Engineering web site. [4] All accesses requested to the server are between 06 February 2011 and 18 February 2011. They were collected in Apache HTTP server combined log format. The total size of logs is 107 MB.

To evaluate the method, accuracy of recommendation of new web pages is used. To evaluate the similarity measure used to compute similarity between two sessions, recommendation (prediction) experiment was performed. Let $S = \texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{C} \rightarrow \texttt{D}$ be a session of length 4. For recommendation experiments the last item of the session (in this case page D) is removed from the session and it is tried to be predicted by using other sessions in the dataset. To compare session $S' = \texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{C}$ with other sessions in the dataset, the similarity measures with/without time information are used. At the end, if page D is in the set of recommendations, the recommendation is considered as accurate, otherwise not accurate.

For recommendation, k-nearest neighbor algorithm was used. Given a test session S, the last URL in S is removed for prediction. k sessions from training set that are most close to S (without last URL) are selected and the last URL of S is predicted based on k nearest neighbor sessions.

Example Let $S = \texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{C} \rightarrow \texttt{D}$ be the test session. The last item of it is removed from the session to be used for recommendation, so $S' = \texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{C}$ is mapped to concept set sequence and k most-similar sessions are selected from the training set. Let $k = 2$ and the closest sequences be $S_1 = \texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{E} \rightarrow \texttt{F}$ and $S_2 = \texttt{A} \rightarrow \texttt{C} \rightarrow \texttt{D}$. S' is aligned with S_1 and S_2 separately. The alignment of S' and S_1 is %beginalign* $\texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{C}, \texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{E} \rightarrow \texttt{F}$ The next web page of S' is predicted as page F. The alignment of S' and S_2 is $\texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{C}, \texttt{A} \rightarrow \texttt{-} \rightarrow \texttt{C} \rightarrow \texttt{D}$. The next web page of S' is predicted as page D. Therefore, the set of predictions as the next web page pf S' is {F, D}. Since the real next web page is D is in prediction set, the recommendation is considered as accurate.

The session similarity measure used in this study is compared with similarity measure considering URLs to assess similarity. According to this measure, the similarity between two web pages is 1.0 if their URLs are the same, 0.0 otherwise. This similarity measure (called URL-equality measure from this point) was compared with our similarity measures, which use URL mapping to concept tree and Rada et al.'s distance [6] with and without time-spent information.

For different k values recommendation accuracies are given in Fig. 1. Recommendation accuracy is the ratio of correct predictions to the sum of correct and false predictions. exp1 is the results by using URL-equality similarity measure. For smaller k values, the recommendation accuracy is very low (around 0.1). For larger k values, the recommendation accuracy increases to 0.2, since the number of recommendations is k and it is more likely to predict next web page true with more predictions. The line exp2 shows results of experiments by using our similarity measure *without*

[4] http://www.ceng.metu.edu.tr.

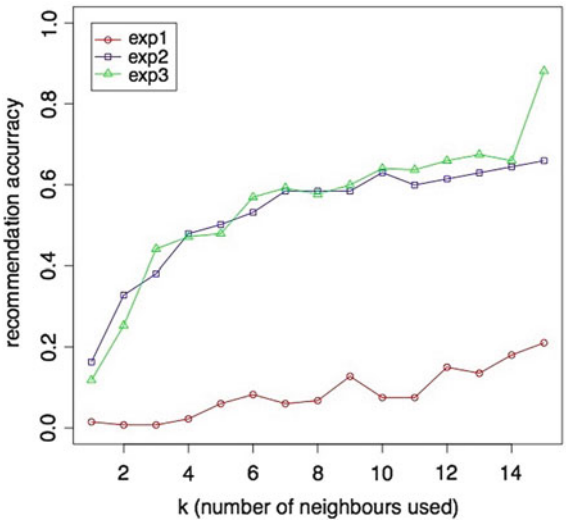


Fig. 1 Prediction accuracy rates for different similarity measures. `exp1` is the URL-equality similarity measure. `exp2` and `exp3` are results of our similarity measures *without* and *with* using time-spent information, respectively

Table 4 Means and standard deviations of prediction accuracies of three different similarity measures

Similarity measure	Mean	Stdev
URL-equality	0.09564	0.0243
Without time-spent info	0.6347	0.04002
With time-spent info	0.64003	0.042

using time-spent information. It is much more accurate than URL-equality similarity. For $k = 10$, the prediction accuracy is around 65 %. The last line, exp3 shows results of experiments using our similarity measure *with* using time-spent information. The results are very close to results of similarity measure without time-spent information. For $k = 10$, similarity with time-spent gives better results.

To find out whether performance differences of similarity measures are significant or not, t-test was performed. $k = 10$ is selected for t-test. Each experiment was run 50 times. First, URL-equality measure was compared with our similarity measure *without* time-spent information. $p = 2.49E^{-79}$ suggests that means of recommendation accuracies of these two similarity measures are significantly different. Second, the effect of using time-spent information was analyzed. Our similarity measures *with* and *without* using time-spent information were compared and p-value was 0.525 which shows that they are *not* significantly different. Means and standard deviations of prediction accuracies are given in Table 4. t-test results show that the experiments of our similarity measure have more accuracy than URL-equality measure. However, using time-spent information does not increase or decrease performance significantly.

7 Conclusion

In this work, a session clustering approach is presented in order to extract navigation patterns and to predict user's next page. In the proposed approach, using raw dataset of web page access logs, sessions of users are constructed and each web page on these sessions are mapped to set of concepts. Two different kinds of similarity measures are used: Rada et al.'s distance [6] to measure the similarity between two concepts, and average set similarity to measure the similarity between two sets of concepts. Then, Needleman–Wunsch dynamic programming algorithm is used to measure similarity between two sessions, which is used for clustering of sessions. Computed clusters represent groups of sessions of users with similar contents. These clusters can be used to understand the behavior of users. They can be used for recommendation of new pages to users during the web site navigation. The introduction of time-spent information to web page similarity measure is also analyzed. Our experiments show that the approach that we have used is quite effective and we have obtained high accuracy results in predicting/recommending the next potential page that will be visited by the user.

Acknowledgments This work is supported by grant number TUBITAK-109E282, TUBITAK.

References

1. Banerjee, A., Ghosh, J.: Clickstream clustering using weighted longest common subsequences. In: Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining, pp. 33–40 (2001)
2. Facca, F.M., Lanzi, P.L.: Mining interesting knowledge from weblogs: a survey. Data Knowl. Eng. **53**, 225–241 (2005). http://dl.acm.org/citation.cfm?id=1066378.1066379
3. Heflin, J., Hendler, J., Luke, S.: SHOE a knowledge representation language for Internet applications. Technical Report CS-TR-4078 (UMIACS TR-99-71), Department of Computer Science, University of Maryland (1999)
4. Karypis, G.: CLUTO—a clustering toolkit. Technical Report #02-017 (Nov 2003)
5. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on web usage mining. Commun. ACM **43**, 142–151 (2000). doi:10.1145/345124.345169
6. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Trans. Syst. Man Cybern. **19**, 17–30 (1989)
7. Srivastava, J., Cooley, R., Deshpande, M., Tan, P.N.: Web usage mining: discovery and applications of usage patterns from web data. SIGKDD Explor. Newsl. **1**, 12–23 (2000). doi:10.1145/846183.846188
8. Yilmaz, H., Senkul, P.: Using ontology and sequence information for extracting behavior patterns from web navigation logs. In: IEEE, ICDM Workshop on Semantic Aspects in Data Mining (SADM'10) (Dec 2010)
9. Zhao, Y., Karypis, G.: Empirical and theoretical comparisons of selected criterion functions for document clustering. Mach. Learn. **55**(3), 311–331 (2004)

A Model of Boot-up Storm Dynamics

Tülin Atmaca, Tadeusz Czachórski, Krzysztof Grochla, Tomasz Nycz and Ferhan Pekergin

Abstract The simultaneous boot of multiple virtual machines or networking devices imposes heavy load on the networking infrastructure. In large network of many virtual machines the coordination of the reboots and device registration is required. In the paper we present a simple analytical model computing the distribution of boot-up time, which is verified by comparison to a simulation model and provide analysis of the influence of several parameters on the overall boot-up time during a boot storm.

1 Introduction

Recently we observe a growing demand for virtualization services. The virtual machines are typically deployed in a datacenter, where a set of host computers runs a set of virtual machines, sharing the same hard disk array. In such environment

T. Atmaca
Institut Mines-Télécom/Télécom SudParis, 9 Rue Charles Fourier, 91000 Evry, France
e-mail: tulin.atmaca@it-sudparis.eu

T. Czachórski · K. Grochla (✉)
Institute of Theoretical and Applied Informatics, Polish Academy of Sciences,
Baltycka 5, 44–100 Gliwice, Poland
e-mail: kil@iitis.gliwice.pl

T. Czachórski
e-mail: tadek@iitis.gliwice.pl

T. Nycz
Institute of Informatics, Silesian University of Technology, ul. Akademicka 16,
44-100 Gliwice, Poland
e-mail: tomasz.nycz@polsl.pl

F. Pekergin
LIPN, Université Paris-Nord, 93430 Villetaneuse, France
e-mail: pekergin@lipn.univ-paris13.fr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_38,

the phenomena called "boot storms" has been observed [17] when multiple virtual machines are rebooted at the same time and the load imposed on the networking infrastructure, the disk arrays, the network management system and the host computer CPUs is very high, such that all machines cannot be served in the same time. The similar phenomena was observed in telecommunication networks, when a temporary power failure causes all networking devices in some areas to boot at the same time, and overloads the network management systems and the file servers distributing the firmware. Depending on the performance of the network, the other hardware resources and the number of machines booting simultaneously, the whole procedure may take up to few hours for a large number of virtual machines or networking devices. The estimation of the boot up time is crucial for the network administrator to estimate the network unavailability time after power failures or the time required for all the virtual machines to become fully operational after the reboot of the host computer. It is also important for the datacenter and network operators to work on the improvement of the system performance. The system performance can be monitored by the network management protocols (e.g. SNMP [12], netconf [3] or TR.069 [5]) and the dedicated software for management of the virtualization system (e.g. virt-tools [21] or VMware vCenter [22]).

Recent measurements give some information on the boot storm phenomena. The inputs/output operation during a boot of an isolated PC have been observed in order to determine the number of these operations and the volume of the data transmitted. These quantities are not constant even for identical machines running under the same operating system because of the differences existing between the applications deployed on them. Hence, the first remark is that the boot process depends on many factors. The number of IO operations and the volume of exchanged data increases for new technologies. However, new disk arrays can support more read-write operations in a shorter time. During a boot process, approximately 90 % of the data operations concern the read from the disk. The number of IO operations can vary from few dozen (30 or 40) to few hundreds. The transferred data volume is about 200–550 MB and approximately one quarter of it is to be written.

The boot storm problem has gained the research attention just recently—one of the first works that mention the term is [17]. However, a proposition on how to tackle the boot storms were proposed in [2] using a management module which simply block new boots when the system is overloaded. Hansen and Jul in [10] propose a distributed storage system designed specifically for virtualization workloads running in large-scale data centers and clouds that equalizes hard drive matrix load during boot storm. However up to now there is no mathematical model of the boot storm phenomena that would allow to evaluate the performance of different solutions proposed.

2 The Model

Let us assume that at the beginning there are N customers to be booted. The customer boot up time represents a request of a single, comparatively long service time which corresponds to loading the operating system from the hard drive disks (HDD) matrix.

An already working virtual machine imposes a periodic relatively low load on the HDD. The same model can also represent network devices registering and sending statistics to server.

The customers already booted go to the pool of registered customers. Each of them addresses periodically to the same server a demand of a short service of constant time. We model this system as a single server having two queues with non-preemptive priority discipline. The low priority queue is fed by customers to be booted and coming from the pool which has N customers at the beginning and 0 customers at the end of the process. The density of their service time (*registration* time) is $f_{reg}(x)$. The customers in priority queue come from the pool of booted customers having 0 customers at the beginning and N at the end of the process. The density of their service time, called *statistical* service is denoted by $f_{st}(x)$.

After each boot up time there is a busy period to serve priority customers already working and arrived during the last boot up time. If there are n customers already registered in the source and the sojourn time of each customer in the source has a distribution with mean $1/\lambda$ and variance σ_A^2, then the number of arrivals during a fixed time T is approximately normally distributed with mean $n\lambda T$ and variance $n\lambda^3\sigma_A^2 T = n\lambda C_A^2 T$, see e.g. [6]. For a variable time x distributed with density function $f_{reg}(x)$ we will compute the mean and variance of this normal distribution as

$$m_n = \int_0^\infty f_{reg}(x) n\lambda x dx$$

and

$$\sigma_n^2 = \int_0^\infty f_{reg}(x) n\lambda^3 \sigma_A^2 x dx$$

having thus

$$p_{n,i} = \frac{1}{\sigma_n\sqrt{2\pi}} e^{-\frac{(i-m_n)^2}{2\sigma_n^2}}$$

After the nth boot up there is the busy period started by periodic customers arriving during this boot up time (they are $n-1$ in the source), there is i arrivals with probability $p_{n-1,i}$. Denote pdf of this busy period by $f_{B\,n-1}(x)$. To determine it we use the diffusion approximation:, a method which is used in performance evaluation since several decades [6–9] for various purposes. Here, the duration of the busy period started with i customers is expressed by the first passage time from $x_0 = i$ to $x = 0$ by a diffusion process having the absorbing barrier at $x = 0$. The density function $\phi(x, t; x_0)$ of such a process is, see e.g. [4]

$$\phi(x, t; x_0) = \frac{e^{\frac{\beta}{\alpha}(x-x_0)-\frac{\beta^2}{2\alpha}t}}{\sqrt{2\pi\alpha t}} \left[e^{-\frac{(x-x_0)^2}{2\alpha t}} - e^{-\frac{(x+x_0)^2}{2\alpha t}} \right]. \quad (1)$$

The density function of the first passage time from $x = x_0$ to $x = 0$ is

$$\gamma_{x_0,0}(t) = \lim_{x\to 0}\left[\frac{\alpha}{2}\frac{\partial}{\partial x}\phi(x,t;x_0) - \beta\phi(x,t;x_0)\right] = \frac{x_0}{\sqrt{2\pi\alpha t^3}}e^{-\frac{(\beta t+1)^2}{2\alpha t}}. \quad (2)$$

$$\text{and} \qquad f_{Bn}(x) = \sum_{i=1}^{\infty} p_{n,i}\gamma_{i,0}(x),$$

or its Laplace transform is

$$\bar{f}_{Bn}(s) = \sum_{i=1}^{\infty} p_{n,i}\bar{\gamma}_{i,0}(s) \quad \text{where} \quad \bar{\gamma}_{i,0}(s) = e^{-i\frac{\beta+\sqrt{\beta^2+2\alpha s}}{\alpha}}. \quad (3)$$

The parameters α, β reflect the arrivals and service time first to moments: $\beta = n\lambda - \mu$ and $\alpha = n\lambda^3\sigma_A^2 + \mu^3\sigma_B^2 = n\lambda C_A^2 + \mu C_B^2$.

The total boot up time T, for $N > 2$, has pdf

$$f_T(x) = f_{reg}(x) \times f_{reg}(x) \times f_{B1}(x) \times f_{reg}(x) \times f_{B2}(x) \times \cdots \times f_{B\,N-2}(x) \times f_{reg}(x) \quad (4)$$

which is the N-fold convolution of $f_{reg}(x)$ with $f_{B1}(x)\ \ldots f_{B\,N-2}(x)$ densities. It may be done with the use of Laplace transform, i.e.

$$\bar{f}_T(s) = \left(\bar{f}_{reg}(s)\right)^N \bar{f}_{B1}(s)\bar{f}_{B2}(s)\ldots\bar{f}_{B\,N-2}(s). \quad (5)$$

3 Numerical Example

Assume the simultaneous boot up of $N = 100, 200, 500$ virtual machines. The mean boot up time is $1/\mu_{reg} = 2$ s and is distributed following 4th order Erlang distribution, hence each of four exponential phases has mean value $1/\mu = 1/4\mu_{reg} = 0.5$, and

$$\bar{f}_{reg}(s) = \left(\frac{2}{2+s}\right)^4$$

Let the mean service of statistical calls be $1/\mu_{st} = 0.001$ s or $1/\mu_{st} = 0.01$ s and be distributed following 8th order Erlang distribution. In the first case

$$\bar{f}_{st}(s) = \left(\frac{8000}{8000+s}\right)^8 \quad \text{or} \quad \bar{f}_{st}(s) = \left(\frac{800}{800+s}\right)^8$$

and the sojourn time in the source for statistical calls be exponentially distributed with $\lambda = 1$. If in the source are n processes, the parameters of the diffusion equations are e.g. in the first case $\beta = n\lambda - \mu = n - 1000$ and $\alpha = n\lambda^3\sigma_A^2 + \mu^3\sigma_B^2 = n + 125$.

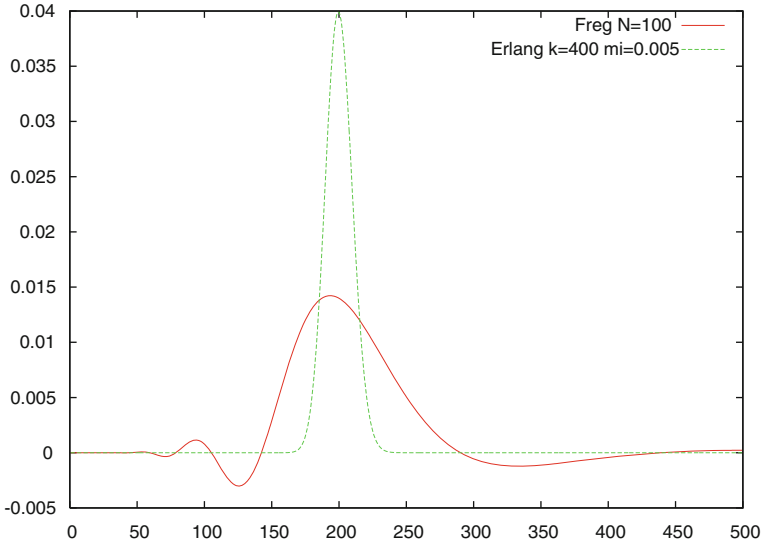


Fig. 1 Density function $f(x) = (2e^{-2x})^{*400}$ and the inverse of $\left(\frac{2}{s+2}\right)^{400}$ due to Stehfest algorithm

For many years we have used the Stehfest algorithm [15] in problems related to the diffusion approximation, e.g. when we inverse functions of the type (3) For any fixed argument x, the function $f(x)$ is there obtained from its transform $\bar{f}(s)$ as

$$f(x) = \frac{\ln 2}{2} \sum_{i=1}^{N} V_i \, \bar{f}\left(\frac{\ln 2}{x} i\right), \tag{6}$$

where

$$V_i = (-1)^{K/2+i} \sum_{k=\left\lfloor \frac{i+1}{2} \right\rfloor}^{\min(i,K/2)} \frac{k^{K/2+1}(2k)!}{(K/2-k)!k!(k-1)!(i-k)!(2k-i)!}. \tag{7}$$

However, we found that in case of multiple convolutions as in (4), (5) (if $N = 100$ these functions include Erlang distribution of 400th order) the results of Stehfest algorithm were totally unsatisfactory, see e.g. Fig. 1. After several trials we have chosen fixed-Talbot algorithm (FT), following the approach proposed by Talbot [19] and based on deforming the standard contour in the Bromwich integral

$$f(t) = \frac{1}{2\Pi i} \int_B \exp(ts) \bar{f}(s) ds.$$

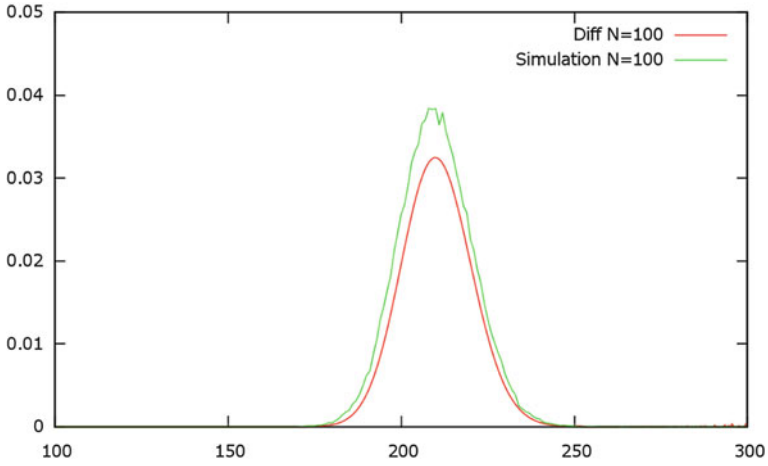


Fig. 2 Density function $f_T(x)$ for $N = 100$ boot ups, mean periodic service 0.001 s.—comparison of simulation and analytical–numerical results based on FT algorithm; axis x: time in seconds

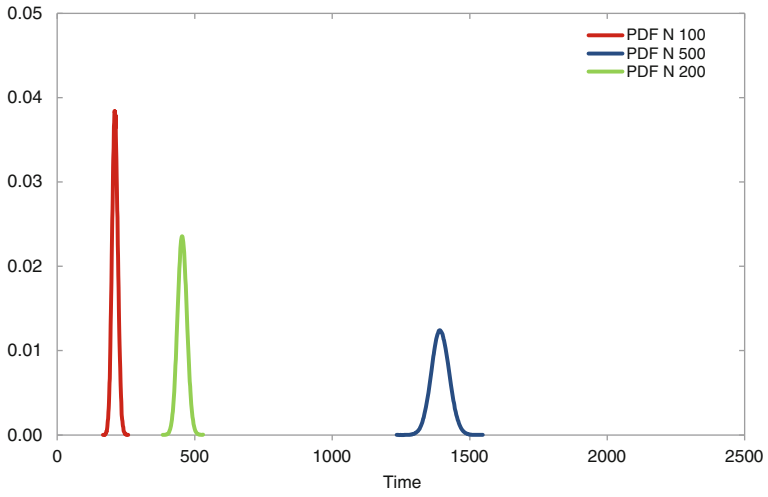


Fig. 3 Density function $f_T(x)$ for $N = 100, 200, 500$ boot ups, mean periodic service 0.001 s

Talbot's contribution is the carefully chosen form of the path B. The approach is extensively tested and compared to other inversion methods in [1]. Figure 2 presents the comparison of the density function $f_T(x)$ for $N = 100$ boot ups with mean periodic service 0.001 s. which is obtained by simulation and by our analytical-numerical approach with the use of FT algorithm. The order of errors observed for other curves displayed in Figs. 3–5 is similar.

Figure 3 displays the influence of the number N of computers to be registered on the boot-up storm distribution: as one may expect, the mean is proportional to

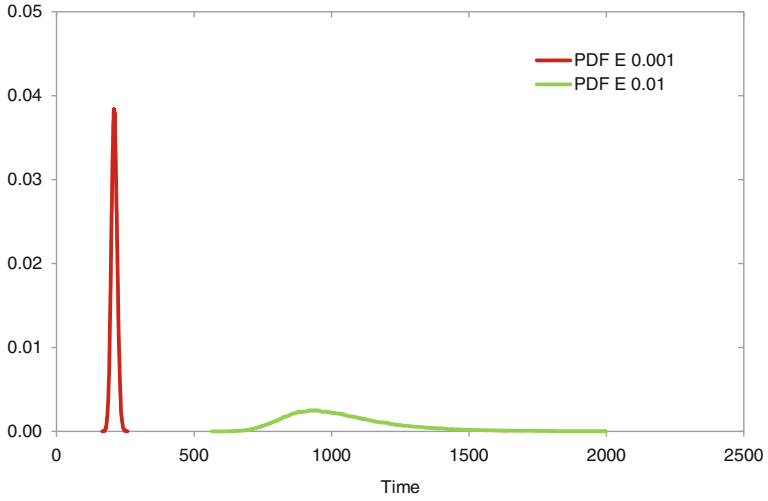


Fig. 4 Density function $f_T(x)$ for $N = 100$ boot ups, mean periodic service 0.001 and 0.01 s

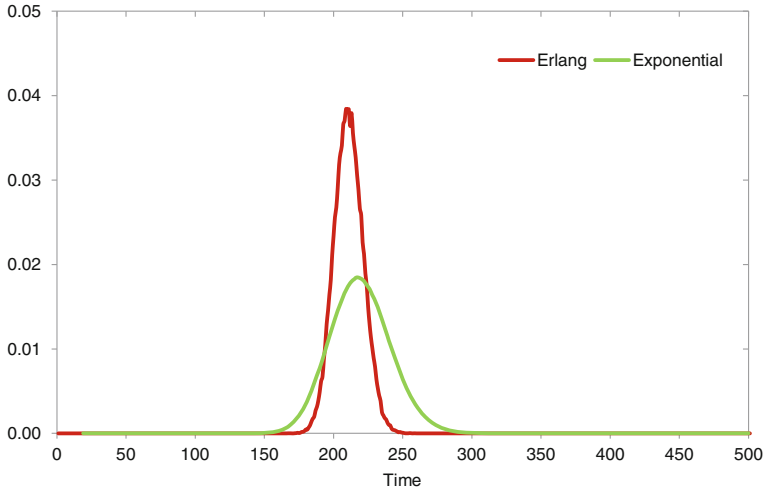


Fig. 5 Density function $f_T(x)$ for $N = 100$ boot ups, Erlang $k = 4$ and exponential distribution

N and the density function becomes more flat as N increases—the variation of the distribution is greater. Similarly, Fig. 4 illustrates the impact of the time needed to perform statistics—if it is larger, the boot up storm time becomes significantly larger (longer busy times after each registration) and also its variation is greater. Figure 5 shows the influence of the registeration time distribution on the ditribution of the boot up time—it compares $f_T(x)$ density functions for E_4 and exponential registeration time distributions.

4 Conclusions

The proposed analytical model is able to capture the main features of the boot-up storm period—it gives the probability density function of its duration—and we may investigate how the number of machines to be registered, the registration time and the activities of registered machines influence it. The model can be used to estimate the total time required to boot all virtual machines depending on changes in single boot up time and the performance of the hardware provided.

The model has been simplified to represent a single hardware resource which is the bottleneck of the boot up process. In future we want to extend it to catch the interactions between load of virtual machines on multiple resources, such as CPUs, memory and hard drive arrays by using multiple service nodes instead of a single one used in this work.

Acknowledgments This research was partially financed by a grant no. 4796/B/ T02/ 2011/40 of Polish National Council of Science (NCN).

References

1. Abate, J., Valko, P.P.: Multi-precision Laplace transform inversion. Int. J. Numer. Meth. Eng. **60**, 979–993 (2004)
2. Abbondanzio, A., et al.: System and method for prevention of boot storms in a computer network. US Patent 7,415,519, 2008
3. Bierman, A., Enns, R., Bjorklund, M., Schoenwaelder, J.: Network configuration protocol (NETCONF). http://tools.ietf.org/html/rfc6241
4. Cox, R.P., Miller, H.D.: The Theory of Stochastic Processes. Chapman and Hall, London (1965)
5. CPE WAN Management Protocol. TR-069 Amendment 4. Broadband Forum. July 2011. Retrieved February 16, 2012
6. Gelenbe, E.: On approximate computer systems models. J. ACM **22**(2), 261–263 (1975)
7. Gelenbe, E.: A diffusion model for packet travel time in a random multi-Hop medium. ACM Trans. Sens. Netw. **3**(2), 1–19 (2007)
8. Gelenbe, E.: Search in unknown random environments. Phys. Rev. E **82**, 061112 (2010)
9. Gelenbe, E., Pujolle, G.: The behaviour of a single queue in a general queueing network. Acta Informatica **7**(2), 123–136 (1976)
10. Hansen J.G., Jul, E.: Lithium: virtual machine storage for the Cloud. ACM SoCC 2010 in Indianapolis, Indiana
11. Kleinrock, L.: Queueing Systems, vol. II. Wiley, New York (1976)
12. McCloghrie, K., Schoenwaelder, J., Perkins, D.: Structure of management information version 2 (SMIv2). http://tools.ietf.org/html/rfc2578
13. Newell, G.F.: Applications of Queueing Theory. Chapman and Hall, London (1971)
14. OMNET ++ site. http://www.omnetpp.org
15. Stehfest, H.: Algorithm 368: numeric inversion of Laplace transform. Commun. ACM **13**(1), 47–49 (1970)
16. Stewart, W.J.: An Introduction to the Numerical Solution of Markov Chains. Princeton University Press, Princeton (1994)
17. Soundararajan, V., Anderson, J.M.: The impact of management operations on the virtualized datacenter. In: Proceedings of the 37th Annual International Symposium on Computer Architecture, pp. 326–337, New York (2010)

18. Stanley, D., Montemurro, M.P., Calhoun, P.R.: Control and provisioning of wireless access points (CAPWAP) protocol binding for IEEE 802.11. http://tools.ietf.org/html/rfc5416
19. Talbot, A.: The accurate numerical inversion of Laplace transforms. J. Inst. Math. Its Appl. **23**, 97–120 (1979)
20. Varga A.: The OMNeT++ discrete event simulation system. In: Proceedings of the European Simulation Multiconference (ESM'2001), Praga (2001)
21. Virt tools. http://virt-tools.org/
22. VMware vCenter. http://www.vmware.com/pl/products/datacenter-virtualization/vcenter-operations-management/overview.html

A Content Recommendation Framework Using Ontological User Profiles

Çağla Yaman and Nihan Kesim Çiçekli

Abstract In this work, a general purpose content recommendation framework using a hybrid recommendation algorithm is represented. Recommendation process is separated into different tasks; from collecting user data to representing the final recommendations. Separating these tasks to modules enables us to modify just one part of the process and compare the results. Its modular design is intended also to enable the same system be used for recommending different types of items and use different kinds of data sources. This flexibility is supported by the use of ontological user and content profiles. Different domain ontologies can be adopted to be used in the recommendation process. Items and properties of them are handled as simple ontology resources. That way, system works independent from what it is recommending as long as it has some properties that can be evaluated. So, the subject of recommendation and its properties can be in any complexity.

1 Introduction

Recommender systems simulate how people make recommendations and evaluate contents. A person may investigate contents himself and decide what aspects of these he likes or dislikes. He can either ask his friends if they liked it. He can prefer the most popular contents. Like humans, there are recommender systems that implements one of these methods or a combination of them.

These strategies let the researchers in the area to investigate different aspects of the problem and implement different approaches. In this work, we followed one

Ç. Yaman (✉) · N. K. Çiçekli
Middle East Technical University, Ankara, Turkey
e-mail: caglayaman@gmail.com

N. K. Çiçekli
e-mail: nihan@ceng.metu.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_39,

strategy which is making recommendation with hybrid approach where both user characteristics and common sense are considered.

In the following sections, the details of the work done are presented. Following section summarizes the research done so far that has influenced this work. The proposed recommendation framework is introduced later. Then, the implementation of an example movie recommendation system is presented. And finally, we conclude by giving a brief summary of the work done and present the findings from evaluations.

2 Background and Related Works

Parallel to the behaviours of people in real world, recommender systems are used in many different areas with many different degrees of complexity. Despite their differences, for all of them, finding the behavioural patterns in people's choices is essential. These patterns can be argued to derive from the characteristics of the objects and the person; or to be valid for more than one user. This duality has let researches to two different recommendation strategies: collaborative filtering (CF) and content-based filtering (CBF) [1]. Collaborative filtering handles the recommendation problem as a social act. Users similar to a user are referred as neighbours of that user [2] and neighbours are thought to be helpful for an individual's decision. In content-based filtering, the properties of the items rated determine the taste of the user. The items that resemble the ones high rated by the user are recommended.

Both methods have their strengths and weaknesses. CF requires a lot of users voting same items. However, users of the system might have evaluated different items which make it difficult to find similarities between users. This is called the sparsity problem [3, 4]. Cold-start problem [3] of CF systems refers to either the problem of a new item which is not rated by any of the users yet or a new user problem who has not rated many items. Also, some users may have rare tastes that does not resemble to any other. CBF is capable of evaluating an item that is not rated by any of the users. Therefore, CBF does not suffer from sparsity [3], but it requires the user has rated enough number of items to understand his preferences. Its success depends also on the significance of the information about items [4]. There is also a strong probability that the recommendations made will be too similar because it recommends items that suits the preferences of the user.

A more effective approach to recommendation problem is hybrid systems [5–11] which aim to eliminate specific drawbacks of both methods by combining them. Hybrid methods use CF and CBF together with different strategies as summarized in [5] and [4]. Some hybrid systems switch between CF and CBF under different circumstances; while some implement a staged process and execute both filtering methods sequentially on inputs. Some use two methods interleaved, by adding CF features into CBF or vice versa.

Hybrid recommendation models suggested in [9–11] generate weighted user profiles representing the preferences over the values of attributes of items. Recommendations are done using both opinions of similar users and these individual

user profiles. The use of semantic technologies improves the performance of a recommendation system providing more meaningful and compatible information about items and user profiles. Some researches use only ontologies to benefit from semantics of the domain studied [12] while some use more advanced semantic technologies like reasoning [3, 13]. In [14, 15], they try to improve the performance of CF methods by replacing the most widely-used user-based CF with item-based CF and using semantic similarities between items. The proposed system in [16] defines users' preferences ontologically also including the complex preferences representing more than one preference affecting each other.

In [5], a hybrid recommendation system using user profiling and semantic technologies such as ontologies and semantic spreading is implemented. User profiles consist of areas of interest which correspond to preferred concepts of domain together with their related concepts extracted by semantic spreading. The system clusters users according to their common concepts producing communities of interest.

3 Recommendation Framework

In this work, a content recommendation framework using user profiles has been developed. The framework implements a hybrid recommendation algorithm. This algorithm adds CBF features into CF stage and uses the output of the CF as input to the CBF stage as the recommendation strategy. The recommendation process is straightforward as three steps: generating the user profiles, clustering users, recommending contents.

Collecting required data This is the part of the process that has left mostly unimplemented. This is done intentionally to enable different kinds of data and data collecting mechanisms to be used in the system. Users that are realizing this framework should

- provide the *individual* node in the user ontology (see Fig. 2). No matter how complicated this preference can be, system will treat it as a single ontological resource.
- provide what does their *property* node matches in domain ontology and its connection to the *individual*.
- write queries to extract information from their domain.

The system can work with any dataset that fulfils that specific requirement: There are items, items have properties, properties have individual values.

Generating user profile The user profile ontology is model is as shown in Fig. 3. It consists of user history, which is the rating of the user to items and user preferences which is the content-based model of user choices.

We give a weight to each property of an item. If a user chooses a book looking at its author, the "author_property" has a high *weight*. The *property* node is the property resource in the domain ontology used. The *preference* represents the property together with its value. For example; when "director" is the *property*, "Oliver Stone" as the *individual* together with a certain *rating* is a *preference*.

Profile Generator generates the ontological content-based user profiles. It first gives the rating of an item to all of the individuals appearing in its description. Then, it separates these individuals to property vectors they are related to. There are multiple instances of one individual in these vectors since they may appear in more than one item. In each property vector, system counts the frequencies of the individuals with high ratings. The individual with the highest frequency is chosen to represent the vector. Then properties are weighted according to those individuals' frequency values. These become our weighted properties. Finally, for each individual, the average of the ratings are calculated and set as the final rating. Notice that, individuals are strongly related to properties; therefore one individual I, appearing as a value of one property P has a different rating than the same individual I appearing in a different property R.

Clustering profiles After the profiles are created, they are clustered so that the users that give similar weights to same properties and similar rating to individuals of these are clustered together. For simplicity, user profiles are reduced to their highest rated preferences of the highest weighted properties. And k-means algorithm is applied. In each iteration, cluster centroids are determine as the popular elements among the profiles.

Recommendation The recommender engine finds the high rated items of a cluster. Then, it compares the properties of each candidate item with the user's profile and estimates their ratings. The rating of a candidate is a positive function of the weight of that property and the ratings of its individuals. Candidates that have the highest ratings are recommended.

4 A Case Study: Movie Recommendation

In order to realize our framework, we used MovieLens[1] dataset as the source for user ratings and Freebase[2] to extract the descriptions of the movies. We used the first 500 users in MovieLens dataset. We divided the ratings into train and test datasets. Our training set contains maximum 30 ratings per user and test set contains 10 ratings per user.

The architecture proposed in Fig. 1 is adapted to movie recommendation task as shown in Fig. 2. The Profile Generator and Recommender Engine are remained unmodified and perform their standard tasks.

The recommendation performance is highly affected by the clustering procedure. Minimum number of preferences a user should have in common with the others in the cluster is determined as 50 %. Low values does not reflect a common taste whereas high values means users are too similar that they are not likely to recommend each other new and different items. In our tests, best result in terms of MAE is achieved when highest rated half of the properties in a profile are chosen. However,

[1] www.movielens.org

[2] www.freebase.com

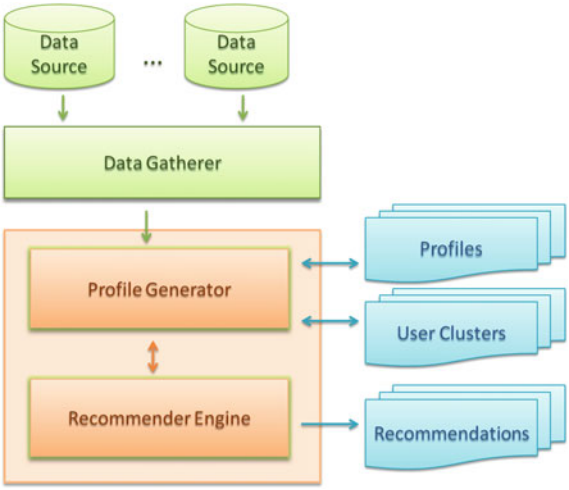


Fig. 1 Framework architecture

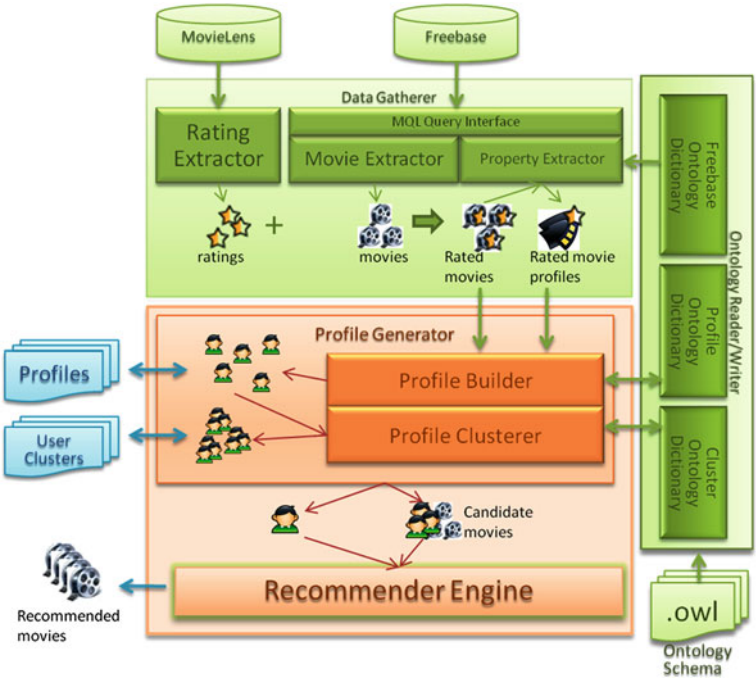


Fig. 2 Movie recommendation architecture

Fig. 3 User profile model

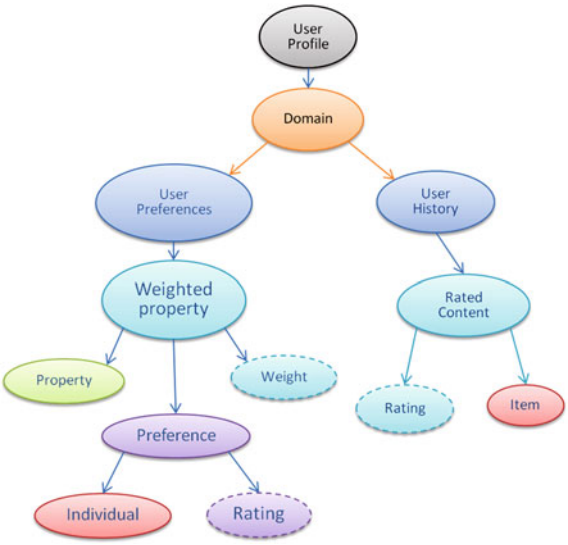


Table 1 UP-CBF performance

Precision	Recall	Total MAE	Recommendation MAE
0.85832	0.97786	0.65846	0.530693

the number of unsuccessful recommendations is higher than the case that contains only the top rated property. As the number of properties per profile increases, the features that are less important for a person is taken into consideration; so the system may conclude that some features are more important than they really are. On the other hand, because it clusters users using more preferences, it groups more similar users together increasing the recall. Recall and MAE improve by the number of users in a cluster increases while precision decreases. So the decision about this parameter can be made considering also the performance issues of the system (Fig. 3).

Four different filtering strategies that are used in evaluation are:

Pure Collaborative (CF): Only the ratings of the users are used.
CF + CBF staged process (CFCB): CBF is applied to the CF results.
User profile based CF (UP-CF): Users are clustered according to their CB profiles. Then CF is used for recommendation.
User profile based CF + CBF staged process (UP-CFCB): CBF is applied to the results of UP-CF.

Table 1 shows the performance of our hybrid recommendation strategy. We compared our hybrid strategy against pure content-based filtering as shown in Table 2. The improve in MAEs for different recommendation techniques and different datasets are shown by the percentage values. UP-CFCB performs better than pure CBF and CFCB. The hybrid filtering method used in this system shows better improvement over pure content-based filtering in terms of MAE than the related work [5] it was

Table 2 Comparison of different strategies

	CBF 100 users	CFCB 100 users 10 clusters	UP-CFCB 100 users 10 clusters
Recommendation MAE	0.40619	0.379515 6.5 %	0.305016 25 %

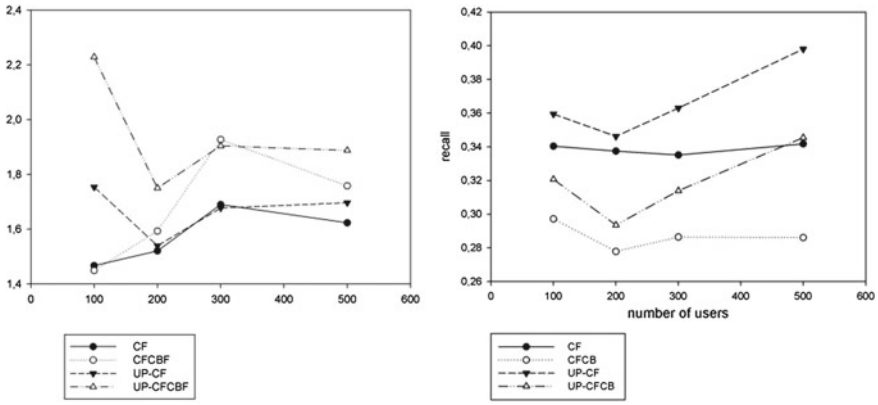


Fig. 4 Comparison of recommendation strategies by means of accuracy (*left*) and recall (*right*)

most influenced from. In [5], system shows approximately 15 % of improvement in MAE when 100 users are used with 75 % of them in training data and 25 % in test data. Under these conditions, our system shows 25 % improvement in the chosen case of 100 users with 10 clusters.

We observe that, as the more user is used, the performance of the system increases and CFCB begins to show closer performance to UP-CFCB. Therefore, we can conclude that the negative effect of the sparsity is reduced.

Finally, we applied different filtering strategies and observed the difference our proposed strategy makes. In tests, the neighbourhood size for clusters is fixed to 10. For UP-CF and UP-CBF, the similarity between cluster users are 50 %, number of weighted properties considered for each user is 2. The graphs in Fig. 4 shows that applying CBF after CF improves accuracy. However, it decreases the recall, because it reduces the number of recommended movies by applying a second filter. User profiling, on the other hand, improves the recall.

5 Conclusion

In this paper, a general-purpose hybrid recommendation framework is proposed. The framework consists of modules for user profiling, clustering and a recommendation engine, implementations of which can be directly used in realizing different

recommendation systems. The other characteristic of the work is the user profile structure. The generated user profiles reflect both the taste of the user and the aspects of the domain they give importance to. This improves the accuracy of the recommendations made and gives the power to explain the reasons of them.

We tested our system to understand the effects of design decisions we made. The clustering phase has a great impact on the recommendation performance. Generating not too specific cluster profiles but including the common preferences of a majority of the users in them results better clustering.

We observed that user profiling shows positive effect on recommendation. It eliminates the wrong suggestions and improves the accuracy. This system can be improved by adding mechanisms for more complex inferences and benefiting from semantic technologies such as semantic spreading.

Implementing different recommendation systems using different data sources based on the proposed framework, will test its flexibility and modularity. The user profiles can be enriched by adding different domain profiles. By default, the system will treat these profiles as if they are totally independent from each other. But, there may be some cases these profiles can benefit from some or all of the preferences defined in each other. The strategies for detecting these properties and using them in cross-domain recommendations are expected to increase the performance of the system.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. Knowl. Creation Diffus. Util. **17**, 734–749 (2005)
2. Vozalis, E., Margaritis, K.G.: Analysis of recommender systems algorithms. HERCMA (Hellenic European Research on Computer Mathematics and Its Applications), Athens, Greece, pp. 732–745 (2003).
3. Blanco-Fernandez, Y., Pazos-Arias, J.J., Gil-Solla, A., Ramos-Cabrer, M., López-Nores, M., García-Duque, J., Fernández-Vilas, A., Díaz-Redondo, R.P.: Exploiting synergies between semantic reasoning and personalization strategies in intelligent recommender systems: a case study. J. Syst. Softw. **81**, 2371–2385 (2008)
4. Burke, R.: Hybrid recommender systems: survey and experiments. User Model. User-Adap. Inter. **12**, 331–370 (2002)
5. Cantador, I., Bellogín, A., Castells, P.: A multilayer ontology-based hybrid recommen-dation model. AI Commun. **21**, 203–210 (2008)
6. Spiegel, S., Kunegis, J., Li, F.: Hydra: a hybrid recommender system. CIKM (Conference on Information and Knowledge Management) Workshop CNIKM (Complex Net-works in Information and Knowledge Management), pp. 75–80 (2009).
7. Fernández, Y.B., Arias, J.J., Nores, M.L., Solla, A.G., Cabrer, M.R.: AVATAR : an Improved solution for personalized TV based on semantic inference. IEEE Trans. Consum. Electron. **52**(1), 223–231 (2006)
8. Good, N., Schafer, J.B., Konstan, J.A., Borchers, A., Sarwar, B.M., Herlocker, J., Riedl, J.T.: Combining collaborative filtering with personal agents for better recommenda-tions. In: Conference of the American Association of Artifical, Intelligence AAAI-99, pp. 439–446 (1999).
9. Li, Q., Kim, B.M.: Constructing user profiles for collaborative recommender system. In: Proceedings of the Sixth Asia Pacific Web Conference, pp. 100–110 (2004).

10. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: Feature-weighted user model for recommender systems. In: Proceedings of 11th International Conference on User Modelling (UM 2007), vol. 4511, pp. 97–106 (2007).
11. Li, X., Murata, T.: A knowledge-based recommendation model utilizing formal concept analysis and association. In: Proceedings of ICCAE2010 (The second IEEE International Conference on Computer and Automation Engineering), 2010.
12. Wang, Y., Stash, N., Aroyo, L., Hollink, L., Schreiber, G.: Using semantic relations for content-based recommender systems in cultural heritage. Workshop on Ontology Patterns, p. 16 (2009).
13. Heim, P., Lohmann, S., Stegemann, T.: Interactive relationship discovery via the semantic web. In: Proceedings of the 7th Extended Semantic Web Conference (ESWC2010), 2010.
14. Mobasher, B., Jin, X., Zhou, Y.: Semantically enhanced collaborative filtering on the web. Lecture Notes in Computer Science. Springer, Heidelberg (2004)
15. Mylonas, P., Andreou, G., Karpouzis, K.: A collaborative filtering approach to personalized interactive entertainment using MPEG-21. In: Maglogiannis, I., Karpouzis, K., Wallace, M. (eds.) Proceeding of the 2007 Conference on Emerging Artificial intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in Ehealth, Hci, Information Retrieval and Pervasive Technologies, vol. 160, pp. 173–191. IOS Press, Amsterdam (2007)
16. Weiß, D., Scheuerer, J., Wenleder, M., Erk, A., Gülbahar, M., Linnhoff-popien, C.: A user profile-based personalization system for digital multimedia content. In: Proceedings of the Digital Interactive Media in Entertainment and Arts (DIMEA), pp. 281–288 (2008).

Part X
Data Engineering

Data Consistency as a Service (DCaaS)

Islam Elgedawy

Abstract Ensuring data consistency over partitioned distributed database systems is a classical problem. Classical solutions proposed to solve this problem are mainly adopting locking or blocking techniques to ensure data correctness. These techniques are not suitable for cloud environments as they produce terrible response times due to the long latency and faultiness of Wide Area Network (WAN) connections among cloud datacenters. To overcome this problem, this paper proposes an inventory-like approach for ensuring data consistency over WAN connections that minimizes the number of exchanged messages over the WAN to enhance response times. As maintaining data consistency is a costly process, we propose to use different levels of data consistency for data objects, as not all data objects have the same importance. Hence, strong consistency will be used only for data objects that are crucial to the correctness of application operations. To save application developers from the hassles of maintaining data consistency, we propose to have a new platform service (i.e. Data Consistency as a Service (DCaaS)) that developers invoke to handle their data access requests fulfilling their different consistency requirements. Experiments show that proposed data consistency approach realized by the DCaaS service provides much better response time when compared with classical locking and blocking techniques.

1 Introduction

Real-life cloud environments usually constituted from a collection of datacenters connected via a WAN. A datacenter is constituted from thousands of machines connected via a LAN (i.e. local area network). Latency in WANs is much bigger than latency in LANs. This difference in latency distribution inside cloud environments

I. Elgedawy (✉)
Computer Engineering Department, Middle East Technical University,
Northern Cyprus Campus, Guzelyurt, Mersin 10, Turkey
e-mail: Elgedawy@metu.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_40,

created a non-homogenous timing model for the cloud. For example, latency between two machines inside a datacenter is in the range of 100 ms; while latency between two machines connected via WAN is in the range of 1000 ms (i.e. when machines are in different continents). This means if an object changes its value in a given datacenter, such change cannot be instantly recognized by other datacenters due to long propagation delay of WAN connections. Hence, existing classical DB concurrency control and transaction management approaches opt to accommodate to the slowest latency inside the cloud, leading to bad services performance. Therefore, we identify WAN connections as the main bottleneck in cloud environments. Indeed ensuring data consistency over partitioned distributed database system is a classical problem that attracted many researchers. However, classical solutions proposed to solve this problem (as the ones described in [4, 5]) are mainly adopting locking or blocking techniques to ensure data correctness. Such classical approaches adopt a pessimistic strategy that assumes conflicts occur frequently. Hence, they suspend all other instances from working (via locking or blocking) when a given instance needs to do some updates for the shared data. These techniques provide very bad performance when applied on cloud environments [6, 7], as they tend to create considerably high overhead over the slow faulty WAN connections due to exchanged synchronization messages and performed reconciliation transactions, which of course badly hurts services availability and customers' response times. Recent approaches known as NoSQL databases [8] (such as Amazon's Dynamo and Apache Cassandra) proposed to go for weaker forms of consistency on the clouds such as eventual consistency [9], in which they trade correctness for availability, that all SaaS (i.e. Software as a Service [1, 2, 10]) service instances are allowed to work normally without any suspension and process their transactions locally. This is known as the optimistic strategy; as it assumes conflict occur rarely. However, when a conflict is detected undo transactions and/or compensating transactions should be performed by the SaaS services, also in some cases some data versions could be lost. Such optimistic approaches require such cases to be handled by the SaaS business logic, which of course, creates big headache for SaaS developers. We argue that SaaS developers should not be handling such cases in their code, and they should be totally decoupled from such problems. Hence, we propose to use a new platform service for handling data consistency issues (i.e. Data Consistency as a Service (DCaaS)) to decouple SaaS developers from managing those issues in their code. SaaS developers will only need to invoke DCaaS service operations in their code to perform the required data access operations, and the DCaaS service will take care of data consistency management issues. To ensure strong data consistency on the clouds, this paper proposes a new breed of DB services that takes into consideration the non-homogenous nature of the cloud timing model. The paper proposes a novel for ensuring global data correctness by guaranteeing strong data consistency on eventually consistent distributed data stores using an inventory-like approach. Such approach requires service providers to divide crucial objects capacity among that DCaaS services instances such that each DCaaS service instance makes sure its incoming users requests do not consume more than its allocated quota. Hence, when data is replicated between data stores no conflicts could arise. When a given DCaaS service instance requires more than

assigned quota due to high volume of requests, it could contact other DCaaS instances to borrow extra quota. If quota borrowing process fails the request is rejected. The proposed approach adopts a lazy replication approach for objects synchronization. Quota allocation is performed in a hierarchal manner that a quota allocated to a given node will be automatically distributed over its children using any specified allocation policy. Our hierarchy is constituted from a cloud as the root, which constituted from datacenters (a.k.a. cloudlets). Each datacenter contains data stores, which could be managed by multiple DCaaS service instances. For example, a flight reservation service provider could allocate each datacenter a quota of seats, such quota will be distributed over its data stores according to a given allocation policy (for example, equally distributed). Each data store quota will be distributed among its DCaaS service instances. Each DCaaS service instance makes sure that reservation requests on a given data store do not exceed its quota. As maintaining strong data consistency is a costly process, we argue that it should be only used for objects that their correctness is crucial for SaaS services. That for important and sensitive data we use strong consistency, while for less important data we could go for weaker consistency notions such as eventual or session consistency [9]. Hence, we propose to allow SaaS services to adopt different data consistency levels for handling their data by defining a Data Consistency Plan (DCP) that indicates the chosen consistency level for each data object. Such plan is submitted to DCaaS services for execution. SaaS providers could automatically change their DCPs on run time without changing their SaaS code, and DCaaS services will automatically redistribute objects quota according to the new plans. Experiments show that proposed DCaaS service adopting the proposed data consistency approach provides much better response time when compared with classical locking and blocking techniques. The rest of the paper is organized as follows. Section 2 introduces the proposed data consistency approach. Section 3, briefly discusses various design aspects of the proposed DCaaS service. Section 4, provides some basic comparative simulation experiments for proposed approaches, and finally Sect. 5 concludes the paper.

2 Ensuring Data Consistency on the Clouds

One way to improve performance on the clouds is to avoid communication via WAN connection by restricting group of users to a given data center (a.k.a. cloudlet) and avoid data sharing between data centers. This creates a smaller variation of cloud computing, known as "cloudlet computing" that has a homogeneous timing model. This computing paradigm ensures local correctness of the data but cannot ensure global correctness of the data, as conflicts might appear when data is replicated between cloudlets. As we are not sure when or where conflicts could occur, this created data uncertainty on cloud environments due lack of global control over the distributed data. Such uncertainty must handled by the SaaS service as part of its business logic. As we argued before, we should decouple SaaS developers from such problems; hence managing data uncertainty should be handled by data consistency approaches

encapsulated in the proposed DCaaS service. In business, handling uncertainty is a fact of life, hence there are many solutions adopted by businesses for managing uncertainty such as reserved inventory, allocations against credit lines, and budgeting. We propose to handle uncertainty for data objects using similar strategies. For example, an airline reservation service could have its database partitioned among many cloudlets. Instead of globally locking flight data object whenever a booking operation is made, we will allocate a quota of chairs for each cloudlet such that each cloudlet locally ensures the correctness of its quota using its own concurrency control approach, hence there will be no conflicts when replication occurs between cloudlets. A cloudlet local concurrency control could use any locking or blocking technique without hurting performance as latency inside cloudlets is small (i.e. within the range of 100 ms), which still provide acceptable response time (more details in Sect. 4). Another example, in banking, if we need to access a given account, instead of locking the account object, we could allocate a budget for each cloudlet to manage incoming withdrawal and deposit requests. The problem now, what would happen if a given request cannot be fulfilled by its local quota or budget? Simply, we will try to borrow quota from other cloudlets that still have unused quota. A DCaaS service instance could borrow from instances located in its cloudlet, or from instances in other cloudlets. As borrowing from outside cloudlets requires communications via WAN connection, only requests requiring extra quota will be affected. Hence, we argue that quota should be distributed between cloudlets in a manner that minimizes the borrowing rate. The quota borrowing protocol works as follows. The DCaaS service instance requesting the quota becomes the leader of the process, and sends its request first to DCaaS instances located inside its datacenter with the required of borrow amount. Each DCaaS service instance received the quota borrow request replies back with the quota amount it can transfer. This amount ranges from zero to the required amount. The leader collects all quota transfer responses and acknowledges other DCaaS instances with the amounts it will take from them. Once a DCaaS instance receives such acknowledgment it reduces its share of quota with the acknowledged amount. Of course, the easiest quota distribution strategy is to equally divide the object capacity among datacenters. However, the proper quota distribution strategy should be based on thorough demand forecast analysis. In case a service provider makes a mistake in allocating the quotas, DCaaS service instances will automatically redistribute the quotas among themselves when requests arrives via the quota borrowing process. It is important to note that the price of wrong quota allocation is longer response times due to the slow quota borrowing process (if WAN connections are used). However, once quota borrowing process is finished, response times dramatically improve, as all incoming requests will be handled locally inside the datacenter. Whenever, a DCaaS service instance is cloned inside a cloudlet to improve performance, cloudlet quota has to be redistributed among available DCaaS service instances. To achieve such goals, we propose different protocols for quota borrowing, object stabilization, and DCaaS fault tolerance in order to ensure the safety and liveness properties of the proposed approach. Due to the limited space, in this paper we will provide a quick overview on those issues and describe these protocols in more details in other publications.

3 Data Consistency as a Service (DCaaS)

We assume that SaaS service database is partitioned among different cloudlets. Each cloudlet is cloning and partitioning its database portion and provides access to it via a PaaS (i.e. Platform as a Service [1, 2, 10]) service. Each cloudlet could create multiple SaaS, PaaS and DCaaS service instances to increase availability, throughput, and to enhance response time. Each SaaS instance handles a user-base of SaaS customers. As SaaS and DCaaS services could be deployed on multiple cloudlets, we require cloudlets to adopt lazy replication protocol among them to replicate their data; such replication process is implemented as a background process so that it cannot affect DCaaS instances access for the PaaS service. Finally, we require each DCaaS instance to keep reference to other DCaaS instances created inside and outside its cloudlet. SaaS developers should invoke DCaaS operations for accessing SaaS database. This means SaaS developers will not have SQL statements in their code; instead they will have invocations for the DCaaS service operations to access their databases. We require the DCaaS service to implement a simple API interface for reading and writing operations. The read operation API is *Read (DataObject X)*, while the write operation API is *write (DataObject X, ObjectValue V)*. For example, *DCaaS.Write (X,1)* is an invocation for the DCaaS write operation to update a value of an object. DCaaS invokes the write operation version corresponding to the data consistency level defined in the corresponding DCP. In case of the strong consistency requirements, we apply the inventory-like approach proposed in Sect. 2. In case of eventual consistency, reads and writes are handled locally using the adopted concurrency control protocol between DCaaS instances and changes are replicated to other cloudlets in a lazy manner. In case of conflicts, we apply Thomas's write rule (i.e. last write wins). In case of session consistency, reads occurs once from DB through PaaS, then any consecutive reads and writes are accessed from the cache only. We encapsulate each data consistency approach as a *component service* to decouple DCaaS code from the actual approach implementation. Each component service communicates with the PaaS service to perform the required operations on the database. We can think of the DCaaS service as the orchestrator for service components. With the help of DCaaS, the SaaS service code is totally decoupled from the adopted data consistency approaches.

3.1 Data Consistency Plan

We require each SaaS provider to define a DCP for its service; hence each SaaS service instance will follow the same service DCP. Currently, we design DCaaS service to support three levels of data consistency (i.e. Strong, Eventual, and Session). Strong consistency implies that the global correctness of the data object is maintained such that any SaaS instance accessing the object is actually reading its up-to-date correct value. Eventual consistency implies that object correctness

is locally maintained (i.e. within its cloudlet) but not globally. However, if there is no conflicts between cloudlets, and no more no new updates are made to the object, eventually all accesses will return the last updated value due to the lazy replication process. Session consistency implies that the SaaS instances read its own writes only. This means object data will be maintained only at the DCaaS cache and does not go to the PaaS service for storage. Hence, those data will be lost after the session terminates. A DCP indicates the required consistency level for each data object. Also it indicates the required stabilization method to be applied in case of object values divergence. DCP also specifies the cloudlet quota for strong consistency objects. We formally define a DCP as a set of Object Access Patterns (OAP) that $DCP = \{OAP(i)\}$, where an $OAP(i) =< i, c, s, q >, i$ is the data object reference, c is the required consistency level, s is the required stabilization method, and q is the cloudlet quota distribution plan and it is defined as a set of cloudlet quota allocations, that $q = \{< \textit{Cloudlet reference}, \textit{CloudletQuota} >\}$. As the number of cloudlets is always small, the size of such quota list is not a problem. We support different stabilization methods such as Thomas rule and basic uncertainty filters (i.e. Min, Max, Avg, and Sum). We require each data object to have only one OAP. For example, a SaaS provider for a flight reservation service X, which require access for two data objects *Customer* and *Flight*, The corresponding DCP could be defined as $DCP(X) = \{< \textit{Customer}, \textit{Eventual}, \textit{Thomas}, \{\} >, < \textit{Flight}, \textit{Strong}, \textit{MAX}, \{< 1, 50 >, < 2, 200 >\} >\}$. This means the consistency of the customer object is eventual and Thomas write rule (i.e. last write wins) will be applied in case of conflicts, while the consistency of the flight object is strong, and maximum value method will applied in case of conflicts. It also shows that we have two cloudlets, the first cloudlet has a quota of 50, and the second one has a quota of 200. To provide flexibility for SaaS providers, we provide them with the option to change their DCPs at run time whenever they like and the DCaaS service will do the necessary adjustments to fulfill the new requirements. However due to space limitation, we will discuss DCP change management in other publications.

3.2 DCaaS Recovery

In case of DCaaS instance failure, its SaaS instance will time out, then connect to another DCaaS instance and resubmit its requests. We will not face a problem for session and eventual consistency objects, as the values will be directly fetched from the data store when requests are resubmitted. However, the problem will be in the strong consistency objects, as the allocated quota for strong consistency objects has to be redistributed among remaining DCaaS instances. Hence, we require remaining DCaaS instances to elect a leader that calculates the new quota for each DCaaS instance, and inform each DCaaS instance with the new quota. DCaaS leader does not compute quota for all strong data objects at the same time, but it computes quotas on demand, hence standard log and cache management processes should be applied, as the cache of any DCaaS instance is limited. When a DCaaS instance

Table 1 Experiments results

Approach	Average (ms)	Minimum (ms)	Maximum (ms)
Locking approach	1600	1038	2242
DCaaS with no quota borrow	200	150	258
DCaaS with 10 % quota borrow	350	249	465
DCaaS with 50 % quota borrow	600	414	815

recovers from failure, it contacts the current leader to redistribute the quote again. If there is no current leader, the newly created or recovered DCaaS instance broadcasts its existence to other DCaaS instances in its cloudlet, and when acknowledged, a leader election could take place. Of course, many optimization procedures could be performed to minimize the overhead of quota redistribution. More details about quota redistribution protocol, and it corresponding optimizations will be discussed in other publications due to space limitation.

4 Experiments

We performed basic simulation experiments using the cloudsim [3] tool that enables us to simulate cloud environments. For simplicity, we assumed that we have only one DCaaS instance per cloudlet; we have two identical cloudlets (i.e. datacenters) with WAN connection of 500 ms, and one user-base accessing the first cloudlet with latency 50 ms. This user-base generates 1000 request per hour, and each request contains one strong read and one strong write operations. Cloudsim assumes optimum conditions for WAN connection (i.e. no faults), hence the experiments results do not reflect real life environment, as real life values are expected to be much worse. We simulated both cloudlets with 5 virtual machines each. Each virtual machine contains 512 MB and 1 KB bandwidth. Each cloudlet is build using two 4-core processors identical servers with 10000 MIPS, 200 GB RAM, 10 TB storage, and IMB bandwidth. We run the simulation for period of 1 day and computed the average, minimum and maximum response time for the whole user-base. In our experiments, we compare between the pure locking approach that locks record on both cloudlets for every request, against the proposed DCaaS service approach when adopting the proposed data consistency approach with quota borrowing rates are 0, 10, and 50 %. Experiments results are listed in Table 1.

As we can see, when the quota is enough, the DCaaS instance does not need to communicate with the other DCaaS instance through the WAN, as all requests are fulfilled within the cloudlet; hence response time was drastically improved. However, when it needs to borrow quota from the other cloudlet response time is increased as WAN connection is used. As we notice that response time increases when the quota borrowing percentage increases. Hence, quota distribution among cloudlets is crucial for minimizing response times, that cloudlets with higher demand rates should get higher percentages of the quota.

5 Conclusion

In this paper, we argued that strong consistency requirements should be adopted only for data objects crucial for application correctness, otherwise weaker forms of data consistency should be adopted. Such different consistency requirements are defined in the form of a Data Consistency Plan (DCP), which is submitted to the proposed DCaaS (i.e. Data Consistency as a Service) platform service that makes sure the defined DCPs are automatically fulfilled. We also proposed an inventory-like approach for ensuring strong data consistency on eventually consistent cloud data stores. Experiments show that proposed approaches provide much better response time when compared with locking and blocking techniques. In this paper, we do not allow different consistency levels for the same data objects; however, in future work we are planning to relax this condition such that the same data object could have different consistency levels depending on the performed SaaS operations and customers' contexts.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: Above the clouds: a Berkeley view of cloud computing. Technical Report EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009
2. Buyya, R., Broberg, J., Goscinski, A.: Cloud Computing: Principles and Paradigms. Wiley, New York (2011). ISBN-13: 978–0470887998
3. CloudSim. http://www.cloudbus.org/cloudsim/
4. Davidson, S.B., Garcia-Molina, H., Skeen, D.: Consistency in partitioned networks. ACM Comput. Surv. **17**(3), 341–370 (1985)
5. Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H., Swinehart, D., Terry, D.: Epidemic algorithms for replicated database maintenance. In: Proceedings of ACM Conference on Principles of Distributed Computing (1987)
6. Gray, J., Helland, P., O'Nei, P.L., Shasha, D.: The dangers of replication and a solution. In: Proceedings of ACMSIGMOD International Conference on Management of Data, pp. 173–182 (1996)
7. Kraska, T., Hentschel, M., Alonso, G., Kossmann, D., Consistency Rationing in the cloud: Pay only when it matters. In: Proceedings of the International Conference on Very Large Data, Bases (2009)
8. NoSQL Databases. http://nosql-database.org/links.html
9. Vogels, W.: Eventually consistent. ACM Queue **6**(6), 14–19 (2008)
10. Wei Y, Blake B.M.: Service-oriented computing and cloud computing: challenges and opportunities. IEEE Internet Comput. **14**(6), 72–75 (2010)

Heuristic Algorithms for Fragment Allocation in a Distributed Database System

Umut Tosun, Tansel Dokeroglu and Ahmet Cosar

Abstract Communication costs caused by remote access and retrieval of table fragments accessed by queries is the main part execution cost of the distributed database queries. Data Allocation algorithms try to minimize this cost by assigning fragments at or near the sites they may be needed. *Data Allocation Problem* (*DAP*) is known to be NP-Hard and this makes heuristic algorithms desirable for solving this problem. In this study, we design a model based on *Quadratic Assignment Problem* (*QAP*) for the *DAP*. The *QAP* is a well-known problem that has been applied to different problems successfully. We develop a set of heuristic algorithms and compare them with each other through experiments and determine the most efficient one for solving the *DAP* in distributed databases.

Keywords Distributed database design · Fragmentation · Heuristics.

1 Introduction

Although distributed databases (*DDBs*) are attractive for very large datasets, their utilization brings new problems. Designing a *DDB* is one of the most complicated problems in this domain. In addition to the classical centralized database design, fragmentation and data allocation are the new two problems to tackle with [1, 2]. Data allocation problem (*DAP*) is an optimization problem with constraints [3]. Disk drive speed, parallelism of the queries, network traffic, load balancing of servers should be considered during the design. Even without most of these decision parameters

U. Tosun · T. Dokeroglu · A. Cosar (✉)
METU Computer Engineering Department, Ankara, Turkey
e-mail: cosar@ceng.metu.edu.tr

T. Dokeroglu
e-mail: tansel@ceng.metu.edu.tr

U. Tosun
e-mail: tosun@ceng.metu.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_41,

it is clear that *DAP* is an NP-Hard problem. File allocation problem (*FAP*), *DAP*, and *QAP* have some similarities. From the perspective of data transmission, *DAP* is the same with *FAP*. However, the logical and semantic relations of fragments differ from these two problems. *QAP* has more similarities with *DAP* that it also keeps track of the resource locality. The *QAP* is first presented with Koopmans and Beckman [4]. In Sect. 2, we give a brief information about the related works for *FAP*, *DAP*, and QAP. Section 3 explains the design of the solution. Section 4 gives the proposed algorithms and Sect. 5 gives experimental environment and the results respectively. The conclusions are presented in Sect. 6.

2 Related Work

FAP, *DAP*, and *QAP* are extensively studied well-known problems [4, 5]. There are static or dynamic allocation algorithms for the *DAP*. Static algorithms use predefined requirements, whereas dynamic algorithms take modifications into consideration [6]. Ceri and Plagatti presented a greedy algorithm for replicated and non-replicated data allocation design [7]. Bell showed that the *DAP* is NP-Hard [8]. Corcoran and Hale [9] and Frieder and Siegelmann [10] solved the *DAP* with genetic algorithms (*GA*). Ahmad and Karlapalem solved the problem of non-redundant data allocation of fragments in distributed database systems by developing a query driven data allocation approach integrating the query execution strategy with the formulation of the data allocation problem [11]. Adl and Rankoohi addressed prominent issues of non-replicated data allocation in distributed database systems with memory capacity constraint [12]. They took into consideration the query optimization and the integrity enforcement mechanisms in the formulation of the *DAP*. Many other NP-Hard problems are designed by using *QAP* [13–15].

3 Solution Formulation with the *QAP*

The data allocation problem is specified with two kinds of dependencies between transactions and fragments as seen in Fig. 1. The dependency of sites to fragments can be inferred from transaction-fragment and site-transaction dependencies. S represents sites, F represents fragments, T represents transactions in Fig. 1. Similarly, freq is the number of requests for the execution of a transaction at a site, trfr is the direct dependency of a transaction on a fragment, and q is the indirect dependency of a transaction on two fragments.

We formulated the cost function of the data allocation problem as the sum of two costs, direct and indirect transaction-fragment dependencies [12]. The dependency between a transaction t and a fragment f is called direct if there is a data transmission from the site containing f for each execution of t. If there is some data to be transferred from a site different from the originating site of transaction, the dependency is considered as indirect. The total cost of data allocation *Cst* is the sum of two costs *Cst*1 and *Cst*2 (Eq. 1).

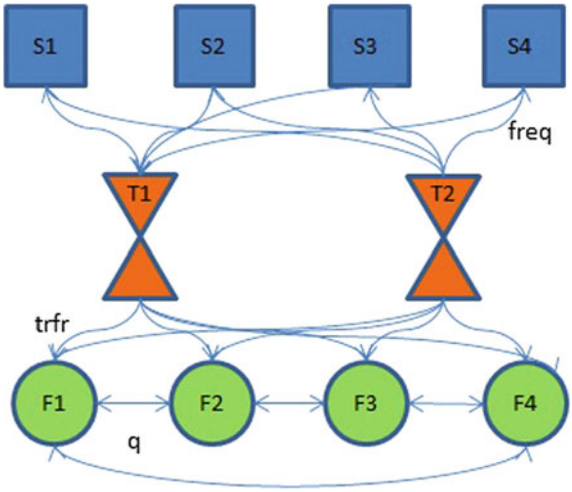


Fig. 1 The dependencies of transactions on fragments and sites on transactions

$$Cst(\Phi) = Cst1(\Phi) + Cst2(\Phi) \tag{1}$$

In Eq. 1, Φ represents the m element vector where Φ_j specifies the site to which f_j is allocated. $Cst1$ is represented by the amount of site-fragment dependencies. It is expressed by the multiplication of two matrices *STFR* and *UC*, where *STFR* stores the site fragment dependencies and *UC* stores the unit communication cost among the sites. The cost of storing a fragment f_j in site s_i is represented by partial cost matrix $PCST1_{n\times m}$. The unit partial cost matrix is formulated in Eq. 2.

$$pcst1_{ij} = \sum_{q=1}^{n} uc_{iq} \times stfr_{qj} \tag{2}$$

$Cst1$ can be represented as in Eq. 3 after having calculated the unit partial cost $pcst1_{ij}$ for each i and j.

$$Cst1(\Phi) = \sum_{j=1}^{m} pcst1_{\Phi_j j} \tag{3}$$

The inter-fragment dependency matrix *FRDEP* is defined as the multiplication of the matrices $QFR_{l\times m\times m}$ and $Q_{l\times m\times m}$. The matrix *QFR* denotes the execution frequencies of the transactions. The *FRDEP* matrix representing the inter-fragment dependency is the multiplication of the matrix *QFR* with the matrix *Q*. *Q* represents the indirect transaction fragment dependency. Equation 4 formulates the indirect transaction-fragment dependency $Cst2$. $Cst2$ is a form of the *QAP* as seen in the equation.

$$Cst2(\Phi) = \sum_{j_1=1}^{m} \sum_{j_2=1}^{m} frdep_{j_1 j_2} \times uc_{\Phi_{j1} \Phi_{j2}} \quad (4)$$

4 Proposed Algorithms for the Data Allocation Problem

GAs randomly generate an initial population of solutions, then by applying selection, crossover, and mutation operations repetitively, creates new generations [16]. The individual having the best fitness value is returned as the best solution of the problem [17]. *PMX* crossover is used in the GA. *PMX* copies a random segment from parent1 to the first child. It looks for the elements in that segment of parent2 that have not been copied starting from the initial crossover point. For each of these elements, say i, it looks in the offspring to see what element j has been copied in its place from parent1, *PMX* places i into the position occupied by j in parent2, since we know that we will not be putting j there. If the place occupied by j in parent2 has already been filled in the offspring by k, we put i in the position occupied by k in parent2. The rest of the offspring can be filled from parent2. The second child is created similarly [18].

Dorigo and colleagues proposed *ACO* as a method for solving difficult combinatorial problems [19]. *ACO* is a metaheuristic inspired by the behavior of real ants where individuals cooperate through self-organization. *FANT* is a method to incorporate diversification and intensification strategies [20]. It systematically reinforces the attractiveness of values corresponding to the best solution found so far the search, and on the other hand by clears the memory while giving less weight to the best solution if the process appears to be stagnating.

Metropolis developed a method by simulating the thermodynamic energy level changes in 1953 [21]. With this method, particles exhibit energy levels maximizing the thermodynamic entropy at a given temperature value. The average energy level is proportional to the temperature. This method is called Simulated Annealing (*SA*). Kirkpatrick applied *SA* on computer related problems in 1983 [22]. Many scientists have applied it to different optimization problems since then [23]. If a metal cools down slowly, it turns into a smooth metal because its molecules have entered a crystal structure. This crystal structure shows the minimum energy state, for an optimization problem. If a metal cools down too fast, the metal turns into a rough piece with bumps. These bumps and jagged edges show the local minimums and maximums. In *SA*, each point of the search space represents a state of the physical system, and the function to be minimized is the internal energy of the system in that state. The goal of the algorithm is to bring the system from an initial state to a state with the minimum possible energy.

5 Experimental Setup and Test Results

5.1 Experimental Environment

In each test, one parameter is varied while the others are fixed. The algorithms are tested by the same test data. Data is generated with rules defined in Sect. 5.2. Experiments are performed using a 2.21 GHz AMD Athlon (TM) 64 × 2 dual processor with 2 GB RAM and MS Windows 7 (TM) operating system. Each processing node has 102 buffers, and page size is 10,240 bytes, disk I/O time is 10 ms (per page), available memory is assumed to be sufficient to perform all join operations in main memory and each table is loaded into memory only once. Test data is generated according to the experimental environment of Adl [12]. The only difference is that we choose the unit costs in range [0,1]. Our test data generator gets number of fragments m, number of sites n and other parameters as input and creates a random *DAP* instance. We choose the fragment size randomly from the range $[\frac{c}{10}, 20 \times \frac{c}{10}]$, where c is a number between 10 and 1,000. We choose the site capacities in $[1, 2 \times \frac{m}{n} - 1]$. The sum of the site capacities should be equal to total fragment size m, where n is the total number of sites. We assumed that the number of sites n is equal to number of fragments m. Each fragment size is chosen randomly. We selected the unit transmission costs as a random number in range [0,1]. We generate a random probability of request for each transaction (*RT*) for each transaction to be requested at a site. Transaction fragment dependency is also represented with probability of access for each fragment (*AF*). The site fragment frequency matrix *FREQ* is determined as the multiplication of probability *RT* and a random frequency in range [1, 1,000]. Transaction fragment dependency matrix is generated as the multiplication of *AF* and a uniformly distributed random value in $[0, f_j]$ where f_j is the jth fragment. Finally the site fragment dependency matrix *STFR* is equal to *FREQ* × *TRFR*. We define the inter-fragment dependency matrix *FRDEP* as multiplication of the matrices $QFR_{l \times m \times m}$ and $Q_{l \times m \times m}$ where *QFR* takes into account the execution frequencies of the transactions and Q represents the indirect transaction fragment dependency.

5.2 Experimental Results

We performed several tests over *GA* to set the appropriate parameters. *GA* uses population size 1,000 and number of generations 200. We used the Fast Ant System [20] with parameter $R = 5$ for managing traces and number of iterations as 20,000. *SA* uses 100,000 as number of iterations and 2,750 as number of runs. After completing the experiments on instances ranging from size 5 to 100, it is concluded that *FANT* performs better than *GA* and *SA*. *FANT* executes faster than *GA* and *SA* on all instances as seen in Table 1, Figs. 2 and 3.

Table 1 Comparison of algorithms on *DAP* instances (cost value is column$\times 10^6$) unit

	Cost	Cost	Cost	*DAP* size	Time (s)	Time (s)	Time (s)
DAP size	*ACO*	*GA*	*SA*		*ACO*	*GA*	*SA*
5	0.04	0.04	0.04	5	9.26	76.27	130.29
10	0.31	0.32	0.31	10	14.52	87.80	143.84
15	0.98	0.99	0.98	15	13.74	90.76	214.02
20	2.61	2.63	2.61	20	17.91	123.79	243.30
25	5.19	5.25	5.19	25	25.86	131.98	351.23
30	10.27	10.39	10.27	30	31.17	132.46	461.89
35	16.39	16.64	16.41	35	43.31	150.06	393.73
40	25.91	26.28	26.02	40	56.59	166.80	420.65
45	37.28	37.73	37.40	45	80.92	191.93	437.74
50	53.93	54.76	54.08	50	105.33	471.98	511.40
55	71.30	72.72	71.40	55	126.00	268.31	516.86
60	90.35	91.76	90.50	60	166.55	315.31	828.14
65	112.31	113.59	112.49	65	204.35	421.93	1,090.77
70	146.41	148.48	146.73	70	320.62	536.15	1,303.21
75	177.90	180.04	178.16	75	309.51	609.77	976.97
80	219.40	223.10	219.81	80	396.18	464.17	1,234.48
85	262.24	267.04	262.89	85	807.43	532.05	898.11
90	316.11	320.88	316.81	90	621.55	563.15	1,336.74
95	370.14	375.49	371.14	95	725.93	629.55	1,128.08
100	428.40	436.19	429.10	100	1,203.99	1,236.30	1,389.19

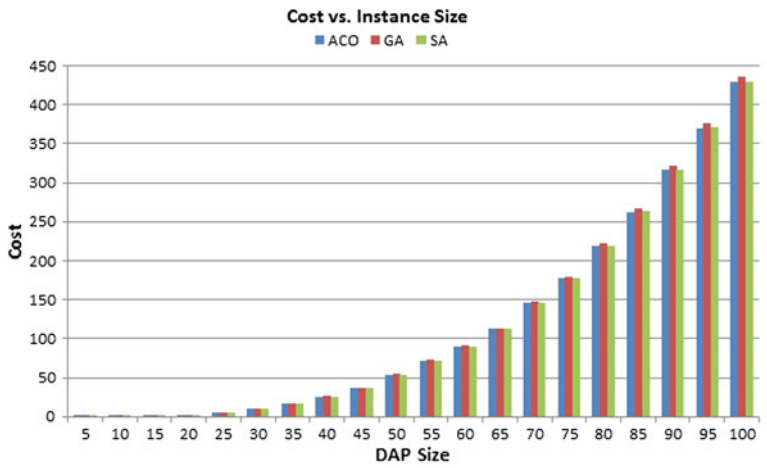


Fig. 2 Cost versus instance size comparisons of the algorithms

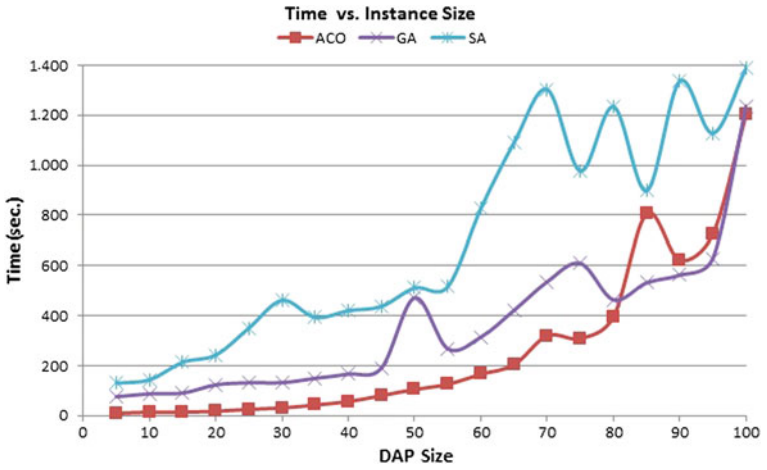


Fig. 3 Time versus instance size comparisons of the algorithms

6 Conclusions and Future Work

We solve the fragment allocation problem in distributed databases by making use of the well known *Quadratic Assignment Problem* solution algorithms. A new set of Genetic, Simulated Annealing, and Fast Ant Colony algorithms are introduced for solving this important problem. In the experiments, the execution times and the quality of the fragment allocation alternatives are investigated. The results are very promising even for very large number of fragments and sites. The model used for deciding the sites where each fragment will be allocated assigns only one fragment to each site. Replication of fragments to multiple sites and assigning multiple fragments to any site have not been considered in this work. As future work, we plan to eliminate these restrictions and develop algorithms that can produce any distributed database schema by allowing replication and horizontal/vertical fragmentation.

References

1. Ozsu, M.T., Valduriez, P.: Principles of Distributed Database Systems, 3rd edn, pp. 245–293. Springer (2011)
2. Dokeroglu, T., Cosar, A.: Dynamic programming with ant colony optimization metaheuristic for the optimization of distributed database queries. In: Proceedings of the 26th International Symposium on Computer and Information Sciences (ISCIS), London, Sept 2011
3. Lee, Z., Su, S., Lee, C.: A heuristic genetic algorithm for solving resource allocation problems. Knowl. Inf. Syst. **5**(4), 503–511 (2003)
4. Koopmans, T.C., Beckmann, M.J.: Assignment problems and the location of economics activities. Econometrica **25**, 53–76 (1957)

5. Laning, L.J., Leonard, M.S.: File allocation in a distributed computer communication network. IEEE Trans. Comput. **32**(3), 232–244 (1983)
6. Gu, X., Lin, W.: Practically realizable efficient data allocation and replication strategies for distributed databases with buffer constraints. IEEE Trans. Parallel Distrib. Syst. **17**(9), 1001–1013 (2006)
7. Ceri, S., Pelagatti, G.: Distributed Databases Principles and Systems. McGraw-Hill, New York (1984)
8. Bell, D.A.: Difficult data placement problems. Comput. J. **27**(4), 315–320 (1984)
9. Corcoran, A.L., Hale, J.: A genetic algorithm for fragment allocation in a distributed database system. In: Proceedings of the 1994 ACM Symposium on Applied Computing (SAC 94), pp. 247–250. Phoenix (1994)
10. Frieder, O., Siegelmann, H.T.: Multiprocessor document allocation: a genetic algorithm approach. IEEE Trans. Knowl. Data Eng. **9**(4), 640–642 (1997)
11. Ahmad, I., Karlapalem, K.: Evolutionary algorithms for allocating data in distributed database systems. Distrib. Parallel Databases **11**, 5–32 (2002)
12. Adl, R.K., Rankoohi, S.M.T.R.: A new ant colony optimization based algorithm for data allocation problem in distributed databases. knowl. inf. syst. **25**(1), 349–372 (2009)
13. Dokeroglu, T., Tosun, U., Cosar, A.: Parallel mutation operator for the quadratic assignment problem. In: Proceedings of WIVACE, Italian Workshop on Artificial Life and Evolutionary Computation, Parma, Feb 2012
14. Mamaghani, A.S., Mahi, M., Meybodi, M.R., Moghaddam, M.M.: A novel evolutionary algorithm for solving static data allocation problem in distributed database systems, In: Second International Conference on Network Applications, Protocols and Services, Reviews Booklet, Brussels (2010)
15. Lim, M.H., Yuan, Y., Omatu, S.: Efficient genetic algorithms using simple genes exchange local search policy for the quadratic assignment problem. Comput. Optim. Appl. **15**(3), 249–268 (2000)
16. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
17. Sevinc, E., Cosar, A.: An evolutionary genetic algorithm for optimization of distributed database queries. Comput. J. **54**(5), 717–725 (2011)
18. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
19. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. Part B **26**(1), 29 (1996)
20. Taillard, E.D., Gambardella, L.M., Gendreau, M., Potvin, J.Y.: Adaptive memory programming: a unifed view of meta-heuristics. EURO XVI Conference tutorial and research (1998)
21. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. **21**(6), 1087 (1953)
22. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
23. He, X., Gu, Z., Zhu, Y.: Task allocation and optimization of distributed embedded systems with simulated annealing and geometric programming. Comput. J. **53**(7), 1071–1091 (2010)

Integrating Semantic Tagging with Popularity-Based Page Rank for Next Page Prediction

Banu Deniz Gunel and Pinar Senkul

Abstract In this work, we present a next page prediction method that is based on semantic classification of Web pages supported with Popularity based Page Rank (PPR) technique. As the first step, we use a model that basically uses Web page URLs in order to classify Web pages semantically. By using this semantic information, next page is predicted according to the semantic similarity of Web pages. At this point, we augment the technique with Popularity based Page Rank (PPR) values of each Web page. PPR is a type of Page Rank algorithm that is biased with page visit duration, frequency of page visits and the size of the Web page. The accuracy of the proposed method is tested with a set of experiments in comparison to that of two similar approaches in the literature.

Keywords Next page prediction · Recommendation · Web usage mining · Page Rank algorithm · Semantic tagging · Semantic similarity

1 Introduction

Navigations of users are useful information resources for recommending new pages. These recommendations are usually specialized in predicting the next page of user. In the literature various techniques have been used in order to analyze the Web logs [2–5, 7] for next page prediction. One of the recent approaches for predicting user's next page navigation is using Page Rank [11], in which in-links of a page's popularity determines the popularity of that page. At this point, popularity can be defined in many different ways. Popularity based Page Rank (PPR) algorithm [4],

B. D. Gunel · P. Senkul (✉)
METU Computer Engineering Department , 06800 Ankara, Turkey
e-mail: senkul@ceng.metu.edu.tr

B. D. Gunel
e-mail: deniz.yanik@ceng.metu.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_42,

which is a variation of Page Rank, focuses on both page visit duration, size and frequency of pages. Using Web usage mining and content mining together is another approach employed in several studies in the literature [9, 10]. In [8], it is observed that semantic classification of Web pages only with their URLs for predicting next page is not always accurate. Therefore, it is advised to apply classification to Web pages with URLs under the support of other Web mining techniques.

In our work, we develop a next page prediction technique that basically calculates the semantic similarity of Web pages and uses this similarity result for next page prediction under the support of Popularity based Page Rank (PPR) values of the pages. We analyze only Web URL's content and tag each URL with Web site's domain related concepts. In next page prediction, we use this semantic classification in order to find pages that are conceptually similar to a given Web URL. In this approach, the semantic tagging based model uses Popularity based Page Rank (PPR) as a support argument. In other words, semantic similarity is used as the basic method for next page prediction and when two candidate page's conceptual similarity is equal, then PPR is used for additional information. We call this approach Semantic Tagging (ST) approach for next page prediction.

2 Related Work

The Page Rank algorithm [2] uses the link structure of pages for finding the most important pages with respect to the search result. There are models that bias Page Rank algorithm with other type of Web usage data, structural data or Web contents. In [3], Usage based Page Rank algorithm is introduced as the rank distribution of pages depending on the frequency value of transitions and pages. They model a localized version of ranking directed graph. In [5], they modify Page Rank algorithm with considering only the time spent by the user on the related page. With Popularity based Page Rank (PPR) [4], a model is developed such that it is basically bias the Page Rank algorithm with duration of the page visit, size of the page and frequency of the page with local and global modeling.

There are also studies that combines two or more web mining techniques in order to improve the generated recommendations. Haveliwala et al. [6] presents a model, which is a combination of semantic information related to Web pages with page rank algorithm. In the offline process they calculate page rank values of different concepts, in the online process they support search results with these rank values. In [9], authors cluster usage profiles and content groups concurrently. Afterwards, profile and content groups are integrated in order to support the next page prediction process. Similarly in [12], Web server logs are extended with content information, which is called *c-logs* and *c-logs* are used for producing associative rules related to content of pages and user navigations.

3 Background

3.1 Page Rank Algorithm

Page Rank algorithm [11] models the whole Web as a directed graph where the nodes are Web pages and the edges are page visits. Link structure of pages for determining the importance (rank value) of pages are used in this approach. In this algorithm, it is stated that if a page has some important in-links to it then its out-links to other pages also become important. In other words if a page is important then pages that it points to are also important. Therefore the algorithm propagates in-links of pages and if the in-links' total is high then the rank value of it is also high.

Basic calculation of Page Rank algorithm is given in Eq. 1. $IN(v)$ represents the in-links of page v, $OUT(v)$ is the out-links of page v, $|OUT_v|$ list the number of out-links of page v and WS is the number of the Web page set that includes all pages in the Web site.

$$PR(u) = \frac{(1-\epsilon)}{WS} + \epsilon * \sum_{v \in IN(u)} \frac{PR(v)}{|OUT_v|} \tag{1}$$

While calculating page rank values, especially for large systems, iterative calculation method is used. In this method, the calculation is implemented with cycles. In the very first cycle, all page rank values are assigned to a constant value such as 1, and with each iteration of calculation, the rank value becomes normalized within approximately 50 iterations under $\epsilon = 0.85$.

3.2 Popularity Based Page Rank

Popularity based Page Rank (PPR) [4] calculation, given in Eq. 2, is modeled in terms of transition popularity and page popularity of the pages that point to (in-links of) the page that is under consideration. In the equation, $IN(x_j)$ is the set that keeps the in-links of that page.

$$PPR_i = \epsilon * \sum_{x_j \in IN(x_i)} \left[PPR_j * TransitionP_{j \to i}\right] + (1-\epsilon) * PageP_i \tag{2}$$

In the equation above, rank distribution of pages in the model depends on the popularity of a page (*PageP*) and transitions (*TransitionP*) that point to that page. In this model, *popularity* is defined in two dimensions. The first one is the page dimension and second one is the page visit transition dimension. For both dimensions popularity is defined in terms of time user spends on page, size of page and visit frequency of page. Calculations of the model are constructed by using coefficients in a different form for assigning rank values to pages than traditional page rank

distribution that assigns equal rank values to all in-links of a page. In popularity calculation, page and transition popularity are calculated separately but in a similar way. Page popularity is needed for calculating random surfer jumping behavior of the user and transition popularity is needed for calculating the normal navigating behavior of the user. However the main idea is common for finding popularity for nodes and edges. The details of the formula can be found in [4].

3.3 Semantic Annotation of Web Pages

Web page classification can be performed in terms of Web page content and from Web URLs. In semantic annotation, semantic terms extracted from the content or URL of a page are mapped to the web page. In [13], Web page classification is introduced as an extension of text classification that uses html pages' content and also Web URLs. Semantic annotation takes advantage of the semi-structured Web page content. In addition to text content, HTML tags and XML markups carry information with which one can infer logical information about Web pages. In Web mining, semantic annotation techniques are heavily used in order to support next page prediction. In general the annotation process can be divided into two main phases. The first step is to determine semantic terms and relations between them (rules, hierarchies etc.). The second step includes constructing the model that includes the semantic terms and mappings of them to Web pages.

4 Semantic Tagging Based Next Page Prediction

In our work, we analyze Web URLs in a semantic way in order to obtain useful information for next page prediction of users. In our work, we tag every URL with a specific concept in a concept hierarchy. In order to obtain the concept similarity of web pages, we define a special concept similarity equation.

$$ConceptSim(P_1, P_2) = \sum_{1 \leq n \leq 3} Sim(S_1, S_2, n) \tag{3}$$

In this equation, we assign each concept level a different weight for measuring the similarity value of each page's conceptual information. In this weight assignment, the main idea is to assign more detailed level higher weight value in order to increase the cumulative concept similarity value. For each level of hierarchy we assign weights with logarithmic distribution starting with 2. In other words, for the first level of detail we assign 2 (λ_1), for the second level of detail we assign 4 (λ_2) and for the third level of detail we assign 8 (λ_3).

In Eq. 3, concept similarity is defined by measuring three levels of detailed information related to Web page's URL, where P_1 and P_2 are the pages to be compared for concept similarity. S_1 and S_2 are the concept sets related to P_1 and P_2, respectively

Table 1 Example URLs

Pages	URLs
P_1	*/people/faculty/john/index.php*
P_2	*/~john/ceng302/FurtherDep.ppt*
P_3	*/~mary/ceng302/Btrees.ppt*

Table 2 Concept similarity example

Pages	1st level concept	2nd level concept	3rd level concept
P_1	John	Faculty	People
P_2	Course	Ceng 302	John
P_3	Course	Ceng 302	Mary

where $1 \leq x,\ y \leq n$ where x,y $\in$ concept levels and n is the current calculated concept level.

$$Sim(S_1, S_2, n) = \begin{cases} \lambda_n & \text{if } \exists\ S_1[x],\quad S_2[y] \mid S_1[x] = S_2[y] \\ 0 & \text{if } \forall\ S_1[x],\quad S_1[x] \mid S_1[x] \neq S_2[y] \end{cases}$$

In semantic tagging process, the first step is to capture the concepts embedded on each Web page URL. After capturing the semantic terms, detail level of each concept should be determined. Following this, we save each level of concept for calculating the similarity of URLs considering concepts later in an online process of recommendation. As the first step of capturing semantic terms from URLs, we investigate the structure of Web URLs. In each URL, we extract some rules related to each valid value of URLs. The constraints and assumptions considered during semantic tagging process are as follows: In our methodology, we consider a 3-level of concept hierarchy. However, in some cases, it is hard to capture 3-level concepts. But as a rule of thumb, at least one concept is captured related to URLs. Concepts are captured from left to right on the URL text, starting from Level 3 to Level 1. Level 3 has the least and Level 1 has the most detailed information about the Web pages. In the calculation of the similarity, since in some cases the captured semantic terms are less than 3, the similarity is searched from the 1st Level to 3rd Level orderly. In comparison, we accept the highest value of coefficients in comparison of different level of concepts. Although it is a manual process, it has a systematic working.

In Table 1, a set of sample URLs are given, in order to illustrate semantic tagging and semantic similarly calculation. In Table 2, each Web page and its related concepts for each level can be seen. As an example, similarity calculation for P_1 and P_2 is as follows.

1. Level, $Sim(ConceptSet_2, ConceptSet_3, 1) = 0$
2. Level, $Sim(ConceptSet_2, ConceptSet_3, 2) = 0$
3. Level, $Sim(ConceptSet_2, ConceptSet_3, 3)\lambda_3 = 8$ [1]

[1] The similarity value is calculated from common *John* tagging in P1 and P2.

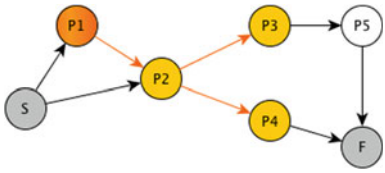


Fig. 1 Directed graph for Next Page Prediction

Table 3 PPR values and page similarities for sample sessions

Page	Popularity based page rank	P1 semantic similarity comparison
P1	0.45500	14
P2	0.141182	4
P3	0.00080	8
P4	0.048265	4
P5	0.00323	0

Final concept similarity values of each Web page pair is as follows:

- $ConceptSim(P_1, P_2) = 8$
- $ConceptSim(P_2, P_3) = 6$
- $ConceptSim(P_1, P_3) = 0$

For predicting the next page, a recommendation set is constructed under the proposed algorithm. The main idea behind predicting next page is to produce recommendations from directed graph that is designed from sessions in Web server logs. In the directed graph, for a certain depth, pages are listed and sorted in descending order by calculated rank values. Hence the next page prediction method can be seen as Markov model that is supported by page similarity values of pages instead of probabilities. This model can be seen as 1st order Markov model that has a page rank value base.

Consider the example navigation graph given in Fig. 1. In this example, if a user visits page P1, the recommendation set for depth 2 includes P2, P3 and P4 pages, and they should be sorted in descending order with respect to page similarity values and PPRs.

Assume that, results are already calculated for Popularity based Page Rank (PPR) and semantic similarity and given in Table 3 as an example. With these values, recommendation set is sorted as {P3, P2, P4}. In that point, semantic similarity values are calculated by comparing semantic similarity of the user's current visit with candidate Web pages.

5 Experimental Evaluation

In the experiments, we analyze and compare the accuracy performance of Usage based Page Rank (UPR), Popularity based Page Rank (PPR) and Semantic Tagging (ST) approaches. In our experiments basically we use the evaluation method

Table 4 Accuracy results with Ksim and Osim

		Ksim	Osim
Top 2	UPR	0.58	0.53
	PPR	0.65	0.59
	ST	0.88	0.68
Top 4	UPR	0.64	0.45
	PPR	0.68	0.45
	ST	0.81	0.53
Top 8	UPR	0.64	0.25
	PPR	0.69	0.31
	ST	0.80	0.36

employed in [3]. In addition, we consider extra evaluation methods. We make experiments with top-2, top-4 and top-8 recommendation comparisons that are measured by *Ksim* and *Osim* similarities. Moreover, with top-8 recommendations, precision and recall values are also calculated.

In our experimental evaluations, we use METU Computer Engineering Department's[2] Web server logs from 29/May/2010 to 18/Feb/2011. In the raw data there are 5.168.361 unique Web page URLs. We use 3-fold cross validation in order to support the independence of the data. For cleaning the data, frequency pruning [1] is used since working with frequent pages produce more stable results in general. The frequency limit of pages are set to 10. After cleaning less frequent data in training set, we have approximately 6000 pages in total and in test data set we have 2000 pages.

After the cleaning process, test data set contains approximately 110 sessions and training data set contains about 225 sessions. In these sessions, a directed Web graph of test data is produced in order to obtain real transition values and to compare them with the predicted ones. In every evaluation, one page is picked in the directed graph that have 2 or more nodes that it points to. Then for that page, every algorithm produces recommendation sets as candidate pages for visiting after that page.

After preprocessing, the formulas for Usage based Page Rank (UPR) and Popularity based Page Rank (PPR) values, given in Sect. 3 are calculated under $\epsilon = 0.85$, which results in 0.15 jumping factor and 50 iterations in calculation of page rank values of each page.

In comparing the predictions with the real page visits, we use two similarity algorithms that are commonly preferred for finding similarities of two sets. The first one is called *Osim* [6] algorithm, which calculates the similarity of two sets without considering the ordering of the elements in the set. It focuses on the number of common elements of two sets with a limit value. The limit value can be seen as the top-n next page recommendations for a visited page.

In Table 4, accuracy results of *Ksim* and *Osim* values are given for each of the approach and top-n limits. Generally it is observed that by the increasing of the

[2] http://www.ceng.metu.edu.tr/

recommendation limit number, the accuracy is decreasing. Comparing previous models UPR and PPR with ST, it is observed that accuracy improvement is above 40 % under *Ksim* and *Osim* similarity metrics.

6 Conclusion

In this work, we devised a next page prediction model that primarily uses semantic similarity of Web pages in order to sort them for recommendation ordering, and supports the model with PPR values in case of ties. We conducted a set of algorithms in order to evaluate the success of Semantic Tagging (ST) based Next Page Prediction. We observed that ST model for next page prediction is a promising approach with higher accuracy than that of previous similar models. In the semantic tagging process we pick the most frequent pages (frequency threshold is 10) and we tag them each related concepts manually. As a future work, automatic tagging can be developed in order to decrease the effort of manual tagging. Furthermore these experiments can be applied to another domain with this automated process. In addition to this, in semantic tagging process, association rules can be defined for next page predictions.

References

1. Deshpande, M., Karypis, G.: Selective markov models for predicting web page accesses. ACM Trans. Internet Technol. **4**, 163–184 (2004). 10.1145/990301.990304
2. Duhan, N., Sharma, A., Bhatia, K.: Page ranking algorithms: a survey. In: Advance Computing Conference, 2009. IACC 2009. IEEE International, pp. 1530–1537 (2009)
3. Eirinaki, M., Vazirgiannis, M.: Usage-based pagerank for web personalization. In: Fifth IEEE International Conference on Data Mining, p. 8, November 2005
4. Gunel, B.D., Senkul, P.: Investigating the effect of duration page size and frequency on next page recommendation with page rank algorithm. In: Proceedings of the Fifth ACM Web Search and Data Mining Conference. ACM (2012)
5. Guo, Y.Z., Ramamohanarao, K., Park, L.: Personalized pagerank for web page prediction based on access time-length and frequency. In: IEEE/WIC/ACM International Conference on Web Intelligence, pp. 687–690, November 2007
6. Haveliwala, T.: Topic-sensitive pagerank: a context-sensitive ranking algorithm for web search. IEEE Trans. Knowl. Data Eng. **15**(4), 784–796 (2003)
7. Liu, H., Keselj, V.: Combined mining of web server logs and web contents for classifying user navigation patterns and predicting users' future requests. Data Knowl. Eng. **61**, 304–330 (2007)
8. Liu, H.: Towards semantic data mining. In: 9th International Semantic Web Conference (ISWC2010), November 2010. http://data.semanticweb.org/conference/iswc/2010/paper/448
9. Mobasher, B., Dai, H., Luo, T., Sun, Y., Zhu, J.: Integrating web usage and content mining for more effective personalization. In: Bauknecht, K., Madria, S., Pernul, G. (eds.) Electronic Commerce and Web Technologies, Lecture Notes in Computer Science, vol. 1875, pp. 165–176. Springer, Berlin (2000)
10. Oberle, D., Bettina, B.B., Hotho, A., Gonzalez, J.: Conceptual user tracking. In: Menasalvas, E., Segovia, J., Szczepaniak, P. (eds.) Advances in Web Intelligence, Lecture Notes in Computer Science, vol. 2663, pp. 955–955. Springer, Berlin (2003)

11. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. Technical Report 1999–66, Stanford InfoLab (November 1999), previous number = SIDL-WP-1999-0120
12. Paulakis, S., Lampos, C., Eirinaki, M., Vazirgiannis, M.: Sewep: a web mining system supporting semantic personalization. In: Boulicaut, J.F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) Knowledge Discovery in Databases: PKDD 2004, Lecture Notes in Computer Science, vol. 3202, pp. 552–554. Springer, Berlin (2004)
13. Qi, X., Davison, B.D.: Web page classification: features and algorithms. ACM Comput. Surv. **41**, 12:1–12:31 (2009). 10.1145/1459352.1459357

New Techniques for Adapting Web Site Topology and Ontology to User Behavior

Oznur Kirmemis Alkan and Pinar Senkul

Abstract The World Wide Web is an endless source of information and the information is mainly represented in the form of Web pages that the users may browse. The way that the users browse the Web depends on the factors like the attractiveness of the Web sites, their structure and navigational organization. The preferences of users are changing through time, which brings difficulties for building Web sites that best suit users' profiles. Therefore, it is an important and challenging task to adapt the Web sites to the users' needs. Adaptation of Web sites becomes more effective when it involves the semantic content of the Web pages. In this paper, a framework is proposed that aims to adapt both the topology and the ontology of the Web sites by using semantic content and Web usage mining techniques.

Keywords Web usage mining · Web site adaptation · Ontology

1 Introduction

There is a huge growth of information sources on the Internet every day and together with this growth, the user base also grows. In such a situation, the necessity of managing this information for a huge number of users with possibly diverse needs arises. The way that the users browse the Web depends on the information that they are searching for. Therefore, the structure of the Web sites, which shapes how the information is organized on the site, in other words, the Web sites' navigational organization, affect users' Web surfing. In addition, Web sites' attractiveness, even

O. K. Alkan · P. Senkul (✉)
Computer Engineering Department, Middle East Technical University (METU),
06800 Ankara, Turkey
e-mail: senkul@ceng.metu.edu.tr

O. K. Alkan
e-mail: oznur.kirmemis@ceng.metu.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_43,

their color schema together with navigational easiness affects users' satisfaction and the popularity of the Web site. Therefore the Web sites design should in a way get adapted to its users' needs.

However, one concern that should be kept in mind when the adaptation of the Web sites is considered is that; users browse the Web not only for the Web sites' structure or links but also for their semantic structure. Therefore, in a way, some sort of semantic knowledge should also be integrated in the adaptation process. In order to achieve this, ontology is used in exploring the Web sites' structure and usage for adapting them towards better structural states [7]. In this work, new techniques for the adaptation of Web sites' topology as well as the underlying ontology are presented. There has not been many studies that are reported so far for solving the Web site topology and ontology adaptation tasks within a single framework. In [7, 8] authors described a system for self-adaptive Web sites, which also addresses the adaptation of the semantic Web, based on the Web usage data. The described solution employs Web usage mining as well as text mining methodologies and both the physical and semantic structure of the Web is targeted. Several different solutions have been proposed in [3, 5, 11] for providing users with assistance in their Web navigation, which are detailed in Sect. 2.

The important considerations that are examined and handled in this paper for the Web site topology and the ontology adaptation within a single framework are summarized below:

- Most of the existing studies utilize the text and other data involved in the Web pages so as to determine the concepts behind Web pages. However, in our work, instead of using the Web page content for determining the concepts associated with the page, this association is determined from the URLs of the accesses (not pages themselves).
- For the topology adaptation task, the proposed solution constructs two lists that present adaptations in terms of *full concept matching* and *partial concept matching*, which are detailed in Sect. 3.
- For the ontology adaptation process, our solution constructs adaptations to the ontology of the Web site by analyzing the concepts that frequently occur together in sessions.

The rest of the paper is organized as follows. In Sect. 2 related work is discussed. Next, the proposed system is introduced in Sect. 3 and the evaluation phase is detailed in Sect. 4. Section 5 gives the concluding remarks and future lines of research.

2 Related Work

Adaptive Web has been a popular subject, especially in the e-commerce domain [2, 9]. Several systems have been developed towards this direction. One example of these applications is the WebWatcher [5], which suggests links to the users based on their online behavior. Initially, users are asked to provide what kind of information

they are seeking for when they enter the site. In addition, before they leave the site, they are asked whether they have found what they were looking for. After that, users' navigation paths are used in order to create suggestions for future visitors that seek the same content. The resulting suggestions are presented by highlighting the already existing hyperlinks. Another work is presented in [3], which is called the Avanti project. In this work, the aim is to discover the user's final objective as well as his next step. A model is built for the user, based partly on the information the user provides about himself and also from his navigation paths. Direct links to the pages that are thought to be relevant to the users are presented to them. A drawback of both the WebWatcher and the Avanti is that, they require users to be in active participation with the system in the adaptation process. The Footprints system [11], on the other hand, only uses the navigation paths of the users; therefore, it does not require any explicit information. The most frequent navigation paths are presented to the visitor in the form of maps and also the percentage of people who have followed those maps are displayed next to each link. However, as in the case of the previous two systems, no adaptation of the site's structure is performed.

In [10], authors present a conceptual framework for the adaptive Web sites. The focus is mainly on the semi-automatic creation of index pages that are created through clustering page visits. The assumption in their solution is that, if a large number of users visit a set of pages frequently, these pages should be related. In order to find out those frequently accessed pages, they have developed two clustering algorithms, namely, *PageGather* and *IndexFinder*. PageGather relies on the statistical methods to discover the candidate link sets, whereas the IndexFinder is a conceptual clustering algorithm, which finds the link sets that are conceptually coherent.

As it is mentioned in [8], the majority of the existing approaches in Web adaptation process do not address the semantic aspects of the Web. The claim here is that, it should not be disregarded that users browse a site mainly for whatever exist on the Web pages; therefore, semantic information should also be utilized in the Web adaptation task. In order to capture this, a framework is proposed in [7, 8] for semantic Web adaptation in which the user's needs and requirements are the driving force. This framework uses text in the pages of the Web sites and Web usage mining to support Web site ontology and topology evolution. However, text based concept determination can lead to problems when the content data does not contain enough information to correctly identify concepts. In addition, accesses to sources that do not have a Web page behind cannot be processed.

3 Proposed Approach

3.1 Definitions

Before describing the framework, definitions of the important concepts for the description of the solution are presented in this section. A ***pageset*** is defined as

a set of pages that are frequently accessed together during the same session [8]. It contains the most frequently visited pages in a session which is represented as: *p-set* $= \{p_1, \ldots, p_n, n, sup, tclass, oclass\}$, where p_i is a unique page in the pageset, n is the size of the pageset, *sup* is the support of the pageset measuring its frequency in the dataset, *tclass* is the location of the pageset in the topology, and *oclass* is the location of the pageset in the ontology.

Four new concepts are introduced as a part of the solution, namely, *access_set, concept_list, full concept matching* and *partial concept matching*. ***Access_set***, different from pageset, contains a set of triples where each triples' first element is an access, the second element is the class value of the access that shows whether the access is a data source like a ".pdf" document or a Web page, and the third element is the set of concepts that are contained in that access. In addition, similar to pageset, the *tclass* is the location of the access in topology. *oclass* values of each access is kept to be able to perform partial and full concept matching and ontology adaptation tasks. As a result, an access_set is represented as follows; $aset = \{<a_1, class_1, oclass_1>, \ldots, <a_n, c_n, oclass_n>, n, sup, tclass\}$.

Concept_list keeps the list of the concepts that exist together in sessions. It is represented as: clist = $\{c_1,\ldots, c_n\}$. Concept_lists are used during the ontology adaptation task and they can include the same concepts for more than once, which is the reason it is defined as a list, not a set.

Full concept matching is the process of seeking to match all the concepts between two accesses when a shortcut link is to be proposed between them. However, in ***partial concept matching***, a shortcut link is proposed when at least one common concept exists between accesses.

3.2 General Framework

The framework described in this study consists of five main processes, which are described in the following paragraphs.

The ***pre-processing*** component works in a similar manner as the pre-processing module defined in almost all of the Web usage mining frameworks. It does log cleaning, user and session identification tasks. After pre-processor completes its job, ***concept determination*** component finds the concepts contained in all the accesses of the discovered sessions. The concept determination is performed in a rule-based manner, details of which are described in the following subsection.

Session mining component identifies access sets. In addition to the Web pages, accesses to information sources are also considered, and the class value of each access, which indicates whether the access is really an access to a Web page or not, is kept. The o_class parameter of these accesses is set from the concepts discovered by the concept determination component. ***Concept mining*** takes the sessions formed by pre-processor, and forms concept lists for each session which is lists of concepts that are contained in all the accesses of the related sessions. Concepts that frequently

occur together in several sessions are identified using Apriori algorithm [4], and the resulting concepts are output to be used by the Web adaptation component.

Finally, ***Web adaptation*** component performs two jobs: topology and ontology adaptation. *Topology adaptation* can be performed both in full and partial concept matching manners; where the former proposes links between accesses in the access_sets, whose oclass values all match for each of the individual accesses; and the latter proposes links between accesses when at least one of the concepts in the oclass values do match. In topology adaptation, shortcuts between two accesses are proposed only if the source access' class parameter indicates that it is an access to a Web page. In addition, topology adaptation directly uses the results of the concept mining component. It proposes addition of new relations between concepts that are found to be occurring together in a sufficient number of sessions.

Rule Based Categorization of Web Pages from URL. In the previous adaptation studies, authors generally use Web content data in order to classify Web pages according to the ontology concepts [8]. This way of determining the concepts requires positive and negative training samples for each of the concepts in the ontology, which becomes a difficult and time-consuming task as the ontology grows. In addition, its success heavily depends on the content of the data in a Web page, which may not correctly reflect the concepts, the Web page includes. If the *oclass* of a pageset is determined incorrectly, this will result in incorrect and inappropriate adaptations. In addition, when Web server log files are examined, the file accesses such as the ones with file extensions ".pdf" or ".mp3", are encountered. Such requests are actually important for understanding the user's interest for the Web site. For example, consider the request for *"/∼mike/ceng700/Schedule/assignment4.pdf"*. With text parsing, it is not possible to determine the concepts behind this access. However, the URL of the Web page itself contains important information about the navigation of the user. When this link is examined, it is obvious that, the user's intention behind accessing this document is to gather the assignment of the course *ceng700*. Therefore, adapting the content of the Web pages in terms of the accesses to the data sources will be highly beneficial.

Considering the above discussions, in our framework, the concepts behind accesses are determined using the URLs. The reason of extracting the concepts from URLs is due to both making the concept determination task easier and more accurate and the whole adaptation task more successful through considering accesses to sources other than only Web pages. URLs are used in the categorization of Web pages in several other studies [1, 6]. In our work, a similar intuition is used in order to discover the concepts behind pages. For instance, consider the access */courses/ceng536/*. Given categories such as *course, research, lecture notes, homework, graduate* and *undergraduate*; it is easy to guess that the page related with this access belongs to the *course* category. Furthermore, by using the domain information that the course codes that have a pattern such as *5xx* is a graduate course, *graduate* concept can be further assigned to this access. In order to realize these ideas, the proposed framework performs concept determination by using a rule-based approach on URLs. Concepts can be related with one or more rules, where each rule is specified as a regular expression.

Whenever the concepts involved in an access are to be determined, the URL of the access is tested against each regular expression so as to determine whether the URL matches the expression or not.

4 Evaluation

The proposed framework is evaluated by using the log data of METU Computer Engineering Department's Web site.[1] In the evaluation phase, the effect of the support parameter on the size of the resulting adaptation lists is analyzed. In addition, the proposed adaptations for the Web site topology and ontology under the optimal support value are presented.

Dataset. The log data initially contains 214010 accesses. After removing the noisy and irrelevant data, we left with 13979 accesses. From these accesses, 211 different sessions are identified.

Results. In order to examine how the number of proposed adaptations changes with the support parameter for the partial and full concept matching tasks, experiments are performed under the values in the range [0.01–0.05] with 0.005 increments. The resulting number of access_sets and the number of proposed adaptations for the Web site topology using partial and full concept matching is displayed in Fig. 1. This figure shows that, as the support threshold increases, the number of access_sets decreases, which is an expected result. The count of the proposed topology adaptations also decreases with the increasing support; however the decrease is much lower than that of the access_sets. In addition, as support value increases, the number of adaptations that full and partial concept matching techniques can discover, gets very close to each other.

The results presented above are meaningful since as we discover less number of access_sets, the number of adaptations that can be proposed for the topology improvement decreases. However, when we take a low support value, we obtain a larger number of adaptations, and since ontology is utilized in the adaptation process, the relevance of the accesses between which a shortcut is proposed does not decrease much.

Full concept matching proposes more relevant adaptations compared to partial concept matching. However, partial concept matching also proposes some surprising and satisfying adaptations that adds a random factor to the adaptation task.

Figure 2 displays the number of relations that are proposed to be added to the ontology of the Web site, under different support values. It shows that the count of the adaptations decreases with the increase in the support; however, it stabilizes after 0.035. From this figure, we can conclude that, the concepts that exist together with high values of support are highly correlated to each other.

[1] http://www.ceng.metu.edu.tr

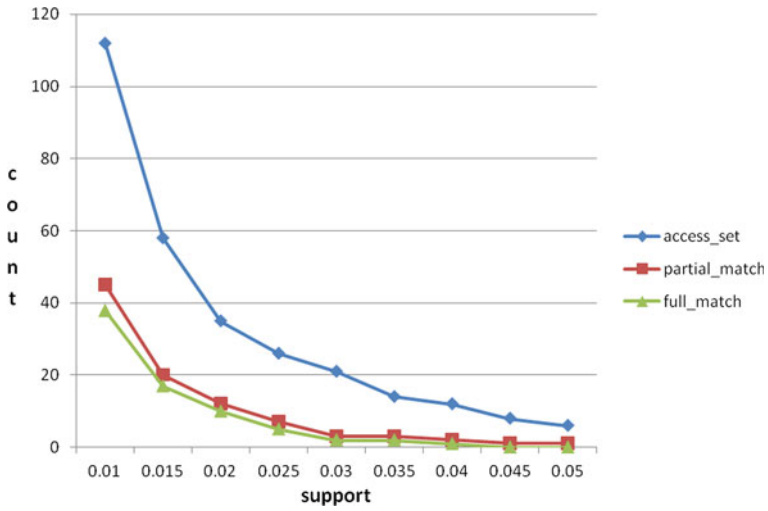


Fig. 1 Number of pagesets found, the size of the adaptation list for web site topology resulted from partial and full concept matching for different support threshold value

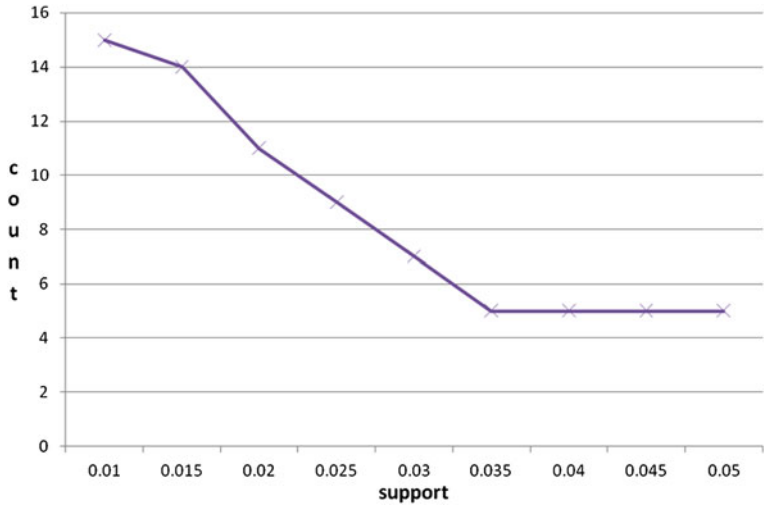


Fig. 2 Number of relations proposed to be added to the ontology for different support threshold values

In Fig. 3, sample adaptation lists proposed for the Web site ontology improvement are displayed. This list is generated under the support parameter set to 0.025.

When the results of the proposed adaptations for the ontology are examined, it is observed that finding relevance between ontological concepts after running Apriori

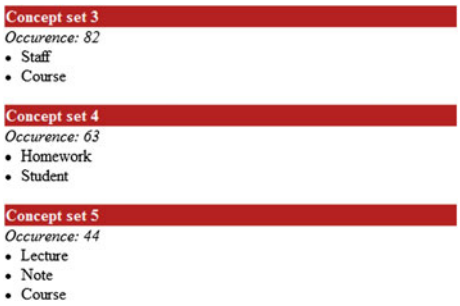


Fig. 3 Resulting ontology adaptation list

Table 1 Relations discovered between ontological concepts

	Rel #2	Rel #3	Rel #4
Apriori on access_ sets	7	1	–
Apriori on concept list	8	3	2

on the concept_lists' discovers more relations than running Apriori on access_set's and then discovering relations. Table 1 below summarizes these results.

5 Conclusion and Future Work

In this paper, a framework for performing Web site topology and ontology adaptation using the server access logs of the Web sites is presented. For the solution, different techniques are proposed in order to increase the satisfaction of the end user for the proposed topology and ontology adaptations. The main contributions of this paper are; performing concept determination from access URLs using a rule-based approach, and discovering topology and ontology improvements within a single framework. Frequency between concepts in sessions is taken into account during the Web site ontology adaptation. In addition, partial concept matching and full concept matching techniques are used in the Web site topology improvement. Partial concept matching can be seen as a way to add a randomization to the resulting adaptations so as to increase the serendipity in the proposed adaptations.

As the future work, the proposed framework can be enhanced to support additional adaptations such as deleting an already existing link. In addition, further adaptations for the Web site ontology can be studied in order to discover new concepts in addition to new relations. The system is also planned to be tested with another Web site that includes more complex topological and ontological patterns that may result in richer adaptation lists.

References

1. Achananuparp, P., Han, H., Nasraoui, O., Johnson, R.: Semantically enhanced user modeling. In: Proceedings of the 2007 ACM Symposium on Applied Computing (SAC '07), pp. 1335–1339. ACM Press, New York (2007)
2. Coenen, F., Swinnen, G., Vanhoof, K., Wets, G.: A framework for self adaptive websites: tactical versus strategic changes. In: Proceedings of WEBKDD'2000 Web Mining for E-Commerce—Challenges and Opportunities, Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, USA (2000)
3. Fink, J., Kobsa, A., Nill, A.: User-oriented adaptivity and adaptability in the AVANTI project. In: Proceedings of Conference Designing for the Web: Empirical Studies, Microsoft, Redmond (1996)
4. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. 3rd edn. Morgan Kaufmannm, San Francisco (2011)
5. Joachims, T., Freitag, D., Mitchell, T.: WebWatcher. A tour guide for the World Wide web. In: Proceedings of International Joint Conference on Artificial Intelligence, pp. 770–775. Nagoya, Japan (1997)
6. Kan, M.-Y., Thi, H.O.N.: Fast webpage classification using URL features. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management. New York, USA (2005)
7. Mikroyannidis, A., Theodoulidis, B.: A theoretical framework and an implementation architecture for self adaptive web sites. In: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web, Intelligence, pp. 558–561 (2004)
8. Mikroyannidis, C.A., Theodoulidis, B.: Web usage driven adaptation of the semantic web. In: Proceedings of End User Aspects of the Semantic Web Workshop, 2nd European Semantic Web Conference, pp. 137–147. Heraklion, Greece (2005)
9. Perkowitz, M., Etzioni, O.: Adaptive web sites: an AI challenge. In: Proceedings of IJCAI-97. Nagoya, Japan (1997)
10. Perkowitz, M., Etzioni, O.: Towards adaptive web sites: conceptual framework and case study. Artif. Intell. **118**(1–2), 245–275 (2000)
11. Wexelblat, A., Maes, P.: Footprints: History-rich tools for information foraging. In: Proceedings of Human Factors in Computing Systems (CHI), pp. 270–277. Pittsburgh, USA (1999)

Temporal Analysis of Crawling Activities of Commercial Web Robots

Maria Carla Calzarossa and Luisa Massari

Abstract Web robots periodically crawl Web sites to download their content, thus producing potential bandwidth overload and performance degradation. To cope with their presence, it is then important to understand and predict their behavior. The analysis of the properties of the traffic generated by some commercial robots has shown that their access patterns vary: some tend to revisit the pages rather often and employ many cooperating clients, whereas others crawl the site very thoroughly and extensively following regular temporal patterns. Crawling activities are usually intermixed with inactivity periods whose duration is easily predicted.

Keywords Web robots · Crawling · Temporal patterns

1 Introduction

Web robots are agents that traverse the Web and access and download Web pages without any significant human involvement [7]. These agents are a fundamental component of many applications and services, e.g., search engines, link checkers, Web services discovery. Nevertheless, some robots open up privacy and security issues as well as performance issues [11]. Although robots employed by major search engines tend to behave, the highly dynamic nature of Web content requires some sort of aggressive crawling policies. For example, to provide up-to-date content, commercial robots frequently revisit Web sites, thus draining server resources and causing potential bandwidth overload and overall performance degradation of the

M. C. Calzarossa · L. Massari (✉)
Dipartimento di Ingegneria Industriale e dell'Informazione,
Università di Pavia, Via Ferrata 1, 27100 Pavia, Italy
e-mail: massari@unipv.it

M. C. Calzarossa
mcc@unipv.it

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_44,

sites. Hence, to avoid damages and economic losses, it is important to identify Web robots and understand their behavior and their impact on the workload of a Web site.

This paper focuses on the analysis of some commercial robots with the objective of characterizing their behavior and their access patterns. The outcomes of this temporal analysis could be very useful for Web site administrators to estimate and predict the traffic due to robots and develop regulation policies aimed at improving site availability and performance.

Our study relies on the Web access logs collected on the European mirror of the Standard Performance Evaluation Corporation (SPEC) Web site. The choice of this site is motivated by its content, i.e., the performance results of standardized benchmarks of the newest generation high-performance computers. This content makes the site very relevant to the entire community of IT specialists and even more to search engines.

The paper is organized as follows. Section 2 briefly presents the state of the art in the area of the Web robot identification and characterization. The methodological approach applied in our study is presented in Sect. 3. The results of the exploratory analysis and of the temporal patterns followed by Web robots are discussed in Sects. 4 and 5, respectively. Finally, Sect. 6 summarizes the major findings of this study.

2 Related Work

The identification and characterization of the traffic generated by Web robots have been addressed in several papers (see e.g., [1, 2, 4–6, 9, 10]). Some studies analyze the overall properties of the traffic, whereas others take into account more specific aspects. A detailed survey of the existing robot detection techniques is presented in [3], where authors classify the techniques into four categories, discuss strengths and weaknesses of the underlying detection philosophy and suggest new strategies that try to overcome current limitations.

As commercial robots employed by search engines produce a large fraction of the overall traffic experienced by Web sites, some papers specifically focused on the characterization of this type of traffic. Dikaiakos et al. [2] compare the behavior of the crawlers of five popular search engines by analyzing access logs collected on various academic Web servers and introduce a set of metrics for their qualitative description. Similarly, Lee et al. [5] investigate the characteristics of some popular Web robots by analyzing a very large number of transactions recorded by a commercial server over a 24 h period. Metrics associated with HTTP traffic features and resource types are then used for the classification of the robots. The analysis, though very detailed, fails to investigate the temporal or periodic patterns of the traffic.

In this study we complement previously published results in that we focus on the temporal behavior of commercial robots. More specifically, our investigation extensively analyzes the temporal properties and patterns of the transactions belonging to Web robots with the aim of identifying models able to represent and predict their behavior.

3 Methodological Approach

Our study relies on the log collected on the European mirror of the SPEC Web site [8] for one year, from April 2009. The information recorded in the log according to the Extended Log File Format, refer to the main characteristics of each HTTP transaction processed by the site, such as, IP address of the client that issued the HTTP request, timestamp of the transaction, method and resource requested, user agent used by the client to issue the request.

The methodological approach adopted for the analysis of the Web log consists of various steps. In particular, after the identification of the transactions belonging to commercial Web robots, exploratory statistical techniques are applied to discover and highlight the main characteristics of these transactions and select the parameters that describe their temporal behavior. The investigation has then to focus on the distributions of these parameters to find out their properties, e.g., correlations, time dependence, and identify their temporal patterns. In particular, the analysis of the times between consecutive transactions of a given robot is the basis for the identification of its sessions, that is, the set of transactions issued within a given time interval. Sessions are typically intermixed with inactivity periods during which a robot does not issue any transaction. Temporal patterns are then studied as sequences of activity and inactivity periods. Through the application of numerical fitting techniques, models that predict these patterns, in terms, for example, of times elapsed between consecutive transactions and between consecutive sessions, are identified. These models can then be used to predict crawling activities and estimate the impact of the traffic of Web robots on the overall workload of a Web site.

4 Exploratory Analysis

A preliminary processing of the log has shown that the site was visited by more than 19,000 different clients and approximately 18 % of them visited the site only once during the entire year. In terms of traffic, the site served some five million HTTP transactions and generated about 130 GB of data. Most of the transactions used the GET method and were successful, that is, the status code of the server responses was 2XX, whereas the number of bad transactions with status code 4XX was almost negligible. Moreover, most of the referrer fields were empty, thus denoting navigation profiles characterized by a very limited degree of interactivity.

From a more detailed analysis of the IP addresses of the clients coupled with their corresponding user agents, we discovered that well-known commercial Web robots generated the majority of the traffic. In particular, as shown in Table 1, we ascribe about two million transactions to `msnbot`, that is, the robot employed by the Microsoft search engine—also known as `bingbot` after Microsoft officially introduced the Bing search engine—and about 1.5 million to `Googlebot`, the robot employed by the Google search engine. Apart from these robots, most of the

Table 1 Characteristics of the traffic generated by commercial Web robots

	# Transactions	# Clients	Data volume (GB)
Microsoft	1,975,232	1,653	39.59
Google	1,424,054	533	47.85
Dotnet	313,861	2	8.03
Yahoo	237,328	262	6.01
Scirus	165,037	1	5.98
Baidu	7,978	327	0.14
Exalead	4,390	1	0.10

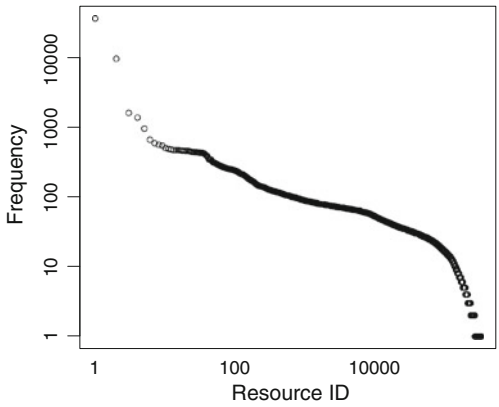


Fig. 1 Log–log scale plot of the popularity of the resources

others are definitely less active. For example, this is the case of `Baiduspider` and `Exabot`, the robots of the Chinese search engine Baidu and of the French search engine Exalead, respectively, that are responsible for less than 8,000 transactions each. The number of different clients involved in the crawling activities varies significantly from organization to organization. For example, Scirus, the engine dedicated to search scientific information on the open Internet, employs one client only, whereas Microsoft relies on a very large number of clients often operating in parallel and cooperating in a crawling session. In terms of data volume, Google contributes for more than 40 % of the data transmitted by the site, whereas Microsoft is responsible for less than 35 % although its number of transactions is about 39 % larger.

From the analysis of the resources requested by the Web robots considered in our study we discovered some interesting findings. More specifically, robots spread their requests over some 300,000 different resources whose popularity, as shown in Fig. 1, varies significantly and does not follow any Zipf distribution. The `robots.txt` file, i.e., the file used by Web site administrators to specify the rules of operation of robots, and the index file of the site were retrieved more than 36,000 and 9,500 times, respectively, whereas most of the other resources were retrieved much fewer times, namely, once or twice for one third of the resources and up to 20 times for about three quarter. This is in line with the

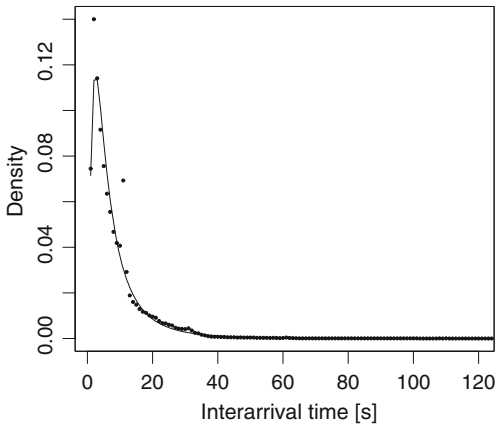


Fig. 2 Interarrival times of the transactions of all robots: measured (*dotted line*) and fitted model (*continuous line*)

behavior expected by robots whose main goal is to index and keep fresh the entire content of a site. We have also discovered that resource popularity is independent of their size.

The rate at which a robot crawls a site depends on many factors, including, among the others, the type of content and how often it is modified as well as the crawling policy adopted. Our investigation has outlined the thorough and extensive crawling activities performed by the Google robots: over one year they downloaded at least once almost all the resources of the site. On the contrary, the Microsoft robots reached a much smaller set of resources, each characterized by a higher revisit rate, on average about 12 visits per resource, compared to five of the Google robots.

5 Temporal Behavior

To study the temporal properties of the overall traffic generated by Web robots, we have first analyzed the interarrival times, that is, the time elapsed between two consecutive transactions of any robot. As shown in Fig. 2, the distribution is positively skewed with most of these transactions arriving very close to each other. Note that even though the plot does not span the entire range of variability of the interarrival times, whose maximum value is 1,818 s, it covers more than 99.99 % of the observations. Indeed, the median and the third quartile of the distribution correspond to 5 and 10 s, respectively and 99.9th percentile to 74 s. The application of numerical fitting techniques has shown that a lognormal distribution best fits the observed data as most of the mass is concentrated on small values. The location and shape parameters identified by fitting are equal to 1.74 and 0.92, respectively.

From the analysis of the behavior of the individual robots we have discovered that some robots visit the site rather regularly with transactions spread over the entire

Table 2 Statistics of the interarrival times, in seconds, of the individual robots

	Mean	St. Dev.	Max	Skewness	Percentiles			
					25th	50th	75th	95th
Microsoft	15.8	28.1	7,420	31.0	3	9	29	56
Google	21.9	58.3	23,645	96.2	4	11	34	70
Dotnet	98.9	5,913.4	1,824,148	227.8	10	10	11	23
Yahoo	131.4	362.9	73,878	97.8	19	71	170	439
Scirus	154.6	10,063.8	2,512,479	169.7	60	60	61	119
Baidu	3,908.6	4,034.9	76,241	2.6	444	3,115	6,935	10,012
Exalead	3,031.3	29,592.2	956,147	15.2	10	10	10	22

year, whereas others visit the site only sporadically, with transactions concentrated in some periods of the year. These different visit patterns are reflected in the interarrival times computed for the individual robots. As can be seen in Table 2, for 95 % of the transactions of all robots, but Baidu and Yahoo, the interarrival times are rather small and do not exceed two minutes. Nevertheless, their maximum values are much larger, especially in the case of robots that sporadically visit the site. For example, the robot of Scirus is characterized by several inactivity periods of more than a week. On the contrary, the robots of Microsoft are almost always active: their maximum period of inactivity is about 2 h.

To investigate the time dependence of the interarrival times of the individual robots, we computed the autocorrelation function with various lags. The transactions of some robots, e.g., Scirus, Yahoo, are not characterized by any time dependence, whereas this is not the case of other robots, e.g., Baidu, Google, Microsoft. Moreover, the slow decrease of the autocorrelation functions computed for these robots suggests the presence of long-term correlations or self-similar behavior that imply heavy-tailed distributions. This conclusion is confirmed by the Hurst parameter whose estimates are always larger than 0.6.

To increase the accuracy of the predictive models, we subdivided the transactions of each robot into sessions, that is, sequences of transactions whose interarrival times are below a certain threshold. In particular, our analysis has suggested a 300 s threshold. The effect of this subdivision is a general decrease of the variability of the interarrival times within a session. This is especially true for the robots that tend to visit the site either periodically or sporadically. For example, the crawling activities of Dotnet robots resulted in some 7,000 sessions of about 45 transactions whose interarrival time is almost constant. The time between two consecutive sessions, that is, the intersession time, is at most one hour for about 90 % of the sessions of all robots.

Table 3 summarizes the main characteristics of the sessions of the various robots. As can be seen, these characteristics vary from robot to robot, especially in terms of number of transactions per session and session duration. On the contrary, the average number of clients employed in a session is usually small, with the exception of Microsoft whose crawling activities rely on a pool of about 28 clients operating

Table 3 Average characteristics of robot sessions

	Duration	Intersession	# Transactions	# Clients	# Sessions
Microsoft	26,443.28	521.03	1,463.13	28.3	1,350
Google	10,845.72	637.75	308.97	1.3	4,609
Dotnet	773.19	3,951.84	45.10	1	6,958
Yahoo	804.85	529.02	9.09	1.1	26,088
Scirus	7,282.34	7,481.30	83.90	1	1,967
Baidu	140.62	4,831.72	1.24	1.24	6,413
Exalead	280.31	70,154.23	23.22	1	189

Times are in seconds

in parallel. It is also worth noting that some of the sessions of the most active robots span days. Finally, let us remark that about 0.35 % of the transactions issued by robots belong to sessions consisting of one transaction only, that is, their interarrival times are larger than the fixed threshold.

To obtain a more accurate prediction of the crawling activities of the robots, we have analyzed the intersession times by applying fitting techniques and we have discovered that these times can be modeled by a Pareto distribution, thus denoting a heavy tail.

6 Conclusions

Web robots are responsible of large fractions of the traffic received by the sites. To cope with their presence, it is then important to understand and predict their behavior. In this paper, we have analyzed the access log of the European SPEC Web site, to investigate the properties of the traffic generated by some commercial Web robots and outline the similarities and differences existing among them. More specifically, we have discovered that some robots employ in their crawling activities a large number of clients working in parallel, whereas others rely on one client only. Moreover, not all robots tend to revisit the resources: a good fraction of the resources was retrieved only once or twice during the entire year. The analysis of temporal behavior of the robots has shown that most robots are characterized by regular access patterns in terms of sessions, number of transactions and times between consecutive transactions within a session. Crawling activities are intermixed with inactivity periods, whose duration follows some specific patterns. We can conclude that, despite these differences, the behavior of individual robots is well defined and, once a robot is identified, it is possible to predict its visit patterns.

As a future work, we plan to assess whether the characteristics of the traffic of commercial robots hold across sites. Based on the identification of the robots, we will then develop regulation policies that predict their behavior.

References

1. Calzarossa, M., Massari, L.: Analysis of Web logs: challenges and findings. In: Hummel, K., Hlavacs, H., Gansterer, W. (eds.) Performance Evaluation of Computer and Communication Systems—Milestones and Future Challenges. Lecture Notes in Computer Science, vol. 6821, pp. 227–239. Springer, Heidelberg (2011)
2. Dikaiakos, M., Stassopoulou, A., Papageorgiou, L.: An investigation of Web crawler behavior: characterization and metrics. Comput. Commun. **28**(8), 880–897 (2005)
3. Doran, D., Gokhale, S.: Web robot detection techniques: overview and limitations. Data Min. Knowl. Disc. **22**, 183–210 (2011)
4. Kwon, S., Kim, Y., Cha, S.: Web robot detection based on pattern-matching technique. J. Inf. Sci. **38**(2), 118–126 (2012)
5. Lee, J., Cha, S., Lee, D., Lee, H.: Classification of web robots: an empirical study based on over one billion requests. Comput. Secur. **28**(8), 795–802 (2009)
6. Lourenco, A., Belo, O.: Catching Web crawlers in the act. In: Proceedings of the International Conference on Web Engineering, pp. 265–272 (2006)
7. Olston, C., Najork, M.: Web crawling. J. Found. Trends Inf. Retrieval **4**(3), 175–246 (2010)
8. SPEC Web Site—European mirror. http://spec.unipv.it
9. Stassopoulou, A., Dikaiakos, M.: Web robot detection: a probabilistic reasoning approach. Comput. Netw. **53**(3), 265–278 (2009)
10. Tan, P., Kumar, V.: Discovery of Web robot sessions based on their navigational patterns. Data Min. Knowl. Disc. **6**(1), 9–35 (2002)
11. Thelwall, M., Stuart, D.: Web crawling ethics revisited: cost, privacy, and denial of service. J. Am. Soc. Inf. Sci. Technol. **57**(13), 1771–1779 (2006)

A Framework for Sentiment Analysis in Turkish: Application to Polarity Detection of Movie Reviews in Turkish

A. Gural Vural, B. Barla Cambazoglu, Pinar Senkul and Z. Ozge Tokgoz

Abstract In this work, we present a framework for unsupervised sentiment analysis in Turkish text documents. As part of our framework, we customize the SentiStrength sentiment analysis library by translating its lexicon to Turkish. We apply our framework to the problem of classifying the polarity of movie reviews. For performance evaluation, we use a large corpus of Turkish movie reviews obtained from a popular Turkish social media site. Although our framework is unsupervised, it is demonstrated to achieve a fairly good classification accuracy, approaching the performance of supervised polarity classification techniques.

1 Introduction

The sentiment analysis of user-generated text content in the Web has attracted lots of research interest in the last decade [6]. The research has focused on various aspects of sentiment analysis including extraction [10] and classification [7] of sentiments. In this work, our focus is primarily on predicting the polarity associated with a short piece of human-generated text, i.e., predicting whether a given piece of text has positive or negative sentiments. This is an important task that finds application in the analysis of online product reviews and user mood detection with implications for monetization in commercial products.

In this work, our focus is on sentiment analysis in Turkish, which is an agglutinative language that makes the sentiment analysis a relatively more complicated problem. The contributions of our work are the following. We propose a framework for sentiment analysis in text written in Turkish, especially focusing on informal and

A. G. Vural (✉) · P. Senkul · Z. O. Tokgoz
Middle East Technical University, Ankara, Turkey
e-mail: gural@ceng.metu.edu.tr

B. B. Cambazoglu
Yahoo! Research, Barcelona, Spain

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_45,

noisy text found in the Web. We customize a lexicon-based sentiment analysis library, SentiStrength [8], to make it work with Turkish text. We evaluate the performance of our framework using a large dataset containing online movie reviews that are written in Turkish. The experimental results indicate a fairly good accuracy in predicting the polarity of movie reviews.

The rest of the paper is organized as follows. In Sect. 2, we provide a survey of the previous work on non-English sentiment analysis and polarity detection in movie reviews. Section 3 gives a brief overview of the SentiStrength library. The sentiment analysis framework we propose for Turkish is described in Sect. 4. Section 5 summarizes the details of our data and presents the performance results. The paper is concluded in Sect. 6.

2 Previous Work

Most sentiment analysis techniques are designed for English. In recent years, however, several works focused on non-English languages. Atteveldt et al. [1] used machine learning techniques to determine the polarity of political news stories in Dutch. They extracted lexical and syntactic features besides three different clusterings of similar words based on annotated material. Ghorbel and Jacot [4] devised a supervised learning strategy using linguistic features obtained through part-of-speech tagging and chunking as well as semantic orientation of words obtained from the SentiWordNet sentiment analysis tool [2] to classify the polarity of movie reviews in French. Since SentiWordNet is for English, the authors translated the French words to English before getting their semantic orientation. Zhang et al. [11] addressed the challenges that are unique to the Chinese language. They evaluated a rule-based polarity classification approach against different machine learning approaches. In literature, the research on sentiment analysis in Turkish is limited. To the best of our knowledge, a detailed analysis is presented only in Erogul's work [3], which rely on supervised machine learning for polarity classification. Our work differs from [3] as we use a lexicon-based approach, completely unsupervised and independent of the problem domain.

A number of works applied sentiment analysis to predict the polarity of movie reviews. Turney [9] used an unsupervised learning technique based on the estimated semantic orientation of extracted phrases. He classified the reviews as "recommended" or "not recommended" according to their average semantic orientation. A prediction accuracy of 65.8 % is reported for a collection of 120 movie reviews. Pang et al. [7] compared the performance of different machine learning techniques on movie reviews taken from the IMDB movie database. The SVM classifier is shown to yield the best performance. Their features included unigrams, bigrams, part of speech information, and the terms positions. Among these, the unigrams were found to yield better performance. Kennedy and Inkpen [5] combined machine learning with a simple technique based on counting the positive/negative words in the movie reviews, demonstrating further improvements. Erogul's aforementioned work [3]

Table 1 Sentiment scores generated by SentiStrength for sample english sentences

Sentence	Positive score	Negative score	Binary prediction
I am going to the school	+1	−1	+1
I like to play chess	+2	−1	+1
I do not like to play chess	+1	−1	+1
I feel sorry for missing the class	+1	−2	−1
I hate your brother	+1	−4	−1
I really love you, but dislike your sister	+4	−3	+1

also uses a movie review dataset for the performance evaluation. That work reports 85 % prediction accuracy in a binary (positive and negative classes) classification scenario.

3 SentiStrength

SentiStrength is a lexicon-based sentiment analysis library developed by Thelwall et al. [8]. Given a short piece of text written in English, the library generates a positive and a negative sentiment score for each word in the text. The positive scores range from +1 (neutral) to +5 (extremely positive) while the negative scores range from −1 (neutral) to −5 (extremely negative).[1] The final positive (negative) sentiment score for the input text is computed by taking the maximum (minimum) of the positive (negative) sentiment score of the words in the text. The library can also produce binary labels about the polarity of the text. Table 1 shows some sample English sentences and the sentiment scores produced by SentiStrength. Interested readers may try the tool on the SentiStrength site.[2]

To compute word sentiment scores, SentiStrength uses several word lists:

- *Sentimental word list* contains more than 2,500 words together with their associated sentiment scores. The sentimental words and their scores are compiled by human editors. The list also includes regular expressions, e.g., the pattern "amaz∗" covers words such as "amazed", "amazing", "amazingly".
- *Booster word list* contains words that strengthen or weaken the sentiment of the succeeding non-neutral words, e.g., "good" has a score of +2, whereas "extremely good" has a score of +3 due to the booster word "extremely".
- *Idiom list* contains some common phrases. The sentiment scores of individual words in the phrase are overridden, e.g., "how are you" has a sentiment score of +2, instead of a neutral score of +1.

[1] There is both a positive and a negative score because the input text may contain sentiments in both directions (e.g., "I love you, but I also hate you.") [8].

[2] SentiStrength, http://sentistrength.wlv.ac.uk/.

- *Negation word list* contains a few negation words. If a negation word is followed by a positive word, the positive sentiment score is multiplied by -0.5. If a negation word is followed by a negative word, the negative sentiment is turned into neutral. The reader may find related examples in Table 1.
- *Emoticon list* contains some common emoticons which are associated with sentiment scores, e.g., ":)" has a score of $+2$.

4 Sentiment Analysis Framework

The motivation behind creating a sentiment analysis framework specific to Turkish, rather than using an existing framework for English, is due to certain differences between Turkish and English. These differences can be summarized as follows. First, Turkish is an agglutinative language, i.e., new and arbitrarily long words can be created by adding many suffixes to a root word. The added suffixes may change the polarity of words. In practice, it is not feasible to detect and add all variants of Turkish words into the sentimental word list. Second, negation words usually occur after the negated word. This is different than English, where negation words typically precede the word they negate. Moreover, in Turkish, the negation word can be in the form of a suffix ("-ma") within the word. Finally, Turkish has several letters that are missing in English ("ç", "ğ", "ı", "ö", "ş", "ü"). In informal writing on the Web, people tend to substitute these Turkish letters with the closest ASCII English letters ("c", "g", "i", "o", "s", "u"). This creates complication in identifying the words.

We designed and implemented a sentiment analysis framework taking into account the above-mentioned differences. Our framework consists of a pipeline of several software modules, each providing some input to the succeeding module in the pipeline. The input to the framework is a piece of text written in Turkish and the output is a prediction about the polarity of the sentiments in the text, i.e., either a positive or a negative class prediction.[3] The proposed framework is illustrated in Fig. 1. In what follows, we describe the modules in this pipeline. Table 2 shows the execution of these modules for a sample input text.

- *Sentence extractor*: This is a simple module which splits the input text into sentences based on certain sentence separators (i.e., ".!?"). Each sentence is then passed to the next module as a separate input.
- *ASCII character converter*: Each word in the input sentence is looked up in a dictionary and checked for spelling errors. If a corresponding term is not found in the dictionary or there is a spelling error, the term is passed as input to an ASCII-tolerant parser to see if the word is written using ASCII character substitution. At this step, the parser may rewrite the term by substituting certain characters (e.g., "guzel" becomes "güzel").

[3] We do not consider the neutral class and break the ties in favor of the negative class.

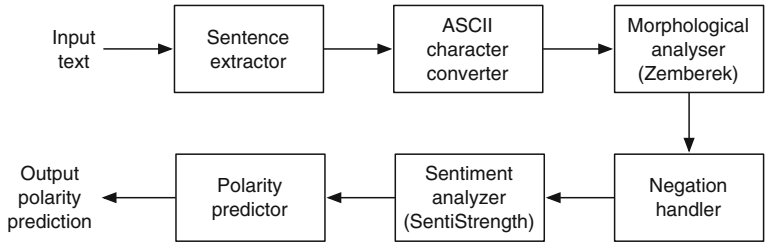


Fig. 1 The pipeline of modules in our sentiment analysis framework

Table 2 The execution of the modules in the pipeline for a sample input text

Module	Output of the module
Original input text	"bu film cok guzel degildi. :(hic kimseye tavsiye etmem."
Sentence splitter	"bu film cok guzel degildi." and":(hic kimseye tavsiye etmem."
ASCII converter	"bu film çok güzel değildi." and ":(hiç kimseye tavsiye etmem."
Morphological analyzer	"bu film çok güzel değil." and ":(hiç kimse tavsiye et-me-m."
Negation handler	"bu film çok güzel değil." and ":(hiç kimse tavsiye _NOT_ etmek."
Sentiment analyzer	"bu film çok güzel[3] değil [*-0.5 negated multiplier]." and ":([-1 emoticon] hiç kimse tavsiye[4] _NOT_ [*-0.5 negated multiplier] etmek."
Polarity predictor	(in all three methods, the polarity is predicted as negative)
`(sentence-binary)`	-1 and -1
`(sentence-max/min)`	$(+1, -2)$ and $(+1, -2)$
`(word-sum)`	-1.5 and -3

- *Morphological analyzer*: We then perform morphological analysis on the words in the sentence. To this end, we use the Zemberek library, which is an open source, platform-independent, and general-purpose natural language processing library for Turkic languages.[4] Zemberek's morphological analyzer basically finds all possible root forms and suffixes of a given word. We always assume that the first morphological analysis result of Zemberek is the correct one and use that. After the morphological analysis, certain suffixes are removed from the selected word form. This is because some suffixes (e.g., tense and person suffixes) are not valuable for sentiment analysis.

[4] Zemberek 2, http://code.google.com/p/zemberek/.

- *Negation handler*: The negation takes places in Turkish most often in two forms, either in the form of a separate word negating one of the preceding words (e.g., "güzel değil" ("not nice")) or in the form of the "-ma" suffix, which is a part of the negated word (e.g., "olmayacak" ("it will not happen")). To handle the negations of the first form, we rely on a SentiStrength feature, which we will briefly describe later. To handle the second form of negations, we modify the sentence and introduce an artificial keyword[5] before the negated word. This artificial word is added to the negation word list of our customized version of the SentiStrength library.
- *Sentiment analyzer*: We customized the lexicon files of the SentiStrength library by translating them to Turkish. The translation is performed by human editors, who also added to the lists new words that were missing in the original SentiStrength. Other than the changes in the lexicon files, we did not perform any modification in the scoring logic of SentiStrength as the codes of the library are not publicly available. To cope with the first form of negation words in Turkish, SentiStrength is initialized with a special parameter (`-negatingWordsOccurAfterSentiment`) to negate sentimental words before as well as after the negation words. SentiStrength by default applies negation to the words within a window of length 1. Our experiments indicated that a window size of 3 gives the best accuracy for Turkish, e.g., in the sentence "güzel film değil" ("it is not a nice movie"), "güzel" is affected from negation although it is not right before the negation word "değil" ("not").
- *Polarity predictor*: This module takes as input the sentiment scores associated with each word in the initial input text as well as the information about sentence splitting. The polarity of the input piece of text is determined according to the sentiment score assigned to the text. In our work, we evaluate three different approaches, which we refer to as `sentence-binary`, `sentence-max/min`, and `word-sum`:
  - `sentence-binary`: For each sentence in the original input text, we use the binary score (i.e., $+1$ or -1) generated by SentiStrength. The sum of the scores over all sentences gives the sentiment score of the text.
  - `sentence-max/min`: For each sentence, we use the maximum of positive word scores and the minimum of negative word scores provided by SentiStrength. The sum of positive scores and negative scores over all sentences gives the sentiment score of the text.
  - `word-sum`: We use the sum of the sentiment scores of all words in the input text as its sentiment score.[6]

  In all three scoring techniques, if the sentiment score of the text is positive, then its polarity is predicted as positive; otherwise, it is predicted as negative.

[5] We use "_NOT_" as the keyword.

[6] Use `-explain` option to obtain the sentiment scores of individual words in the text.

Table 3 Properties of the movie review dataset used in the experiments

Property	Positive reviews	Negative reviews
Number of reviews	30,000	30,000
Reviews including an emoticon	4,093	2,477
Average number of words	36.02	37.07
Average number of sentences	3.75	3.82
Average word length	6.03	5.92

5 Experiments

5.1 Dataset

To evaluate our sentiment analysis framework, we try to predict the polarity of online movie reviews written in Turkish. The review data is obtained from a movie site called Beyazperde,[7] a well-known website that provides information about movies. Beyazperde allows its users to enter comments on movies and state their opinion about the movie by selecting an icon (positive or negative), which forms the ground-truth polarity labels in our data. For the experiments, we picked a random sample of positive and negative reviews, each with equal number of documents. The properties of our dataset is shown in Table 3.

5.2 Results

We evaluate the performance in terms of accuracy, i.e., the ratio of the number of reviews whose polarity is correctly predicted to the total number of reviews. Table 4 reports the accuracy values for the three scoring techniques mentioned in Sect. 4, as well as the true/false positive/negative rates. In this experiment, we activate all modules in the processing pipeline. According to the table, the `word-sum` scoring technique is the best performing technique, while `sentence-binary` performs considerably worse than the other two scoring techniques. This result indicates that a fine-grain (at the word level) aggregation of the sentiment scores is more promising. In Table 4, we also observe that the prediction performance is better for positive reviews. This is in contrast to what is reported by Thelwall et al.[8] for an English dataset. Overall, our performance does not reach the performance of Erogul's supervised machine learning approach on the same data (85 % accuracy). However, given that our technique is unsupervised and independent of the problem domain, we believe that the accuracy achieved by our framework (76 % accuracy) is promising.

[7] Beyazperde, http://www.beyazperde.com

Table 4 Performance results (over all review instances)

	`sentence-binary` (%)	`sentence-max/min` (%)	`word-sum` (%)
Accuracy	70.39	74.83	75.90
True positive rate	36.89	39.83	40.70
False positive rate	13.11	10.17	9.30
True negative rate	33.50	35.00	35.30
False negative rate	16.50	15.00	14.70

Table 5 Accuracy when some modules are turned off

Inst.	Modules	`sentence-binary` (%)	`sentence-max/min` (%)	`word-sum` (%)
All	All modules	70.39	74.83	75.90
	No ASCII conversion	69.48	73.72	74.67
	No morphological analysis	67.34	71.53	72.30
+	All modules	73.78	79.67	81.40
	No ASCII conversion	72.36	77.29	78.98
	No morphological analysis	72.12	78.12	79.21
−	All modules	67.00	70.00	70.39
	No ASCII conversion	66.60	70.15	70.36
	No morphological analysis	62.56	64.95	65.40

Table 5 shows the accuracies when the ASCII conversion or morphological analysis modules are turned off. We note that turning off the morphological analysis also turns off the negation handling for the within-word negations. According to the table, most of the achieved accuracy is due to the customized SentiStrength library. Nevertheless, including the ASCII conversion and morphological analysis modules in the framework brings reasonable improvement. In particular, for negative reviews, the accuracy increases by about 5 % for all three scoring techniques when morphological analysis is turned on. Turning the ASCII conversion module on seems to help more in case of positive reviews.

6 Conclusions

We proposed a framework for unsupervised sentiment analysis in Turkish text documents. Our framework used various linguistic tools as well as the customized version of the SentiStrength tool. We evaluated the performance of our framework in

predicting the polarity of movie reviews. The experiments over a large corpus of Turkish movie reviews indicate reasonable prediction accuracy. We plan to make the customized SentiStrength library freely available to the research community, to enable the reproducibility of our findings and to support the research on sentiment analysis in Turkish.

Acknowledgments This work is supported by grant number TUBITAK-112E002, TUBITAK. We thank Umut Erogul for providing us the data.

References

1. Atteveldt, V., Kleinnijenhuis, J., Ruigrok, N., Schlobach, S.: Good news or bad news? conducting sentiment analysis on dutch text to distinguish between positive and negative relations. J. Inf. Technol. Polit. **5**(1), 73–94 (2008)
2. Baccianella, A.E.S., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings 7th Conference International, Language Resources and Evaluation (2010)
3. Erogul, U.: Sentiment analysis in turkish. Master's thesis, Middle East Technical University (2009)
4. Ghorbel, H., Jacot, D.: Sentiment analysis of french movie reviews. In: Pallotta, V., Soro, A., Vargiu, E., (eds.) Advances in Distributed Agent-Based Retrieval Tools. Studies in Computational Intelligence, vol. 361, pp 97–108. Springer, Berlin (2011)
5. Kennedy, A., Inkpen, D.: Sentiment classification of movie reviews using contextual valence shifters. Comput. Intell. **22**(2), 110–125 (2006)
6. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retr. **2**, 1–135 (2008)
7. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of ACL-02 Conference Empirical Methods in Natural Language Processing, pp. 79–86 (2002)
8. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment strength detection in short informal text. J. Am. Soc. Inf. Sci. Technol. **61**(12), 2544–2558 (2010)
9. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In Proceedings of 40th Annual Meeting on Association for, Computational Linguistics, pp. 417–424 (2002)
10. Yi, J., Nasukawa, T., Bunescu, R., Niblack, W.: Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques. In Proceedings of 3rd IEEE International Conference Data Mining, pp. 427–434 (2003)
11. Zhang, C., Zeng, D., Li, J., Wang, F.-Y., Zuo, W.: Sentiment analysis of Chinese documents: From sentence to document level. J. Am. Soc. Inf. Sci. Technol. **60**(12), 2474–2487 (2009)

Part XI
Methods and Algorithms

Structure in Optimization: Factorable Programming and Functions

Laurent Hascoët, Shahadat Hossain and Trond Steihaug

Abstract It is frequently observed that effective exploitation of problem structure plays a significant role in computational procedures for solving large-scale nonlinear optimization problems. A necessary step in this regard is to express the computation in a manner that exposes the exploitable structure. The formulation of large-scale problems in many scientific applications naturally give rise to "structured" representation. Examples of computationally useful structures arising in large-scale optimization problems include unary functions, partially separable functions, and factorable functions. These structures were developed from 1967 through 1990. In this paper we closely examine commonly occurring structures in optimization with regard to efficient and automatic calculation of first- and higher-order derivatives. Further, we explore the relationship between source code transformation as in algorithmic differentiation (AD) and factorable programming. As an illustration, we consider some classical examples.

Keywords Algorithmic differentiation · Source code transformation · Factorable programming

T. Steihaug (✉)
Department of Informatics, University of Bergen, Box 7803,
5020 Bergen, Norway
e-mail: trond.steihaug@ii.uib.no

S. Hossain
Department of Mathematics and Computer Science, University of Lethbridge,
Lethbridge, AB, Canada

L. Hascoët
INRIA, Sophia-Antipolis, France

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_46,

1 Introduction

For simplicity, we will consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is sufficiently smooth. Methods of interest are those that require derivatives up to order three.
Let $e^{(i)}$ be the ith row of the identity matrix. A function f is separable if it can be written as

$$f(x) = \sum_{i=1}^{n} \phi_i(e^{(i)}x),$$

and can be decomposed into user-defined scalar functions ϕ_i. Given m matrices $U_i \in \mathbb{R}^{n_i \times n}$, $n_i \leq n$ where row k, $1 \leq k \leq n_i$ is a row of the identity matrix, a partially separable function [1] is given by

$$f(x) = \sum_{i=1}^{m} \phi_i(U_i x),$$

where each function $\phi_i : \mathbb{R}^{n_i} \mapsto \mathbb{R}$ is provided by the user. The functions $\phi_i, i = 1, \ldots, m$ are called element functions [1] and the variables $v^{(i)} \in \mathbb{R}^{n_i}$, $v^{(i)} = U_i x$ are called elemental variables [2]. Linear combinations of elemental variables are called internal variables [2–4], $u^{(i)} = W_i U_i x$. If W_i has more columns than rows, the element function ϕ_i will be functions of fewer than n_i variables. Bouaricha and Moré [5] describe software ELSO that computes the gradient of functions provided in partially separable form. To take advantage of partially separable structure one defines $\phi(x) = (\phi_1(x)\ \phi_2(x)\ \ldots\ \phi_m(x))^T$, then $f(x) = \phi(x)^T e$ where e is the vector of all ones. By employing algorithmic differentiation forward mode, the sparse Jacobian $\phi'(x)$ is computed yielding the gradient $\nabla f(x) = \phi'(x)^T e$. Gay [6] describes a method for detecting partially separable form of AMPL expressions which is then utilized in Hessian computations. Partially separable function minimization with AD on distributed memory parallel computing system has been considered in [7].
Let $u^{(i)} \in \mathbb{R}^n$ be m given vectors. A unary function [8] is given by

$$f(x) = \sum_{i=1}^{m} \phi_i({u^{(i)}}^T x), \quad \phi_i : \mathbb{R} \mapsto \mathbb{R}. \tag{2}$$

Let $g : \mathbb{R}^m \mapsto \mathbb{R}$ be a separable functions given by $g(v) = \sum_{i=1}^{m} \phi_i(v_i)$. Each function ϕ_i is provided by the user. Denoting the ith row of U by ${u^{(i)}}^T$, the unary function (2) becomes, $f(x) = g(Ux)$. The gradient and the Hessian matrix of f at

TF		EF	
$c_1u + c_2$	$\frac{c_1}{u} + c_2$	$u + v$	$u * v$
$c_2 e^{c_1 u}$	$c_2 \log(c_1 u)$	$-u$	c
$c_2 \sin(c_1 u)$	$c_2 \cos(c_1 u)$	$\frac{1}{u}$	
$c_2 \min\{u, c_1\}$	$c_2 \max\{u, c_1\}$	e^u	$\log(u)$
		$\sin(u)$	$\cos(u)$
c_1, c_2 are constants and u is a variable		u, v are variables and c is a constant	

Fig. 1 TF transformation function [11]. EF elemental function [20]

x are obtained in special forms,

$$\nabla f(x) = U^T \nabla g(Ux), \quad \nabla^2 f(x) = U \nabla^2 g(Ux) U^T. \tag{3}$$

Since g is separable, the Hessian matrix $\nabla^2 g$ is a diagonal matrix. Of particular interest is the case when $m = n$ and U has full rank [9]. In this case the unary function is a change of variables in g.

The notion of factorable functions predates that of partially separable functions and unary functions in optimization. A function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a factorable function [10] if it can be represented as the last function in a finite sequence of functions $\{\phi_i\}_{i=1}^{L}$ where $\phi_i : \mathbb{R}^n \mapsto \mathbb{R}$:

$$\begin{aligned}
&\phi_i(x) \equiv {u^{(i)}}^T x \text{ where } u^{(i)} \text{ are constant vectors}, \quad i = 1, \ldots, \ell \\
&\phi_i(x) \equiv \phi_{j<i}(x) \circ \phi_{k<i}(x), \quad i = \ell + 1, \ldots, L, \quad \circ \in \{\times, +, -, /, \hat{}\} \\
&\qquad \text{or} \\
&\phi_i(x) \equiv \tau_i(\phi_{j<i}(x)), \quad \tau_i : \mathbb{R} \mapsto \mathbb{R} \\
&f(x) = \phi_L(x)
\end{aligned}$$

The sequence $\{\phi_1(x), \ldots, \phi_\ell(x), \ldots, \phi_L(x)\}$ is called a factored sequence. The notation $\phi_{j<i}(x)$ means that there exists $j < i$ so that $\phi_j(x)$ is an element of the factored sequence defined above. The function τ_i is called a transformation function such as exponential, trigonometric and logarithm, but may also be user defined functions. In [11–14] the initialization is given by $\phi_i(x) \equiv x_i$, $i = 1, \ldots, \ell$, and $\ell = n$. Figure 1 shows examples of transformation functions. It is pointed out by Kedem [12] that the notion of factorable functions corresponds to a simple Fortran subroutine that consists of expression evaluations without "IF" and "GOTO" statements and with very limited loops. In the book on automatic differentiation Rall in 1981 [15] points out that what is called codeable functions in [15] are in fact factorable functions. In [16] a "nonlinear" factorable form that includes bilinear terms is used to solve mixed-integer nonlinear programming optimization problems. Methods for these classes of functions using a partial update Newton are considered in [17].

Almost any function used for computational purposes can be put into a factorable form. Examples of functions which cannot be are given in [18, Chap. 3] and [19]. The remainder of the paper is organized as follows.

Section 2 discusses factorable programming problems and functions. A factorable programming problem, discussed in Sect. 2 is a nonlinear optimization problem where the objective and the constraint functions are factorable functions. The definition of a factorable function from 1967 introduces the concept of structure in an optimization problem. The definition that is used by most authors is a recursive definition, with the initialization given by $\phi_i(x) \equiv x_i$, which is reviewed in the preceding paragraphs. The final part of Sect. 2 is a discussion of the relationship between a factorable function and algorithmic differentiation. Section 3 provides illustration of source transformation of selected factorable and generic unary functions from the literature.

2 Factorable Programming Problems

Factorable programming problems were introduced by McCormick [19, 21] in 1974. A factorable programming problem is of the form

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} \quad X^L(x) \\ &\text{Subject to} \quad l_i \le X^i(x) \le u_i, \quad \text{for} \quad i = 1, \ldots, L-1, \end{aligned}$$

where $X^i : \mathbb{R}^n \mapsto \mathbb{R}$. Here $X^i(x) = x_i$ for $i = 1, \ldots, n$ and for given $X^p(x)$, $p = 1, \ldots, i-1$ function X^i is defined recursively as

$$X^i(x) = \sum_{p=1}^{i-1} T_p^i(X^p(x)) + \sum_{p=1}^{i-1}\sum_{q=1}^{p} V_{q,p}^i(X^p(x)) \cdot U_{p,q}(X^q(x)), \tag{4}$$

where T's, U's, and V's are transformation functions of a single variable. The lower and upper bounds $l_i \le u_i$ are given constants (may include $\pm\infty$.) It follows immediately that functions $X^i(x), i = 1, \ldots, L$ in (4) can be written as factorable functions.

A factorable programming language combined with the program SUMT [22] for the general nonlinear optimization problem was derived by McCormick in 1974 in [21] and extended by McCormick and Ghaemi [11]. The functions $X^i(x), i = 1, \ldots, L$ in [11, 21] are called concomitant variable functions (cvfs). The cvfs consist of two terms: the first term is separable and the second is a quadratic term. The inputs to the program [11, 21] are split between these two terms. The input is line based and, for the separable part of cvf number i, each line of the input is element p in the sum together with the type of transformation T_p^i and the index p of the cvf X^p. Similarly, for the quadratic term, two transformations and the two cvfs need to be specified for each element in the (double) sum. A modeling language for nonlinear programming problems for factorable functions of the form (4) and the use of SUMT was developed by Pugh [23] in 1972. In this modeling language one can also specify sums and products, $\sum_{i=m_1}^{m_2} \cdot$ and $\prod_{i=m_1}^{m_2} \cdot$, in addition to transformations of a single

variable. The double sum $\sum_{i=1}^{5}\sum_{j=1}^{5} x_{ij}$, for example, can be represented as the string `SI(I,1,5,SI(J,1,5,X(I;J))).`
In a technical report from 1967 McCormick introduces the term "factorable nonlinear convex programming" for a class of problems whose nonlinear function have second partial derivatives given in a special form. The technical report is published in Fiacco and McCormick [24, pp. 184–188]. The point taken in [24, 25] is that a factorable function is one where the analytic derivation of the Hessian matrix directly yields this form. The processing of the modeling language [11, 21, 23] assembles the Hessian matrix on this special form.

2.1 Extending Factorable Functions

A somewhat more general definition of factorable functions will allow the transformations τ_i to be functions of several variables [12], i.e. $\tau_i(\phi_{i_1}, \ldots, \phi_{i_s})$, $i_1, \ldots, i_s < i$. In [12] Fortran programs are augmented with non standard data types and operators and the non standard constructs are translated by a pre-compiler into standard Fortran. The gradient of a factorable function given on the form [10] can be shown by a minor modification of the proof in [13], to be of the form

$$\nabla f(x) = \sum_{i=1}^{\ell} u^{(i)} \alpha_i(x),$$

where $\alpha_i(x) \in \mathbb{R}$ is composed of product of factored-sequence functions and the first derivative of the transformations. The Hessian matrix is of the form

$$\sum_{i=1}^{\ell}\sum_{j=1}^{\ell} u^{(i)} \alpha_{ij}(x) u^{(j)^T},$$

where $\alpha_{ij}(x) \in \mathbb{R}$ are composed of factored-sequence functions and, the first and second derivative of the transformations in the sequence. Jackson and McCormick [13] show that the higher derivatives too, will have a polyadic structure (the gradient will be a sum of monads and the Hessian matrix be a sum of dyads.)
We would like to emphasize that the polyadic structure of factorable function is preserved also in the case when $\phi_i(x)$ themselves are factorable functions for $i = 1, \ldots, \ell$. It follows immediately that for unary functions, the gradient and Hessian of f are given by (3) for $m = \ell$. The approach taken in this paper is that $\phi_i, i = 1, \ldots, \ell$, in general, are *user-defined functions given as computer programs*. An extension of partial separability introduced in [2] is to write the function as $f(x) = \sum_{i=1}^{\ell} \tau_i(\phi_i(x))$, where $\tau_i : \mathbb{R} \mapsto \mathbb{R}, i = 1, \ldots, \ell$ are called group functions and ϕ_i are partially separable functions. This is again an example of factorable functions.

2.2 The General Evaluation Procedure in AD

For a given value x the general evaluation procedure in automatic differentiation is given by

$$\begin{aligned} v_{n-i} &= x_i, && i = 1, \ldots, n \\ v_i &= \widehat{\phi}_i(v_j)_{j \prec i}, && i = 1, \ldots, \ell, \text{ where } \widehat{\phi}_i \text{ is an elemental function} \\ y &= v_\ell. \end{aligned}$$

Examples of elemental functions in AD are displayed in Fig. 1.
Each value v_i can be interpreted as an intermediate function $v_i(x)$ of the independent variable $x \in \mathbb{R}^n$ [20]. This interpretation exposes the relationship between AD and factorable functions. The transformations in [11, 21] are just combinations of elemental functions used in AD. Importantly, utilizing the fact that the derivatives have a polyadic structure, is not an alternative in AD since the number of elemental functions will be very high.
Distinct from the view in AD, the factored–sequence functions ϕ_i are user-specified functions and a source code transformation tool will naturally yield the derivatives of function f in a structure-preserving (e.g., polyadic) form. To illustrate this point, we consider an example by Jackson and McCormick [13] and unary functions using AD source transformation tool Tapenade [26].

3 Examples of Source Code Transformations

The following example is from Jackson and McCormick [13]. The function is given by

$$f(x) = a^T x \sin(b^T x) e^{c^T x}. \tag{5}$$

To make an efficient hand-coded evaluation of the gradient and the Hessian we rewrite the function. Let $a, b, c \in \mathbb{R}^n$ and let A be a $n \times 3$ matrix and $\phi : \mathbb{R}^3 \mapsto \mathbb{R}$:

$$A = [a\ b\ c], \quad \phi = (\phi_1, \phi_2, \phi_3), \quad g(\phi) = \phi_1 \sin(\phi_2) e^{\phi_3}, \quad \text{then } f(x) = g(A^T x).$$

The gradient and the Hessian matrix of f at x are:

$$\nabla f(x) = A \nabla_\phi g(A^T x), \qquad \nabla^2 f(x) = A \nabla^2_\phi g(A^T x) A^T,$$

where $\nabla_\phi g(\phi) = (\sin(\phi_2) e^{\phi_3}, \phi_1 \cos(\phi_2) e^{\phi_3}, \phi_1 \sin(\phi_2) e^{\phi_3})^T$ and

$$\nabla^2_\phi g(\phi) = \begin{pmatrix} 0 & \cos(\phi_2) e^{\phi_3} & \sin(\phi_2) e^{\phi_3} \\ \cos(\phi_2) e^{\phi_3} & -\phi_1 \sin(\phi_2) e^{\phi_3} & \phi_1 \cos(\phi_2) e^{\phi_3} \\ \sin(\phi_2) e^{\phi_3} & \phi_1 \cos(\phi_2) e^{\phi_3} & \phi_1 \sin(\phi_2) e^{\phi_3} \end{pmatrix}.$$

```
DO nd=1,nbdirs
  .......
  phi6bd(nd) = yb*phi5d(nd); phi5bd(nd) = yb*phi6d(nd)
  phi1bd(nd) = phi4d(nd)*phi6b + phi4*phi6bd(nd)
  phi4bd(nd) = phi1d(nd)*phi6b + phi1*phi6bd(nd)
  phi3bd(nd) = phi3d(nd)*EXP(phi3)*phi5b + EXP(phi3)*phi5bd(nd)
  phi2bd(nd) = COS(phi2)*phi4bd(nd) - phi2d(nd)*SIN(phi2)*phi4b
  xbd(nd, :) = b(:)*phi2bd(nd) + a(:)*phi1bd(nd) + c(:)*phi3bd(nd)
END DO
```

Fig. 2 Source transformation: forward mode $\nabla^2 f(x)$ based on the gradient code

For a given x the numbers of arithmetic operations are approximately $6n$ to compute the function and $11n$ to compute the gradient and the function.
As a factorable function, (5) can be decomposed into

$$\begin{aligned}
&\phi_1(x) = a^T x,\ \phi_2(x) = b^T x,\ \phi_3(x) = c^T x,\\
&\phi_4(x) = \sin(\phi_2(x)), \phi_5(x) = \exp(\phi_3(x)), \phi_6(x) = \phi_1(x) * \phi_4(x), \text{and}\\
&f(x) = \phi_6(x) * \phi_5(x).
\end{aligned}$$

The gradient code produced in source transformation reverse AD of a Fortran 90 implementation of the function shows that the numbers of arithmetic operations are approximately the same for the source transformation and the hand-coded gradient [10].

3.1 The Hessian Matrix with the Source Transformation Tool Tapenade

The Hessian matrix is computed using n matrix vector product $He^{(i)}, i = 1, \ldots, n$ where $e^{(i)}$ is the ith column of the identity matrix. In Fig. 2 we only show the innermost loop. The hand-coded second derivative requires approximately $\frac{5}{2}n^2$ arithmetic operations utilizing the symmetry. The number of arithmetic operations for the code in Fig. 2 is approximately $5n^2$. As a further illustration of the use of AD for structured problems we consider computing the gradient of a unary function (2). Hand-coded derivatives will be computed using (3). Assuming, for simplicity, that U is a square matrix, the number of arithmetic operations to compute the gradient will be roughly $4n^2$ plus the n function calls g_i'. From Fig. 3 it follows that the number of arithmetic operations is about the same as in hand-coded calculation for the function and the gradient in source transformation. The user must either use a source transformation tool of the user-specified functions or a hand-coded version of the scalar functions ϕ_i. In Fig. 3 we illustrate the use of source transformed user-specified functions.

Looking for further similarities between the AD adjoint in Fig. 3 and the mathematical gradient of a unary function (3), one may wonder what became of the transposition U^T. A closer look at the generated code `unary_b` reveals that it does

```
subroutine UNARY(x, n, U, y)
  integer n,i ; real x(n), U(n,n) v(n),y
  do i = 1,n
     v(i) = SUM(U(i,:)*x(:))
  enddo
  y = 0.0
  do i = 1,n
     y = y + F(i,v(i))
  enddo
end subroutine UNARY
```

```
subroutine UNARY_B(x, xb, n, u, y, yb)
  do i=1,n
     v(i) = SUM(u(i, :)*x(:))
  end do
  yb = 0.0
  do i=n,1,-1
     result1b = yb
     call F_B(i, phi(i), vb(i), result1b)
  end do
  xb = 0.0
  do i=n,1,-1
     xb = xb + u(i, :)*vb(i)
     vb(i) = 0.0
  end do
  yb = 0.0
end subroutine unary_b
```

Fig. 3 Example of a unary function and the source transformation using reverse mode

compute the correct gradient, but with no explicit transposition. Actually, from a computer science point of view, an AD tool has no idea that there is a matrix product, and even less that it should be transposed. What makes things work is data-flow reversal [27], which we can sketch as follows:

$$\texttt{x(:)} \xrightarrow{\texttt{U(i,:)}} \texttt{phi(i)} \qquad \Rightarrow \qquad \texttt{phib(i)} \xrightarrow{\texttt{U(i,:)}} \texttt{xb(:)}$$

In the original code, data flow leads from `x(:)` to `phi(i)`, using the constant `U(i,:)` on the way. The adjoint code, by reversing the data flow, leads from `phib(i)` to `xb(:)`, still using `U(i,:)` on the way. No transposition nor index manipulation is involved but the effect is the same. More generally, we observe that data flow reversal plays, in the AD adjoint code, the role played by transposition in the expression of the mathematical gradient.

4 Concluding Remarks

The relationship between algorithmic differentiation on elemental function level and factorable function is well known. However, the use of more computationally intensive elementals or user-defined elementals are usually not a part of the evaluation procedure in AD. The point taken here is that source transformation can be an attractive tool for a user when the functions are composed of a factored-sequence of known user-specified functions.

References

1. Griewank, A., Toint, Ph.L.: On the unconstrained optimization of partially separable functions. In: Powell, M.J.D. (ed.) Nonlinear Optimization 1981, pp. 301–312. Academic Press, New York (1982)
2. Conn, A.R., Gould, N.I.M., Toint, Ph.L.: An introduction to the structure of large scale nonlinear optimization problems and the LANCELOT project. In: Glowinski, R., Lichnewsky, A. (eds.) Computing Methods in Applied Sciences and Engineering, pp. 42–51. SIAM, Philadelphia (1990)
3. Conn, A.R., Gould, N.I.M., Toint, Ph.L.: LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A), 1st edn. Springer, Berlin (1992)
4. Conn, A.R., Gould, N.I.M., Toint, Ph.L.: Improving the decomposition of partially separable functions in the context of large-scale optimization: a first approach. In: Hager, W.W., Hearn, D.W., Pardalos, P.M. (eds.) Large Scale Optimization: State of the Art, pp. 82–94. Kluwer Academic Publishers, Amsterdam (1994)
5. Bouaricha, A., Morè, J.J.: Impact of partial separability on large-scale optimization. Comput. Optim. Appl. **7**, 27–40 (1997)
6. Gay, D.M.: More AD of nonlinear AMPL models: computing Hessian information and exploiting partial separability. In: Berz, M., Bischof, C., Corliss, G., Griewank, A. (eds.) Computational Differentiation: Techniques, Applications, and Tools, pp. 173–184. SIAM, Philadelphia (1996)
7. Conforti, D., De Luca, L., Grandinetti, L., Musmanno, R.: A parallel implementation of automatic differentiation for partially separable functions using PVM. Parallel Comput. **22**, 643–656 (1996)
8. McCormick, G.P., Sofer, A.: Optimization with unary functions. Math. Program. **52**(1), 167–178 (1991)
9. Steihaug, T., Suleiman, S.: Global convergence and the Powell singular function. J. Glob. Optim. 1–9 (2012). doi: 10.1007/s10898-012-9898-z. http://www.dx.doi.org/10.1007/s10898-012-9898-z
10. Hascoët, L., Hossain, S., Steihaug, T.: Structured computation in optimization and algorithmic differentiation. ACM Commun. Comput. Algebra **46**(3) (2012)
11. Ghaemi, A., McCormick, G.P.: Symbolic factorable SUMT: What is it? How is it used? Technical Report T-402. Institute for Management Science and Engineering, The George Washington University, Washington DC (May 1979)
12. Kedem, G.: Automatic differentiation of computer programs. ACM Trans. Math. Softw. **6**(2), 150–165 (1980)
13. Jackson, R.H.F., McCormick, G.P.: The polyadic structure of factorable function tensors with application to high-order minimization techniques. J. Optim. Theory Appl. **51**(1), 63–94 (1986)
14. Jackson, R.H.F., McCormick, G.P.: Second-order sensitivity analysis in factorable programming: theory and applications. Math. Program. **41**(1–3), 1–27 (1988)
15. Rall, L.B.: Automatic Differentiation: Techniques and Applications. Lecture Notes in Computer Science, vol. 120. Springer, Berlin (1981)
16. Smith, E.M., Pantelides, C.C.: Global optimisation of nonconvex minlps. Comput. Chem. Eng. **21**(Suppl.), S791–S796 (1997)
17. Goldfarb, D., Wang, S.Y.: Partial-update Newton methods for unary, factorable, and partially separable optimization. SIAM J. Optim. **3**(2), 382–397 (1993)
18. McCormick, G.P.: Nonlinear Programming: Theory, Algorithms and Applications. Wiley, New York (1983)
19. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: part I convex underestimating problems. Math. Program. **10**(1), 147–175 (1976)
20. Griewank, A., Walther, A.: Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. 2nd edn, Number 105 in Other Titles in Applied Mathematics. SIAM, Philadelphia (2008)

21. McCormick, G.P.: A mini-manual for use of the SUMT computer program and the factorable programming language. Technical Report SOL 74–15. Department of Operations Research, Stanford University, Stanford (August 1974)
22. Mylander, W.C., Holmes, R., McCormick, G.P.: A Guide to SUMT-Version 4: The Computer Program Implementing the Sequential Unconstrained Minimization Technique for Nonlinear Programming. RAC-P-63, Research Analysis Corporation, McLean (1971)
23. Pugh, R.E.: A language for nonlinear programming problems. Math. Program. **2**, 176–206 (1972)
24. Fiacco, A.V., McCormick, G.P.: Nonlinear Programming: Sequential Unconstrained Minimization Techniques. Wiley, New York (1968)
25. McCormick, G.P.: Minimizing structured unconstrained functions. Technical Paper RAC-TP-277. Research Analysis Corporation, McLean, Virginia (October 1967)
26. Hascoët, L., Pascual, V.: Tapenade 2.1 user's guide. Technical Report 0300, INRIA (2004)
27. Hascoët, L.: Reversal strategies for adjoint algorithms. In: Bertot, Y., Huet, G., Lévy, J.-J., Plotkin, G. (eds.) From Semantics to Computer Science. Essays in Memory of Gilles Kahn, pp. 487–503. Cambridge University Press, New York (2009)

A Hybrid Implementation of Genetic Algorithm for Path Planning of Mobile Robots on FPGA

Adem Tuncer, Mehmet Yildirim and Kadir Erkan

Abstract This paper proposes a hybrid design and implementation of Genetic Algorithm (GA) for the path planning of mobile robots on a Field Programmable Gate Array (FPGA). GAs have been widely used to generate an optimal path by taking the advantage of its strong optimization ability; however, GA's computation time may be longer for complex problems. Especially, calculation of the fitness function takes a long time. A solution to accelerate it is to implement the GA in hardware. Intellectual Property (IP) hard core provides faster computation. In this study, fitness function of the GA is implemented on IP hard core while the other operators of GA run on a Microblaze soft processor. The experimental results showed that the fitness module by IP hard core can run 98.95 times faster than the fitness module by the Microblaze soft processor. The overall performance of the GA is accelerated 37.5 % by hybrid implementation with both hard and soft cores. We used the Pioneer P3-DX Mobile Robot and Xilinx XUPV5-LX110T FPGA device.

Keywords Genetic algorithms · FPGA · IP core · Microblaze · Path planning

1 Introduction

Path planning tries to find a feasible path for mobile robots to move from a starting node to a target node in an environment with obstacles [1]. There are so many methods that have been developed to overcome the path planning problem. Each method differs in their effectiveness depending on the type of application environment and each one of them has its own strengths and weaknesses. Compared to traditional search and optimization methods, such as calculus-based and enumerative strategies,

A. Tuncer (✉), M. Yildirim and K. Erkan
Networked Control Systems Laboratory, Kocaeli University,
41380 Kocaeli, Turkey
e-mail: adem.tuncer@kocaeli.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_47,

the evolutionary algorithms are robust, global and generally more straightforward to apply in situations where there is little or no prior knowledge about the problem to solve [2].

In the last decade, genetic algorithms have been widely used to generate the optimal path by taking the advantage of its strong optimization ability [3]. Genetic algorithms have been recognized as one of the most robust search techniques for complex and ill-behaved objective functions. The basic characteristic that makes the GA attractive in developing near-optimal solutions is that they are inherently parallel search techniques [4]. They can search all working environment simultaneously in a parallel manner and so they can reach a better solution more quickly.

However, computation time of GA may be longer for complex problems. A solution to accelerate it is to implement the GA in hardware. Due to pipelining, parallelization and no function call, a hardware implemented GA generates a really significant improvement in performance over a software GA [5]. The design of a general purpose GA should be flexible and easily configurable, especially the objective or cost function. These changes in configurations can easily be implemented in software but not with hardware. So hardware implementations of genetic algorithms had not been feasible until the field programmable gate arrays (FPGA) were developed [6].

The FPGA has gained its popularity in implementing hardware because of its low cost and fast design. This technology provides great flexibility to hardware designers. Designers can use FPGAs to create efficient hardware designs. The first Hardware based GA (HGA) was reported in 1995 [6].

Intellectual Property (IP) cores include two types which are hard and soft cores. Hard cores are physical manifestations of the IP design. Soft cores, which are more portable and flexible than hard cores, are logical existence as integrated circuit netlist or hardware description language code [7]. There are soft processors provided by various companies. For example, Picoblaze and Microblaze are soft cores designed for FPGA from Xilinx. In this study, the Microblaze soft core processor is used.

While the whole system can be implemented as a hard or soft core on FPGA, some parts of the system can also be implemented as soft (using C programming) and the rest of can be implemented as hard (using hardware description language, VHDL). Hardware description language allows the features of pipelining and parallelization to the architecture.

In this study, a hybrid design and implementation of GA for the path planning of autonomous mobile robots on an FPGA is proposed. A camera and digital image processing is used for localization of the mobile robot itself, obstacles and the target. The Pioneer P3-DX of Mobile Robots is used for indoor applications. The Pioneer P3-DX of Mobile Robots is a popular research robot that is programmable and suitable for classroom and laboratory use. So many researchers have used this robot in their study [8, 9]. A genetic algorithm is used for collision-free path planning. Grid-based environment model, which is frequently used in indoor applications, is used as the motion area of mobile robot [10]. Our previous system architecture in Ref. [10] is used; however an FPGA is employed for GA instead of a computer. All

operators of GA are designed and implemented in a hybrid fashion on a Microblaze soft processor and IP hard core.

2 Path Planning with Genetic Algorithms

2.1 Representation of Environment and Chromosome

Many path planning methods use a grid-based model to represent the environment space. It has been determined that calculation of distance and representation of obstacle are easier with grid-based representation. The grid-based environment space is represented in two ways, by the way of the coordinates plane [4, 11, 12] or by the way of orderly numbered grids [1, 12, 13]. Coordinates can be represented with both the binary or decimal numbers.

A chromosome represents a candidate solution [14] for the path planning problem. A chromosome or a path consists of a starting node, a target node and the hopping nodes which mobile robot passes over them. These nodes or steps in the path are called as genes of the chromosome. Different coding methods are used to create chromosomes, depending on the representation method of the environment. Binary coded string method [4, 15] is used in general, however decimal coded string method is also used [1, 13] and it is thought as to be more flexible. Decimal coding needs less computational overhead in time and space.

2.2 Fitness Function and Selection

The purpose of the path planning problem is to find an optimal path between a starting and a target node. Optimal path may be the shortest, the least time and energy requiring path to trip on it. Generally, in the path planning problems, the fitness function is considered as a shortest path. Fitness function value is defined as the sum of distances between nodes in a path.

In this study, the rank based fitness assignment is used instead of the proportional assignment method. This prevents a few better chromosomes to be dominant in the population. In the last step, chromosomes are selected according to fitness values and then put into a mating pool to produce new chromosomes.

2.3 Crossover and Mutation Operators

Generally, crossover combines the features of two parent chromosomes to form two offspring. Single-point crossover operator is used in this study. The genes of two

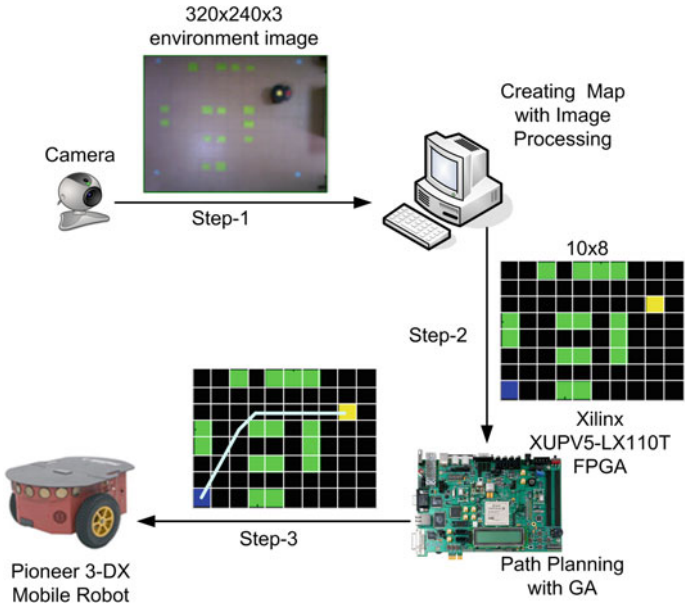


Fig. 1 Proposed motion planning system for mobile robot

chromosomes after the crossover point are swapped. All candidate chromosomes in the population are subjected to the random mutation after the crossover operation. This is a random bit-wise binary complement operation or a random small change in a gene, depends on the coding of chromosomes, applied uniformly to all genes of all individuals in the population with a probability of mutation rate. The mutation operation expands the search space to regions that may not be close to the current population, thus ensuring a global search [13]. Mutation operation increases the diversity of the population and avoids the premature convergence.

3 Motion Planning System with FPGA

Figure 1 shows the motion planning system used in this study. Localization is the determination of the positions of the mobile robot, the obstacles and the target. Generally positions are given by a user or identified by a camera. In the case of real-time and dynamic working, such as moving obstacles or targets are used in the environment, localization by means of a camera should be preferred.

Because real-time and dynamic applications are made in this study, a camera and image processing techniques are used for localization. The camera is mounted on the ceiling and it sends the real-time images of the environment to a computer. Each object is labeled with a different color on it; green for the obstacles, blue for

the target and yellow for the mobile robot. In order to determine the heading angle of the mobile robot, it is also labeled with a red color. Direction of the line that connects the centers of yellow and red colored circles on the robot equals to the heading angle of it. The overall system architecture is explained in our previous study Ref. [10]; however in this study, an FPGA is employed for GA instead of a computer. The benefits of using an FPGA are followed; first, the mobile robot has a limited carrying capacity and a computer consumes much of the capacity. Instead of a computer, carrying an FPGA is more advantages. The second, the FPGA is cheaper than a computer.

4 A Hybrid Implementation of Genetic Algorithm

A Field Programmable Gate Array (FPGA) consists of a programmable chip where the logic gates can be rearranged as needed by its user [16]. It provides great flexibility to hardware designers. While designers can use FPGAs to quickly create efficient hardware designs, many systems require a combination of both software and hardware [17]. Soft cores are more portable and flexible than hard cores, but they are slower on the contrary. Microblaze soft core is highly configurable; allowing the user to select a specific set of features required by user's design and has 32-bit general purpose registers, a 32-bit address bus. The Microblaze instruction execution is pipelined. For most instructions, each stage takes one clock cycle to complete [18]. The source code for the application can be written in high-level languages, such as C and C++ for Microblaze.

The advantage of Intellectual Property (IP) core is impressive acceleration in execution time due to the algorithms being executed in parallel in hardware and not sequentially as in software. In this study, Microblaze soft core and IP hard core are used together in a hybrid fashion.

The process that usually requires the most computation time in the evolutionary algorithm is the fitness evaluation [19]. In this study, the execution times of each GA operators were measured. It was observed that the fitness evaluation takes more time than other operators do. In order to speed up the fitness evaluation, its operation can be moved into hardware. Thus, while the other operators of the GA are still implemented over a Microblaze soft processor, the fitness evaluation is implemented in IP hard core. VHDL is used for IP hard core and the rest of the other parameters are written by using C programming.

Figure 2 shows the block diagram of our proposed hybrid implementation of GA system. It mainly consists of two blocks which are soft and hard cores. The On-chip Peripheral Bus (OPB) is used to integrate the IP core into the Microblaze soft processor. The OPB is a part of the IBM Core Connect™ on-chip bus standard. Control and state signals are used for communication between Microblaze and IP hard core. The control signal takes the possible values of "go" and "idle". Microblaze sends the map, which has starting, target nodes and obstacles, and the chromosomes to the IP hard core when the control signal is "idle". The IP hard core module starts

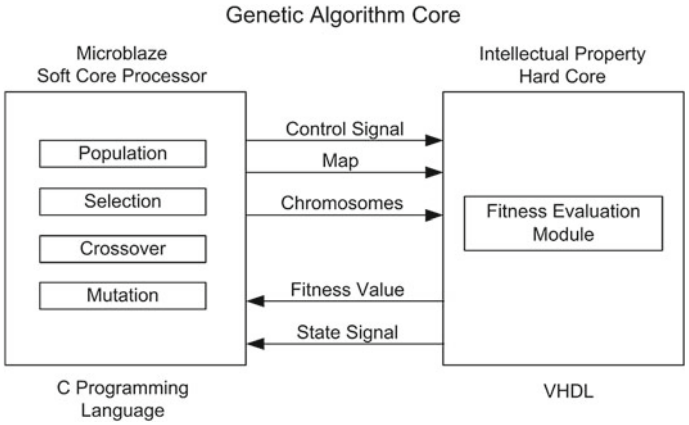


Fig. 2 Block diagram of proposed hybrid implementation

Table 1 Comparison of experimental results

	Computation time of a generation		
	Microblaze softy core	Hybrid design	Acceleration rate
Fitness function	840 µs	8.8 µs	98.95 times
All of the GA	0.16 s	0.1 s	37.5 %

to its fitness evaluation cycle when it receives the control signal as "go" from the Microblaze. The state signal takes the possible values of "ready" and "busy". While IP hard core evaluates the fitness, state signal takes "busy" value, after evaluation, it takes "ready" value to send the fitness values to Microblaze.

The experimental results are given in Table 1. According to the table, the fitness module by IP hard core can run 98.95 times faster than the fitness module by the Microblaze soft processor. The overall performance of the GA is accelerated 37.5 % by hybrid implementation with both hard and soft cores.

5 Conclusion

In this study, a hybrid design and implementation of GA for the path planning of mobile robots on an FPGA is proposed. The fitness function of the GA is based on IP hard core and the other operators of GA are implemented by software running over a Microblaze processor. Experiments are performed on the Pioneer P3-DX Mobile Robot and Xilinx XUPV5-LX110T FPGA device. The results showed that the IP hard core based fitness module can run faster than the fitness calculation on Microblaze processor. The overall performance of the GA is accelerated hybrid implementation with both hard and soft cores.

References

1. Hu, Y., Yang, S.X.: A knowledge based genetic algorithm for path planning of a mobile robot. In: Proceedings of the 2004 IEEE International Conference on Robotics and Automation, vol. 5, pp. 4350–4355 (2004)
2. Tu, J., Yang, S.X.: Genetic algorithm based path planning for a mobile robot. Robotics and automation. In: Proceedings ICRA '03 IEEE International Conference on Robotics and Automation, vol. 1, pp. 1221–1226 (2003)
3. Al-Taharwa, I., Sheta, A., Al-Weshah, M.: A mobile robot path planning using genetic algorithm in static environment. J. Comput. Sci. **4**, 341–344 (2008)
4. Elshamli, A., Abdullah, H.A., Areibi, S.: Genetic algorithm for dynamic path planning. In: Canadian Conference on Electrical and Computer Engineering, vol. 2, pp. 677–680 (2004)
5. Mostafa, H.E., Khadragi, A.I., Hanafi, Y.Y.: Hardware implementation of genetic algorithm on FPGA. In: 21th National Radio Science Conference, pp. 1–9 (2004)
6. Scott, S.D., Samal, A., Seth, S.: HGA: a hardware-based genetic algorithm. In: Proceedings of the Third International ACM Symposium on Field-Programmable Gate Arrays (FPGA'95), pp. 53–59 (1995)
7. Wang, J., Loo, S.M.: Case study of finite resource optimization in FPGA using genetic algorithm. Int. J. Comput. Appl. **17**(2), 95–101 (2010)
8. Chang, H.J., Lee, C.S.G., Lu, Y., Hu, Y.C.: P-SLAM: simultaneous localization and mapping with environmental-structure prediction. IEEE Trans. Robot. **23**(2), 281–293 (2007)
9. Teimoori, H., Savkin, A.V.: Equiangular navigation and guidance of a wheeled mobile robot based on range-only measurements. Robot. Auton. Syst. **58**(2), 203–215 (2010)
10. Tuncer, A., Yildirim, M., Erkan, K.: A motion planning system for mobile robots. Adv. Electr. Comput. Eng. **12**(1), 57–62 (2012)
11. Manikas, T.W., Ashenayi, K., Wainwright, R.L.: Genetic algorithms for autonomous robot navigation. IEEE Instrum. Meas. Mag. **10**(6), 26–31 (2007)
12. Tuncer, A., Yildirim, M.: Chromosome coding methods in genetic algorithm for path planning of mobile robots. In: 26th International Symposium of Computer and Information Sciences (ISCIS 2011), pp. 377–383 (2011)
13. Li, Q., Zhang, W., Yin, Y., Wang, Z., Liu, G.: An improved genetic algorithm of optimum path planning for mobile robots. In: Sixth International Conference on Intelligent Systems Design and Applications, ISDA '06, vol. 2, pp. 637–642 (2006)
14. Gelenbe, E., Liu, P., Lainé, J.: Genetic algorithms for route discovery. IEEE Trans. Syst. Man Cybern. Part B Cybern. **36**(6), 1247–1254 (2006)
15. Sugihara, K., Smith, J.: Genetic algorithms for adaptive motion planning of an autonomous mobil robot. In: Proceedings, IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA'97, pp. 138–143 (1997)
16. Allaire, F.C.J., Tarbouchi, M., Labonté, G., Fusina, G.: FPGA implementation of genetic algorithm for UAV real-time path planning. J. Intell. Robot. Syst. **54**, 495–510 (2009)
17. Lysecky, R., Vahid, F.: A study of the speedups and competitiveness of FPGA soft processor cores using dynamic hardware/software partitioning. In: Proceedings of the Design, Automation and Test in Europe, 2005, vol. 1, pp. 18–23 (2005)
18. XILINX Inc.: MicroBlaze processor reference guide. http://www.xilinx.com (2010). Accessed 16 Feb 2010
19. Gomez-Pulido, J.A., Vega-Rodriguez, M.A., Sanchez-Perez, J.M., Priem-Mendes, S., Carreira, V.: Accelerating floating-point fitness functions in evolutionary algorithms a FPGA-CPU-GPU performance comparison. Genet. Program. Evolvable Mach. **12**(4), 403–427 (2011)

Highly-Parallel Montgomery Multiplication for Multi-Core General-Purpose Microprocessors

Selçuk Baktir and Erkay Savaş

Abstract Popular public key algorithms such as RSA and Diffie-Hellman key exchange, and more advanced cryptographic schemes such as Paillier's and Damgård-Jurik's algorithms (with applications in private information retrieval), require efficient modular multiplication with large integers of size at least **1024** bits. Montgomery multiplication algorithm has proven successful for modular multiplication of large integers. While general purpose multi-core processors have become the mainstream on desktop as well as portable computers, utilization of their computing resources have been largely overlooked when it comes to performing computationally intensive cryptographic operations. In this work, we propose a new parallel Montgomery multiplication algorithm which exhibits up to **39** % better performance than the known best serial Montgomery multiplication variant for the bit-lengths of 2048 or larger. Furthermore, for bit-lengths of **4096** or larger, the proposed algorithm exhibits better performance by utilizing multiple cores available. It achieves speedups of up to **81** %, **3.37** times and **4.87** times for the used general-purpose microprocessors with **2**, **4** and **6** cores, respectively. To our knowledge, this is the first work that shows with actual implementation results that Montgomery multiplication can be practically and scalably parallelized on general-purpose multi-core processors.

Keywords Montgomery multiplication · RSA · Multi-core architectures · General-purpose microprocessors · Parallel algorithms

S. Baktir (✉)
Department of Computer Engineering, Bahçeşehir University,
Istanbul, Turkey
e-mail: selcuk.baktir@bahcesehir.edu.tr

E. Savaş
Faculty of Engineering and Natural Sciences, Sabanci University,
Istanbul, Turkey
e-mail: erkays@sabanciuniv.edu

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_48,

1 Introduction and Motivation

Many public key cryptosystems such as RSA, Diffie-Hellman, elliptic curve cryptography and recently pairing-based cryptography utilize multiplication as the most important operation which dominates the execution time. The efficiency of multiplication operation determines the practicality and in some cases the feasibility of cryptographic applications. Furthermore, due to the ever increasing need for higher security levels, developing faster multiplication algorithms for larger numbers becomes the focal point of many research activities.

Paillier encryption scheme [13], based on a setting similar to RSA, provides one of the most efficient and practical homomorphic encryption algorithms. However, it leads to message expansion after encryption. Damgård and Jurik [2] generalize this algorithm for applications that require multiple encryption such as computationally-private information retrieval (CPIR) [11] and multi-hop homomorphic encryption scheme that encrypts already encrypted messages introduced in the scenario given in [5]. Especially in CPIR [11], for instance, the binary tree which aims to privately extract one data item out of a total of 256 eventually leads to a modular multiplication where the modulus size is 8192-bit for 80-bit security. For the same application at 128-bit security level, we have to perform multiplication operations with numbers as large as 24576-bit.[1] It is crucial to be able to perform modular multiplication efficiently for such large operands and multi-core processors can be effectively put into use in executing parallelized multiplication operations for accelerating the aforementioned cryptographic applications. Emergence of multi-core processors on common desktop, notebook and server computers with no additional cost proclaim both the research opportunity and motivation for developing parallel algorithms for cryptographic applications. So far, the research on the subject has been focused on multi-core architectures [15], specifically built for multiplication operations of moderate size such as 1024 or 2048 bits, since inter-core communication dominates the overall computation in general-purpose multi-core processors for these bit lengths. However, we find out that this tendency starts changing for bit sizes of 2048-bit and higher if an efficient parallel multiplication algorithm is used. We present for the first time a practical and scalable parallel Montgomery multiplication algorithm [12] for general-purpose multi-core processors and present proof-of-concept implementation results.

2 Mathematical Background

In many cryptographic algorithms, a chain of multiplication operations need to be performed. The RSA algorithm [14] and the Diffie-Hellman key exchange scheme [3], and more recently the generalization of Paillier's probabilistic public-key scheme (with applications in private information retrieval) [2], require computing a sequence

[1] Recommendation for Key Management, Special Publication 800-57 Part 1 Rev. 3, NIST, 05/2011.

of modular multiplications of large integer operands, e.g. at least 1024 bits in length. In algorithms such as RSA, where the predefined modulus is a random number, the required modular reduction of the result of an integer multiplication is more costly than the multiplication itself. However, the Montgomery residue representation and the resulting Montgomery multiplication algorithm have proven useful in reducing this complexity [9, 12]. In Montgomery multiplication, firstly the operands are converted to their respective Montgomery residue representations, then the desired sequence of operations are performed using Montgomery multiplication, and finally the result is converted back to the normal integer representation. The Montgomery multiplication algorithm (given with Algorithm 1) computes $A \cdot B \cdot 2^{-m}$ for the input operands A and B which are the Montgomery residue representations of the two integers X and Y such that $A = X \cdot 2^m$ and $B = Y \cdot 2^m$. Note that Algorithm 1 keeps the residue representation intact, i.e., $A \cdot B \cdot 2^{-m} \equiv (X \cdot Y) \cdot 2^m \pmod{n}$ which allows for further computations avoiding extra operations. Algorithm 1 explains the general Montgomery multiplication algorithm. A detailed analysis of different Montgomery multiplication algorithms can be found in [10].

Algorithm 1 Montgomery multiplication

Input: $A, B \in Z_n$ where n is an odd integer, $n' = -n^{-1} \bmod 2^m$ where $m = \lceil \log_2 n \rceil$.
Output: $A \cdot B \cdot 2^{-m} \bmod n$.
1: $t \leftarrow A \cdot B$
2: $t \leftarrow (t + (t \cdot n' \bmod 2^m) \cdot n)/2^m$
3: **if** $t \geq n$ **then**
4: Return $t - n$
5: **else**
6: Return t
7: **end if**

Algorithm 2 CIOS method for Montgomery multiplication

Input: $A, B \in Z_n$ for n odd, $n' = -n^{-1} \bmod 2^{s \cdot w}$ for word size w and $s = \lceil \lceil \log_2 n \rceil / w \rceil$.
Output: $A \cdot B \cdot 2^{-s \cdot w} \bmod n$.
1: **for** $i = 0 \rightarrow s - 1$ **do**
2: $C \leftarrow 0$
3: **for** $j = 0 \rightarrow s - 1$ **do**
4: $(C, S) \leftarrow t_j + a_j \cdot b_i + C$
5: $t_j \leftarrow S$
6: **end for**
7: $t_s \leftarrow S$
8: $t_{s+1} \leftarrow C$
9: $C \leftarrow 0$
10: $m \leftarrow t_0 \cdot n'_0 \bmod 2^w$
11: $(C, S) \leftarrow t_0 + m \cdot n_0$
12: **for** $j = 1 \rightarrow s - 1$ **do**
13: $(C, S) \leftarrow t_j + m \cdot n_j + C$
14: $t_{j-1} \leftarrow S$
15: **end for**
16: $(C, S) \leftarrow t_s + C$
17: $t_{s-1} \leftarrow S$
18: $t_s \leftarrow t_{s+1} + C$
19: **end for**
20: **if** $[t_s\, t_{s-1}\, t_{s-1} \cdots t_0]_{2^w} \geq n$ **then**
21: Return $[t_s\, t_{s-1}\, t_{s-1} \cdots t_0]_{2^w} - n$
22: **else**
23: Return $[t_{s-1}\, t_{s-1} \cdots t_0]_{2^w}$
24: **end if**

Among all the Montgomery multiplication algorithms listed in [10], the CIOS method (Algorithm 2), requires the least storage and has the best timing performance, and therefore it is the most preferred Montgomery multiplication algorithm.

3 Parallel Montgomery Multiplication

All algorithms commonly proposed for Montgomery multiplication, including the CIOS algorithm and others listed in [10], are word based algorithms. They perform the required partial product computations and modular reductions interleaved together and on a word by word basis, yielding the serial nature of these algorithms. In order to parallelize Montgomery multiplication, for two and three core architectures, bipartite [6, 7] and tripartite [16] Montgomery multiplication algorithms, respectively, were proposed. In [1, 4, 15], specialized multi-core hardware architectures are proposed for parallel implementations. However they are intended for hardware based implementations and not targeted for general purpose microprocessors. In [1], the authors give a theoretical analysis of possible parallelizations of the SOS version of Montgomery multiplication given in [10], and implementation results on prototype multi-core systems using softcore processors on FPGA devices. The proposed design in [1] utilizes fast communication between the utilized softcores and local memories both of which are specifically tailored to the proposed parallel implementation. Therefore, their approach represents a hybrid architecture that takes advantage of both software and hardware. In this section, we propose a novel parallel Montgomery multiplication algorithm which is specifically designed for software realizations, and thus, suitable for general-purpose multi-core processors. For our parallel Montgomery multiplication algorithm, we exploit the inherent parallelism in integer multiplication. As shown in Fig. 1, integer multiplication has an inherent parallelism which could be exploited by running the multiple cores available on a processor in parallel. Here the partial products required for integer multiplication are computed in parallel and then accumulated to yield the actual product.

Algorithm 3 Parallel Integer Multiplication

Input: Integers $A = [a_{s-1}\ a_{s-2}\ \cdots\ a_0]_{2^d}$ and B of size $m = d \cdot s$ bits where s is the number of cores available.
Output: **ParallelMultiply**$(A, B) = A \cdot B$.
1: **for** $i = 0$ to $s - 1$ **do**
2: $t_i \leftarrow a_i \cdot B \cdot 2^{i \cdot d}$ {performed at core $i + 1$ in a multi-core implementation}
3: **end for**
4: **for** $i = 1$ to $s - 1$ **do**
5: $t_0 \leftarrow t_0 + t_i$
6: **end for**
7: Return (t_0)

We adapt Algorithm 3 to the original Montgomery multiplication algorithm (Algorithm 1) for application on multi-core processors. The resulting parallel Montgomery multiplication algorithm is presented with Algorithm 4.

Algorithm 4 Parallel Montgomery multiplication

Input: $A, B \in Z_n$ where n is an odd integer and $n' = -n^{-1} \bmod 2^m$ where $m = \lceil \log_2 n \rceil$.
Output: $A \cdot B \cdot 2^{-m} \bmod n$.
1: $t \leftarrow$ **ParallelMultiply**(A, B) {Algorithm 3}
2: $u \leftarrow$ **ParallelMultiply**$(t, n') \bmod 2^m$ {Algorithm 3}
3: $u \leftarrow$ **ParallelMultiply**(u, n) {Algorithm 3}
4: $u \leftarrow (u + t)/2^m$
5: **if** $u \geq n$ **then**
6: Return $(u - n)$
7: **else**
8: Return (u)
9: **end if**

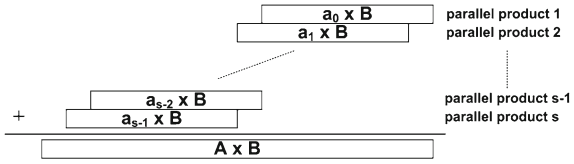


Fig. 1 Inherent parallelism in multiplication $\left(A = [a_{s-1}\ a_{s-2}\ \cdots\ a_0]\right.$ in base $\left. 2^{\lceil \frac{\log_2 A}{s} \rceil}\right)$

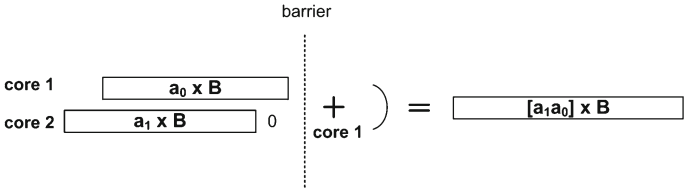


Fig. 2 Parallel integer multiplication on 2 cores

On a multi-core processor, one can also parallelize the additions given on lines 4 to 6 (required for the accumulation of the partial products) of Algorithm 3. When the number of cores available is a power of 2, this partial product accumulation can be achieved in a binary tree fashion, as shown below, with at most $\lceil \log_2 s \rceil$ steps where s is the number of cores available.

$$\begin{aligned} &\texttt{for}\ i = 1\ \texttt{to}\ \log_2 s\ \texttt{do} \\ &\quad \texttt{for}\ j = 0\ \texttt{to}\ \frac{s}{2^i} - 1\ \texttt{do} \\ &\quad\quad t_j \leftarrow t_j + t_{j+\frac{s}{2^i}} \end{aligned}$$

In the above setting, all the cores are exploited as evenly as possible with the maximal utilization, resulting in the minimal latency. However, this optimal chain of additions would not always be possible. Now we provide some addition chains for efficient implementations of Algorithm 4 on processors with 2, 4 and 6 cores. For performing the integer multiplication $A \times B$ on 2 cores, A is divided into two equal parts, as

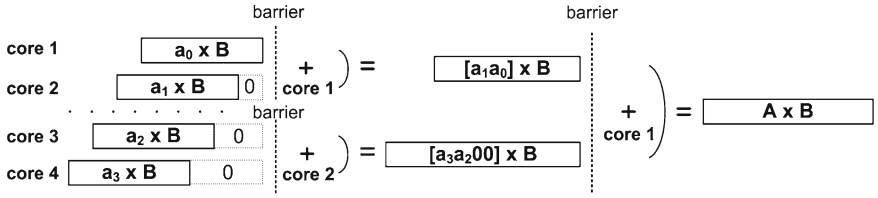


Fig. 3 Parallel integer multiplication on 4 cores

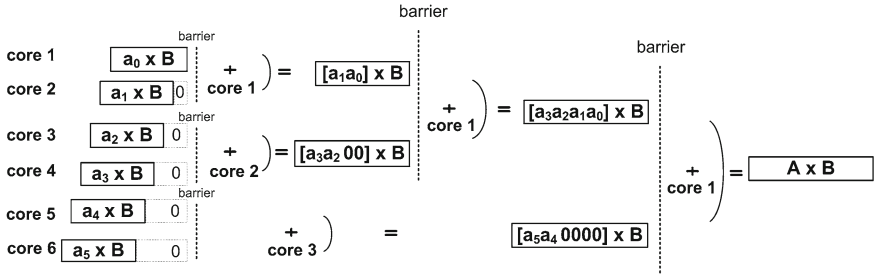


Fig. 4 Parallel integer multiplication on 6 cores

$[a_1\ a_0]$ in base $2^{\left\lceil \frac{\log_2 A}{2} \right\rceil}$, and the partial products $a_0 \times B$ and $a_1 \times B$ are computed simultaneously on separate cores. Finally, these partial products are accumulated as given in Fig. 2. Similarly, on a 4-core processor, for computing $A \times B$, A is divided into four equal parts, as $[a_3\ a_2\ a_1\ a_0]$ in base $2^{\left\lceil \frac{\log_2 A}{4} \right\rceil}$, and on a 6-core processor A is divided into six equal parts as $[a_5\ a_4\ a_3\ a_2\ a_1\ a_0]$ in base $2^{\left\lceil \frac{\log_2 A}{6} \right\rceil}$. The partial products are accumulated with the addition chains given with Figs. 3 and 4 for 4- and 6-core parallel implementations, respectively.

4 Timing Performance

In order to show the performance advantages of our algorithm in parallel implementations on general purpose multi-core processors, we implemented both the CIOS algorithm and our algorithm for the operand sizes of 1024, 2048, 4096, 8192, 16384 and 32768 bits. We used 2, 4 and 6 core general-purpose processors for our implementations, and made use of the efficient addition chains given with Figs. 2, 3 and 4, respectively. We utilized OpenMP (Open Multi-Processing)[2] for the parallel implementations of our algorithm. We would like to note that our implementations here

[2] OpenMP Tutorial at Supercomputing 2008, http://openmp.org/wp/2008/10/openmp-tutorial-at-supercomputing-2008/ (Last accessed on 26 February 2012).

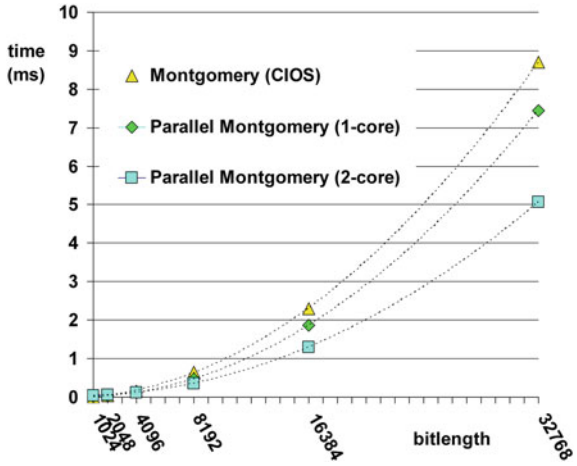


Fig. 5 2-core timings

are only proof-of-concept implementations. We coded in plain C and did not use any assembly or other low-level optimizations for either our algorithm or the CIOS algorithm. We wrote and compiled our code for 32-bit implementation (using only 32-bit instructions) even though some of the multi-core platforms we used in our tests are 64-bit processors. As our development platform, we used Microsoft Visual Studio 2010 10.0 and compiled our code in Release mode. The performances of both the CIOS algorithm and our algorithm could be improved similarly by utilizing low-level optimizations and 64-bit instructions where applicable. Furthermore, in our parallel Montgomery multiplication algorithm, we used simple *schoolbook* multiplication for computing the required partial products. Replacing these slow partial product computations with faster (subquadratic-complexity) algorithms such as Karatsuba [8] would result in further speedups in our algorithm. The timing graphs for our implementations on 2, 4 and 6 core general-purpose processors running Windows 7 Professional can be seen in Figs. 5, 6 and 7, respectively. Detailed timings and achieved speedups (compared to the CIOS method) can be found in Tables 1, 2 and 3.

We observe in Figs. 5, 6 and 7 that our algorithm performs significantly faster than the CIOS method, and furthermore efficiently utilizes multiple cores for improved performance, for growing operand sizes. As seen in Tables 1, 2 and 3, it achieves up to 81 %, 3.37 times and 4.87 times speedups for the used general-purpose microprocessors with 2, 4 and 6 cores, respectively. For the operand sizes of 2048 bit and smaller, the multi-core performance of our algorithm is worse than its single-core performance due to the overhead from using the OpenMP library.

On the 2 and 4 core processors, the single-core performance of our algorithm is better than the CIOS method for operand sizes of 2048 bit and larger (see Tables 1, 2). Whereas, on the 6 core processor, the single-core performance of our algorithm performs better starting with the larger operand size of 4096 bit (see Table 3). This

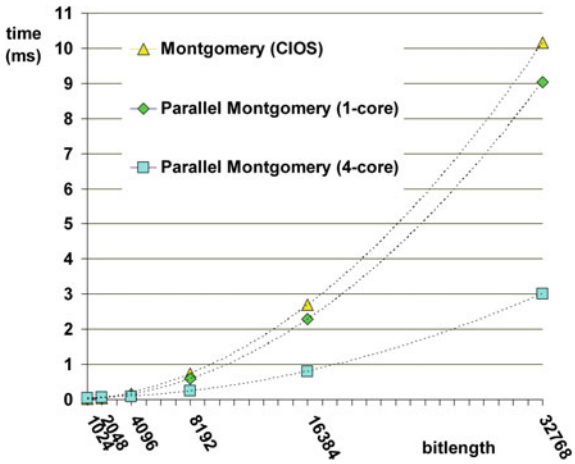


Fig. 6 4-core timings

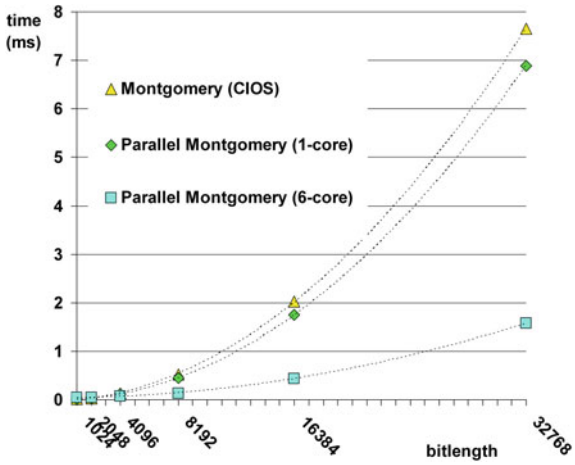


Fig. 7 6-core timings

discrepancy is most possibly due to the fact that the addition chains used for the partial product accumulations on 2 and 4 core processors (given in Figs. 2 and 3, respectively) are optimal whereas the one used for the 6-core processor (as given in Fig. 4) is not. On the used 2, 4 and 6 core general-purpose processors, the single-core performance of our algorithm is up to 39, 26 and 17 % better, respectively, compared to the CIOS method.

Table 1 Timings for Intel Dual-Core Pentium E6500 @2.93 GHz (2.96 GB RAM)

Operand size in bits		**1024**	**2048**	**4096**	**8192**	**16384**	**32768**
Algorithm	**time (ms)**						
Montgomery-CIOS (single core)		0.0091	0.0341	0.1488	0.6564	2.3043	8.7010
Parallel montgomery on single core		0.0121	0.0321	0.1222	0.4708	1.8675	7.4530
	Speedup (%)	**−25**	**6**	**22**	**39**	**23**	**17**
Parallel montgomery on 2 cores		0.0372	0.0635	0.1227	0.3620	1.2953	5.0640
(with OpenMP support)	**Speedup (%)**	**−75**	**−46**	**21**	**81**	**78**	**72**

Table 2 Timings for Intel Core 2 Quad Q8300 @2.5 GHz (4 GB RAM)

Operand size in bits		**1024**	**2048**	**4096**	**8192**	**16384**	**32768**
Algorithm	**time (ms)**						
Montgomery-CIOS (single core)		0.0103	0.0402	0.1692	0.7321	2.6929	10.1625
Parallel montgomery on single core		0.0111	0.0382	0.1499	0.5835	2.2798	9.0456
	Speedup (%)	**−7**	**5**	**13**	**26**	**18**	**12**
Parallel montgomery on 4 cores		0.0448	0.0554	0.0866	0.2394	0.8044	3.0140
(with OpenMP support)	**Speedup**	**−77** (%)	**−27** (%)	**95** (%)	**×3.06**	**×3.35**	**×3.37**

Table 3 Timings for Intel Xeon W3670 (6-core) @3.2 GHz (8 GB RAM)

Operand size in bits		**1024**	**2048**	**4096**	**8192**	**16384**	**32768**
Algorithm	**time (ms)**						
Montgomery-CIOS (single core)		0.0077	0.0298	0.1267	0.5168	2.0259	7.6476
Parallel montgomery on single core		0.0102	0.0314	0.1152	0.4432	1.7449	6.8921
	Speedup (%)	**−24**	**−5**	**10**	**17**	**16**	**11**
Parallel Montgomery on 6 Cores		0.0432	0.0381	0.0682	0.1331	0.4390	1.5712
(with OpenMP support)	**Speedup**	**−82** (%)	**−22** (%)	**86** (%)	**×3.88**	**×4.62**	**×4.87**

5 Conclusion and Future Work

We presented for the first time an efficient parallel and scalable Montgomery multiplication algorithm for software implementations on general-purpose multi-core processors. The speedups gained in our algorithm through parallelization are scalable with both the operand size and the number of available processor cores. We identify the improvement of our implementations though low-level optimizations such as coding in Assembly, exploiting the faster 64-bit instructions where available and utilization of sub-quadratic complexity multiplication algorithms for the required partial product computations, as future research. Furthermore, we plan on using our improved implementations for efficient applications of classical cryptographic schemes such as RSA and Diffie-Hellman, as well as more advanced schemes such the Damgård-Jurik's algorithm, in future work.

References

1. Chen, Z., Schaumont, P.: A parallel implementation of montgomery multiplication on multicore systems: algorithm, analysis, and prototype. IEEE Trans. Comput. **60**, 1692–1703 (2011)
2. Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In: Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography, PKC '01, pp. 119–136, London. Springer, London (2001)
3. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Inf. Theory **IT-22**, 644–654 (1976)
4. Fan, J., Sakiyama, K., Verbauwhede, I.: Montgomery modular multiplication algorithm on multi-core systems. 2007 IEEE Workshop Signal Process. Syst. **10**, 261–266 (2007)
5. Gentry, C., Halevi, S., Vaikuntanathan, V.: i-hop homomorphic encryption and rerandomizable yao circuits. In: Rabin, T. (ed.) CRYPTO Lecture Notes in Computer Science, vol. 6223, pp. 155–172. Springer, Heidelberg (2010)
6. Kaihara, M.E., Takagi, N.: Bipartite modular multiplication. In: Proceedings of Cryptographic Hardware and Embedded Systems—CHES 2005 Lecture notes in Computer Science, vol. 3659, pp. 201–210. Springer, Heidelberg (2005)
7. Kaihara, M.E., Takagi, N.: Bipartite modular multiplication method. IEEE Trans. Comput. **57**(2), 157–164 (2008)
8. Karatsuba, A., Ofman, Y.: Multiplication of multidigit numbers on automata. Sov. Phys. Dokl. (Engl. Transl.) **7**(7), 595–596 (1963)
9. Koç, Ç.K., Acar, T.: Montgomery multiplication in $GF(2^k)$. Des. Codes Cryptogr. **14**(1), 57–69 (1998)
10. Koç, Ç.K., Acar, T., Kaliski, B.: Analyzing and comparing montgomery multiplication algorithms. IEEE Micro **16**, 26–33 (1996)
11. Lipmaa, H.: First CPIR protocol with data-dependent computation. In: Proceedings of the 12th International Conference on Information Security and Cryptology, ICISC'09, pp. 193–210, Berlin. Springer, Heidelberg (2010)
12. Montgomery, P.L.: Modular multiplication without trial division. Math. Comput. **44**(170), 519–521 (1985)
13. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Advances in Cryptology—EUROCRYPT 1999, pp. 223–238. Springer, Heidelberg (1999)
14. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
15. Sakiyama, K., Batina, L., Preneel, B., Verbauwhede, I.: Multicore curve-based cryptoprocessor with reconfigurable modular arithmetic logic units over $GF(2^n)$. IEEE Trans. Comput. **56**, 1269–1282 (2007)
16. Sakiyama, K., Knezevic, M., Fan, J., Preneel, B., Verbauwhede, I.: Tripartite modular multiplication. Integration **44**(4), 259–269 (2011)

A Comparison of Acceptance Criteria for the Daily Car-Pooling Problem

Jerry Swan, John Drake, Ender Özcan, James Goulding and John Woodward

Abstract Previous work on the Daily Car-Pooling problem includes an algorithm that consists of greedy assignment alternating with random perturbation. In this study, we examine the effect of varying the move acceptance policy, specifically Late-acceptance criteria with and without reheating. Late acceptance-based move acceptance criteria were chosen because there is strong empirical evidence in the literature indicating their superiority. Late-acceptance compares the objective values of the current solution with one which was obtained at a fixed number of steps prior to the current step during the search process in order to make an acceptance decision. We observe that the Late-acceptance criteria also achieve superior results in over 75 % of cases for the Daily Car-Pooling problem, the majority of these results being statistically significant.

J. Swan (✉) · J. Drake · E. Özcan · J. Woodward
Automated Scheduling, Optimisation and Planning (ASAP) Research Group, Nottingham, UK
e-mail: jps@cs.nott.ac.uk

J. Drake
e-mail: jqd@cs.nott.ac.uk

E. Özcan
e-mail: exo@cs.nott.ac.uk

J. Woodward
e-mail: jrw@cs.nott.ac.uk

J. Goulding
School of Computer Science, Horizon Digital Economy Research Institute, University of Nottingham, Jubilee Campus, Wollaton Road, NG8 1BB Nottingham, UK
e-mail: jog@cs.nott.ac.uk

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_49,

1 Introduction

There is increasing economic and environmental interest in minimizing the consumption and emission of petrochemicals arising from the use of personal vehicular transport. The task of assigning passengers to drivers in order to increase vehicle occupancy while minimizing the additional journey length incurred is known as the Daily Car-Pooling problem (DCPP). The DCPP can be considered to be a generalization of the Dial-A-Ride Problem (DARP) in which the vehicles are heterogeneous [3]. As is the case with all variants of the Vehicle Routing Problem (VRP), it is known to be NP-hard [13]. We can consider the DCPP to be a VRP with the additional constraints of Pickup and Delivery (VRPPD) and Time Windows (VRPTW), the latter being an example of *quality of service* criteria in which we seek to minimize the inconvenience suffered by all participants.

The specific problem addressed in this paper is the *'To Work' DCPP*, in which a pool of users (employees) participate in vehicle sharing for travel to a central destination (the workplace). Some members of the pool are designated to be drivers (*servers*), the remainder are passengers (*clients*). All employees have a time window in which their journey must take place. Servers stipulate a maximum journey time and their vehicle has an associated capacity. Clients not assigned to any server have an associated penalty. The objective is to assign clients to servers so as to minimize the sum of the total distance travelled and the penalties incurred for unserviced clients. In the instances considered here, all quantities are integers. In [6], Cordeau and Laporte give an extensive overview of the DARP and observe that two decades of research has made it routinely possible to schedule hundreds of employees. Cordeau and Laporte also observe that approaches can be differentiated according as they perform clustering and routing as distinct, sequential phases or whether they interleave these activities. In Baldacci et al. [1], the DARP is solved by Lagrangean relaxation. If the location of all participants are known *a priori*, the DCCP is said to be *static*. We restrict ourselves to the static case. In [3], Calvo et al. give an algorithm for the DCPP (an adaptation of the capacitated p-median algorithm in [9]) that consists of greedy assignment alternating with random perturbation. The greedy assignment phase proceeds by seeking to minimize a marginal quantity termed *regret*—an estimate of the total extra mileage that would be incurred over all journeys for a passenger.

Maniezzo et al. [8] describe a solution to the Long-Term Car Pooling Problem (a variant in which the role of driver alternates between pool members) that employs Ant-Colony Optimization (ACO). ACO is an example of a *metaheuristic* technique that seeks to perturb candidate solutions beyond local optima. It is clearly also possible to apply other metaheuristic techniques such as Simulated Annealing, Evolutionary Strategies, Genetic Algorithms and Tabu Search.

Recent research [10] indicates that the choice of acceptance criterion is one of the more significant metaheuristic mechanisms. In this article, we examine the effect of varying the acceptance criterion. In particular, we investigate the use of *Late-acceptance Hillclimbing*. The Late-acceptance Hillclimbing (LA) metaheuristic [2] is a simple yet suprisingly effective strategy—a new solution is accepted if it is no

worse than the k-th most recent incumbent solution. The stated advantages of the LA are that it is reliant on only the single parameter k; it is not sensitive to initialisation and has been shown to be superior to (or at least competitive with) best-known results in a number of domains (e.g. [2, 11, 14]).

2 Experimental Framework

For our experiments, we made use of the HYPERION framework [12], implemented in the Java programming language. HYPERION provides a combinatorial optimization framework parameterized by concepts of STATE, LOCALITY (i.e. neighbourhood) and OBJECTIVE FUNCTION. We used datasets A and B as described in [1]. Class A is an adaptation of the datasets of [4, 5, 7] and consists of 12 problems with the number of employees ranging from 50 to 225. Class B is adapted from real-world data and consists of 23 problems with the number of employees ranging from 100 to 250. The authors state that both classes of problems are intended to simulate real-world applications.

We configured HYPERION with a STATE given by the pair (J, U) where J is the set of *Journies* (i.e. the set of assignments of clients to servers) and U is the set of unmatched clients. The LOCALITY is identical to that employed by Calvo et al., viz. the set of all states reachable from the present one via the unmatching of a single passenger. The objective function to minimize is then given by the total path cost plus the sum of the penalties incurred by unmatched passengers. We configured the framework with acceptance policies *Improving or Equal* (IE), *Late-acceptance* (LA) and *Late-acceptance with reheating* (LR). Figure 1 describes the top-level of the experimental framework: the essential difference from the pseudocode given in [3] is the addition of extension points to provide for the initialization, evaluation and internal-state update of the variant acceptance criteria. These extension points operate as follows: the IE policy requires no additional initialization or state-update, and accepts new values that are greater than or equal to the current value. The operation of the late-acceptance policies is derived from [2]: initialization of the late-acceptance policies involves the creation of a history list of fixed length, the values of which are given by o_1, the objective value obtained from phase 1 of the matching. Both late-acceptance policies accept a new value if it is better than or equal to either the current value or the k-th most recent value, where k is given by the iteration count modulo the length of the history list. An accepted solution replaces the solution to which it is being compared in the history list. Whenever the number of non-accepting iterations exceeds the threshold parameter *MAX_IDLE_ITER*, the internal-state update for LARH achieves reheating by adding an offset to all entries in the history list, which in our experiments was fixed at $0.1 * o_1$.

```
JourneyMatches hdcpp( List<Server> servers, List<Client> clients )
{
  // phase 1
  JourneyMatches matches = hdcppPhase1 ( servers, clients) ;
  // phase 2
  double bestValue = objectiveFn ( matches ) ;
  JourneyMatches bestMatching = matches ;

  initializeAcceptanceCriterion( bestValue ) ;

  long numUnimprovingIter = 0 ;
  for ( long iter = 0 ; ; ++iter )
  {
      JourneyMatches newMatches = hdcppPhase2InnerLoop( matches ) ;
      double currentValue = objectiveFn. valueOf( matches ) ;
      double newValue = objectiveFn. valueOf( newMatches ) ;
      boolean accept = acceptanceCriterion( currentValue, newValue, iter ) ;
      if ( accept )
      {
          matches = newMatches ;
          if ( currentValue < bestValue )
          {
              bestValue = currentValue ;
              bestMatching = matches ;
          }
      }
      if ( currentValue >= bestValue )
          ++numUnimprovingIter ;
      if ( terminationCondition(  matches, currentValue, newMatches,
                                  newValue, iter, numUnimprovingIter )  )
          break ;
      // update state of acceptanceCriterion
      // e.g. reheat etc. as appropriate
      acceptanceCriterion.update( ) ;
  }
}
```

Fig. 1 Algorithm DCPP

3 Results

Table 1 gives $(\bar{x}, \sigma^2)$ of the objective function value obtained with the IE, LA and LR acceptance criteria for 30 runs of the A and B datasets. The 'label' and 'size' columns give the instance name and number of employees, respectively. The termination criterion was 10,000 un-improving moves (decided experimentally). LA and LR have a history list length of 100, *MAX_IDLE_ITER* for LR was set to 200. Table 1 also compares the late-acceptance criteria for statistical significance (t-test with $p = 0.05$) against the IE criterion. For acceptance criteria A_1 and A_2, $A_1 \geq A_2$ indicates A_1 outperforms A_2 on average whilst $\gg$ indicates that this difference is statistically significant (conversely $\leq$ and $\ll$). It can be seen from these tables that there is relatively low variance in solution quality: this may be due to the constructive phase resulting in a basin of attraction in the solution-space.

Table 1 (Mean, standard deviation) and t-test comparison against IE criterion (vs) of objective function values of 30 runs of the datasets A and B by acceptance criterion

Label	Size	LA	Versus	LR	Versus	IE
A01	50	(1202,77)	$\geq$	(1190,54)	$\gg$	(1224,79)
A02	75	(1638,87)	$\leq$	(1619,62)	$\geq$	(1619,90)
A03	100	(1459,79)	$\geq$	(1461,104)	$\geq$	(1502,85)
A04	120	(2381,22)	$\gg$	(2387,34)	$\gg$	(2438,38)
A05	120	(2318,145)	$\ll$	(2253,123)	$\ll$	(2120,85)
A06	134	(2472,101)	$\leq$	(2453,123)	$\geq$	(2456,84)
A07	150	(2372,142)	$\geq$	(2353,144)	$\gg$	(2446,209)
A08	170	(2976,76)	$\leq$	(2972,61)	$\leq$	(2955,45)
A09	170	(2777,48)	$\geq$	(2770,57)	$\geq$	(2857,74)
A10	195	(3397,101)	$\ll$	(3357,95)	$\ll$	(3288,36)
A11	199	(2060,61)	$\gg$	(2040,63)	$\gg$	(2117,96)
A12	225	(2345,52)	$\gg$	(2335,43)	$\gg$	(2435,102)
B01	100	(1704,76)	$\gg$	(1718,80)	$\geq$	(1743,82)
B02	100	(1531,23)	$\geq$	(1531,27)	$\geq$	(1542,33)
B03	100	(1697,129)	$\leq$	(1615,109)	$\gg$	(1688,124)
B04	100	(2255,23)	$\leq$	(2255,32)	$\leq$	(2255,30)
B05	100	(1910,105)	$\gg$	(1932,107)	$\gg$	(2021,118)
B06	100	(1477,61)	$\geq$	(1464,84)	$\geq$	(1491,78)
B07	100	(1343,36)	$\gg$	(1360,33)	$\gg$	(1386,58)
B08	150	(2047,47)	$\gg$	(2036,39)	$\gg$	(2081,69)
B09	150	(1980,36)	$\gg$	(1987,37)	$\gg$	(2018,47)
B10	150	(2768,88)	$\geq$	(2787,93)	$\geq$	(2808,110)
B11	150	(2217,112)	$\gg$	(2254,112)	$\geq$	(2283,144)
B12	150	(1866,82)	$\geq$	(1855,90)	$\geq$	(1883,96)
B13	200	(2706,59)	$\gg$	(2691,99)	$\gg$	(2793,107)
B14	200	(2689,104)	$\geq$	(2703,94)	$\geq$	(2722,131)
B15	200	(3467,66)	$\gg$	(3463,69)	$\gg$	(3524,81)
B16	200	(3690,100)	$\leq$	(3686,111)	$\geq$	(3686,112)
B17	200	(4111,126)	$\gg$	(4135,128)	$\gg$	(4249,115)
B18	200	(2716,111)	$\gg$	(2750,77)	$\geq$	(2786,91)
B19	250	(3542,82)	$\gg$	(3566,79)	$\geq$	(3592,105)
B20	250	(3680,108)	$\geq$	(3633,103)	$\gg$	(3696,126)
B21	250	(3869,47)	$\gg$	(3872,58)	$\gg$	(4024,112)
B22	250	(3732,95)	$\gg$	(3707,91)	$\gg$	(3805,113)
B23	250	(3436,121)	$\gg$	(3404,119)	$\gg$	(3552,117)

LA outperforms IE in 77.1 % of instances, specifically 7 out of 12 A instances and 20 out of 23 B instances. 63 % of this performance difference is statistically significant (3 out of 7 A instances and 14 out of 20 B instances). LR outperforms IE in 88.6 % of instances, specifically 9 out of 12 A instances and 22 out of 23 B instances. 54.8 % of this performance difference is statistically significant (5 out of 9 A instances and 12 out of 22 B instances). LR is therefore performs particularly

well on dataset B. It is interesting that for instances A05 and A10 the IE strategy outperforms both late-acceptance criteria. Instance B03 is also interesting as LA performs worse than IE, but LR performs significantly better than it.

4 Conclusion

We have applied two variants of late-acceptance hillclimbing to a greedy algorithm for the Daily-Car Pooling Problem and compared them with naïve hillclimbing. Both late-acceptance strategies are superior to the naïve approach in most cases and this is often statistically significant for larger instances.

The superior performance of the reheating variant of late-acceptance can be explained in part by the 'fitness-cycling' effect of reheating, which generally means that there are a larger number of iterations before the 'number-of-unimproving-moves' termination criterion is met. Future work will attempt to gain further insight into the outlying instances (A05, A10, B03) and attempt to further distinguish between the two late-acceptance strategies: in particular, we anticipate that LR would perform significantly better than LA in almost all cases if the number of experiments were increased.

References

1. Baldacci, R., Maniezzo, V., Mingozzi, A.: An exact method for the car pooling problem based on Lagrangean column generation. Oper. Res. **52**(3), 422–439 (2004)
2. Burke, E.K., Bykov, Y.: A late acceptance strategy in Hill-Climbing for exam timetabling problems. In: PATAT '08 (2008)
3. Calvo, R.W., De Luigi, F., Haastrup, P., Maniezzo, V.: A distributed geographic information system for the daily car pooling problem. Comput. Oper. Res. **31**(13), 2263–2278 (2004)
4. Christofides, N., Eilon, S.: An algorithm for the vehicle dispatching problem. Oper. Res. Q. **20**(3), 309–318 (1969)
5. Christofides, N., Mingozzi, A., Toth, P.: The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (eds.) Combinatorial Optimization, pp. 315–338. Wiley, Chichester (1979)
6. Cordeau, J.F., Laporte, G.: The dial-a-ride problem (darp): variants, modeling issues and algorithms. 4OR **1**, 89–101 (2003)
7. Fisher, M.L.: Optimal solution of vehicle routing problems using minimum K-trees. Oper. Res. **42**(4), 626–642 (1994)
8. Maniezzo, V., Carbonaro, A., Hildmann, H.: An ANTS heuristic for the Long-Term Car-Pooling Problem. In: Onwuboulu, G., Babu, B. (eds.) New Optimization Techniques in Engineering. Springer, Heidelberg (2002)
9. Mulvey, J.M., Beck, M.P.: Solving capacitated clustering problems. Eur. J. Oper. Res. **18**(3), 339–348 (1984)
10. Özcan, E., Bilgin, B., Korkmaz, E.E.: A comprehensive analysis of hyper-heuristics. Intell. Data Anal. **12**(1), 3–23 (2008)
11. Özcan, E., Bykov, Y., Birben, M., Burke, E.K.: Examination timetabling using late acceptance hyper-heuristics. In: Proceedings of the Eleventh Conference on Congress on Evolutionary Computation (CEC'09), pp. 997–1004. IEEE Press, Piscataway, NJ, USA (2009)

12. Swan, J., Özcan, E., Kendall, G.: Hyperion—a recursive hyper-heuristic framework. In: Coello Coello, C.A. (ed.) 5th International Conference on Learning and Intelligent Optimization (LION 5), LNCS (2011)
13. Toth, P., Vigo, D.: An overview of vehicle routing problems. In: The vehicle routing problem, Society for Industrial and Applied Mathematics, pp. 1–26. Philadelphia, PA, USA (2001)
14. Verstichel, J., Berghe, G.V.: A late acceptance algorithm for the lock scheduling problem. In: Voss, S., Pahl, J., Schwarze, S. (eds.) Logistik Management, pp. 457–478. Physica-Verlag HD, Heidelberg (2009)

Part XII
Applications

A Monitoring System for Home-Based Physiotherapy Exercises

Ilktan Ar and Yusuf Sinan Akgul

Abstract This paper describes a robust, low-cost, vision based monitoring system for home-based physical therapy exercises (HPTE). Our system contains two different modules. The first module achieves exercise recognition by building representations of motion patterns, stance knowledge, and object usage information in gray-level and depth video sequences and then combines these representations in a generative Bayesian network. The second module estimates the repetition count in an exercise session by a novel approach. We created a dataset that contains 240 exercise sessions and tested our system on this dataset. At the end, we achieved very favourable recognition rates and encouraging results on the estimation of repetition counts.

1 Introduction

Physical therapy (or physiotherapy) is a medical science that concerns with the diagnosis and treatment of patients who have injuries or other problems that limit their capabilities to perform functional activities. Physical therapists provide care to patients by offering a treatment to reduce pain, prevent disability, and restore function. These treatments usually include physiotherapy exercises. However, human power, money, and time resources are not generally sufficient to do one-to-one sessions with all patients. These problems lead to HPTE and there is a need to monitor this type of treatment.

I. Ar (✉)
Kadir Has University, Cibali, 34083 Istanbul, Turkey
e-mail: ilktana@khas.edu.tr
URL: http://vision.gyte.edu.tr

Y. S. Akgul
Gebze Institute of Technology, Kocaeli, 41400 Gebze, Turkey
e-mail: akgul@bilmuh.gyte.edu.tri
URL: http://vision.gyte.edu.tr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_50,

Major achievements for human motion tracking systems for rehabilitation are surveyed by Zhou and Hu [9]. Soutscheck et al. [7] presented an automatic system to supervise and support rehabilitation and fitness exercises. Their solution outputs angular measurements of the knee joint by 2D and 3D tracking of knee positions using specialized sensors. Fitzgerald et al. [4] developed a system which utilizes ten inertial motion tracking sensors in a wearable body suit and a laptop/computer that communicates with this suit by Bluetooth connection. Jung et al. [5] developed a sensor driven motion tracking system to analyze upper body functions of a person.

In this paper, we propose a robust, low-cost, vision based monitoring system for HPTE. Instead of expensive systems which require specialized hardware as in the above works, the proposed system use a low-cost Microsoft Kinect sensor which contains a depth and an RGB camera. The novelty of this paper is two-fold. First, we define a generative Bayesian network which combines motion patterns, stance knowledge, and object usage information to recognize the exercise type in the given video sequence. Second, we develop an approach to estimate the repetition count of an exercise in the given session.

2 Dataset

We created a dataset of HPTE to demonstrate shoulder and knee exercises by consulting physiotherapists. A total of 240 exercise sessions (30 for each exercise type) are stored as gray-level and depth videos. In these exercise sessions, five volunteers performed eight exercises in six series. The exercise sessions are restricted to contain one actor performing one exercise repeatedly. Details of the exercises with sample frames are shown in Fig. 2.

The gray-level and depth videos are captured by Microsoft Kinect sensor with NI framework [6]. The resolution of videos are set to 320×240 pixels. The fps value is selected as 25. Frames of depth and gray-level videos are stored as 256 gray level images. Time duration of exercise sessions varies between 30 sec up to a minute. The depth sensor sometimes could not measure 11 bits per-pixel depth information due to reflection of surface etc (shown as black pixels in Fig. 2a, f, g). To solve this problem, we follow the same procedure as in [8].

3 The Monitoring System

The design of the monitoring system contains an exercise recognition and a repetition count estimator module; the former is responsible for the exercise recognition process and the latter is responsible for the estimation of the repetition count, as shown in Fig. 1. Exercise recognition module is divided into two parts: low-level and high-level.

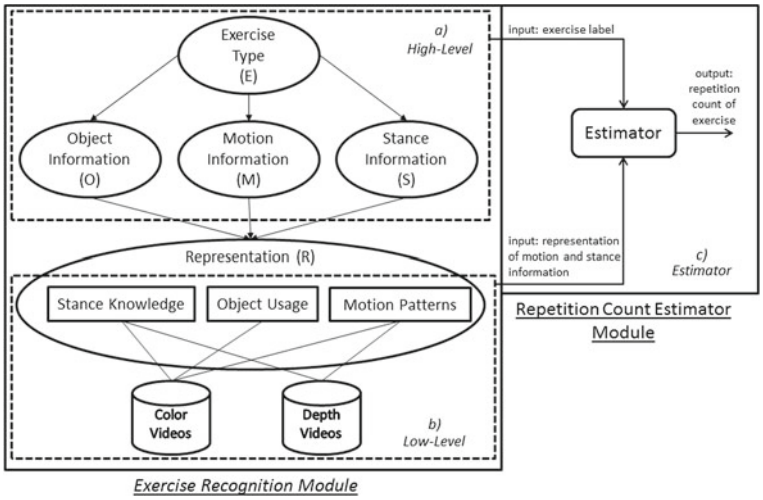


Fig. 1 The design of the monitoring system

We believe that the key patterns in an exercise video sequence are motion, stance, and object information. The low-level part builds representations of these key patterns by using gray-level and/or depth videos. Representations are clustered as R node in Fig. 1 to provide a better view of the graphical model. The high-level part contains of a generative Bayesian network which uses the graphical model in Fig. 1a to represent conditional independence relation between key patterns. The high-level part also benefits from the relations between object, stance, and motion information as described in Fig. 2.

The repetition count estimator module is dependent on the outputs of the exercise recognition module. This module gets exercise label, motion and stance representation as inputs and outputs the repetition count for the given exercise session.

4 The Low-Level Part of Exercise Recognition Module

The low-level part of exercise recognition module utilizes gray-level and depth videos to form representations of motion, stance, object information.

4.1 Representation of Motion Information

Motion information in videos is the main element of exercise recognition. Exercises have different motion patterns as in Fig. 2. To represent motion information in a video by motion patterns, we employed our previous method in [1]. First local motion information is obtained from depth, gray-level, or both videos by histogramming 3D

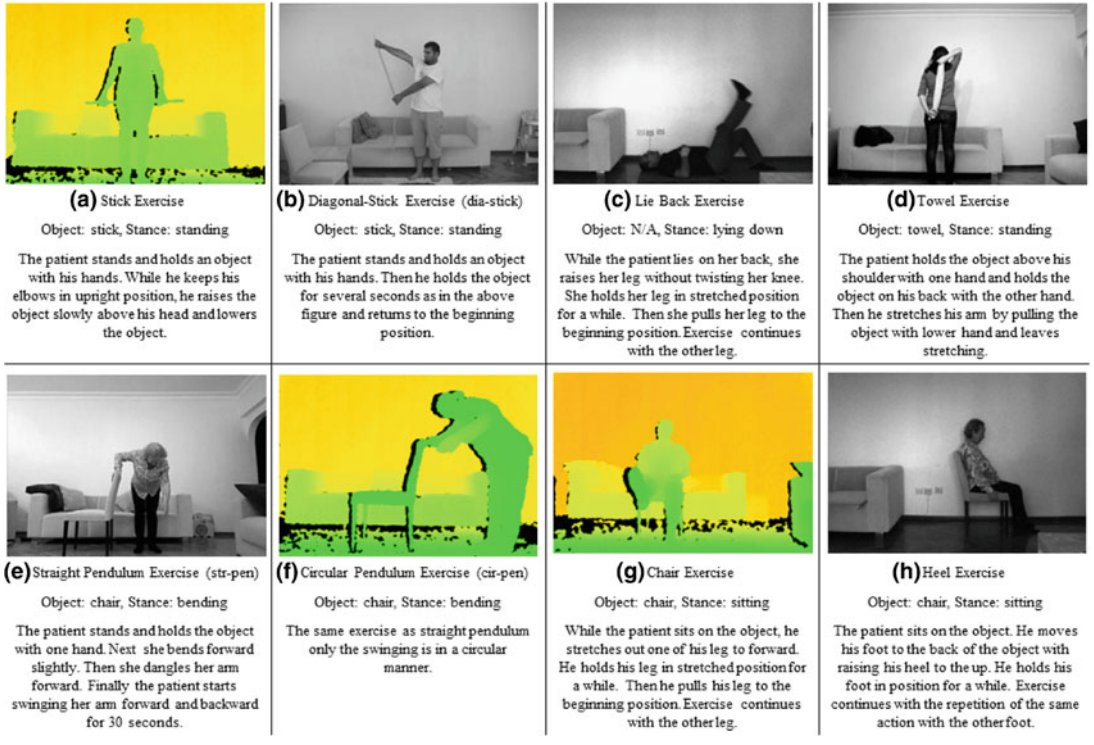


Fig. 2 Details of the exercises in HPTE dataset. In the each cell of the figure, the exercise types in the dataset are displayed with a sample figure followed by the related object, stance, and motion descriptions. The sample frames are taken from gray-level videos (**b**,**c**,**d**,**e**,**h**) and depth videos (**a**,**f**,**g**)

Haar-like features. Then statistical methods are used to describe motion information in the whole sequence as a global representation. We called the motion information for a given video sequence, which is obtained by our previous method, as MI.

4.2 Representation of Stance Information

Stance/pose information about an exercise session supports the other information sources when there are problems like occlusion, noise, high differences in temporal variances or etc.

First static background images are formed for a given depth and/or gray-level video by using the first few frames. Next a foreground extraction is performed for the selected frames (20 frame out of a video) of the given video and silhouette images are produced by thresholding. If both depth and gray-level videos are used, silhouette images are merged by the morphological union operation. Then, the largest blob in each silhouette image is windowed, these windows are parsed into 3 different grids with sizes 6×8, 8×8, and 8×6. Finally, the mean ratio of the foreground pixels in each cell of the each grid is calculated to form Stance Information SI vector.

4.3 Representation of Object Information

While most of the physiotherapy exercises include object interaction, object information in the video sequences reveals important clues about the type of these exercises.

First frames are selected at predefined uniform time-intervals (one frame out of 20 frames) in order to represent object information for a given video sequence. Then the object detection algorithm in [3], which uses bag of words models to detect objects, is adopted to check the availability of the corresponding object in the selected frames. The count of frames which includes the corresponding object are calculated and divided by the total number of selected frames. These ratios $OI(v, o)$ (where v is the video id, o is the object id) represent the object information in the video sequence.

5 The High-Level Part of Exercise Recognition Module

The high-level part of exercise recognition module aims to classify the exercise in the given video by using the representations obtained at the low-level part. We prefer to define a generative Bayesian network structure for the assignment of exercise label $e \in E$ to the video of the each exercise session because of the robustness of Bayesian networks for representing of joint distributions and encoding conditional independence assumptions.

The generative Bayesian network uses the graphical model in Fig. 1a to represent conditional independence relationships between random variables: exercise (E), object information (O), motion information (M), stance information (S), and Representation (R). Label assignment process $L(r)$ is defined as

$$L(r) = \underset{e \in E}{argmax} \sum_{S,M,O} P(E, S, M, O, R), \tag{1}$$

where r is the representation ($r \in R$) of the given video and $P(E, S, M, O, R)$ is the joint probability distribution table. $P(E, S, M, O, R)$ is defined by using the conditional dependencies in the graphical model (Fig. 1a) as

$$P(E, S, M, O, R) \propto P(E)P(S|E)P(M|E)P(O|E)P(R|S, M, O), \tag{2}$$

where $P(E) = 0.125$ (because of eight different exercises), $P(S|E)$, $P(M|E)$, and $P(O|E)$ terms can be calculated easily by Fig. 2. $P(R|S, M, O)$ term needs to be converted by using axioms as

$$P(R|S, M, O) = \frac{P(S, M, O|R)P(R)}{P(S, M, O)}. \tag{3}$$

$P(R)$ and $P(S, M, O)$ values in above equation are neglected because these are the same for any given exercise sessions. $P(S, M, O|R)$ is efficiently represented as

$$P(S, M, O|R) \propto P(S|R)P(M|R)P(O|R). \tag{4}$$

Finally, the values of $P(S|R)$, $P(M|R)$, and $P(O|R)$ are needed to calculate exercise label $L(r)$. $P(O|R)$ is equal to the $OI(v_r, o)$ obtained at the end of representation of object information process. $P(S|R)$ and $P(M|R)$ are related to SI and MI, respectively. For this relation, linear kernel Support Vector Machines (SVMs) are trained and then the Gibbs distribution is used to translate SVM scores into predictions.

6 Repetition Count Estimator Module

A HPTE session consists of a number of repetitions of the same exercise. It is important to record the repetition count for treatment analysis.

A new sub-global representation $SGR(\tau)$ for exercise session s is defined as

$$SGR(\tau) = MI_b(s_\tau)||SI_b(s_\tau), \tag{5}$$

where s_τ is the sub-sequence of s from frame 0 to τ, $MI_b(s)$ is the motion information about s by using both the gray-level and the depth video of s, and $SI_b(s)$ is the stance information about s by using both the gray-level and the depth video of s. The

Table 1 Exercise recognition module's recognition results on HPTE dataset

	Stick	Dia-stick	Lie back	Towel	Str-pen	Cir-pen	Chair	Heel
Stick	28/29	2/1	0/0	0/0	0/0	0/0	0/0	0/0
Dia-stick	3/1	27/29	0/0	0/0	0/0	0/0	0/0	0/0
Lie back	1/1	0/0	28/29	0/0	0/0	0/0	0/0	1/0
Towel	1/1	0/0	0/0	29/29	0/0	0/0	0/0	0/0
Str-pen	0/0	0/0	0/0	0/0	25/28	5/2	0/0	0/0
Cir-pen	0/0	0/0	0/0	0/0	3/1	27/29	0/0	0/0
Chair	0/0	0/0	0/0	0/0	2/1	1/0	27/29	0/0
Heel	0/0	0/0	1/0	0/0	1/0	0/0	1/1	27/29

In the table, x/y means that x is obtained without using depth videos, y is obtained using both gray-level and depth videos

exercise label for $GRS(\tau)$ is the same as $L(r)$, where r describes the representation of s, because s contains the same exercise $e \in E$ with different repetition counts. Confidence value (CV) for $SGR(\tau)$ is produced by using the remaining sessions in the dataset as training set and defining a new SVM formulation as

$$CV(\tau) = \sum_i a_i k(su_i, SGR(\tau)) + b, \tag{6}$$

where su_i describes the support vectors, a_i describes weights, b describes bias, and k describes the kernel function. It is important to mention that the training set are divided into two groups as $L(r)$ labeled videos and the others. Finally, the examination of CV with increasing τ indicates the repetition count. The count of zero crossings in the derivative of $\mathrm{CV}(\tau)$ with respect to τ would produce the exercise repetition counts.

7 Experimental Results

We evaluated our system with leave-one-actor-out procedure in each experiment and listed the results in the form of confusion matrix in Table 1.

The exercise recognition module successfully recognized 90.8 % of the 240 exercise sessions by using only gray-level videos. The most misclassified exercises were circular and straight pendulum exercises. There is a circular motion in circular pendulum exercise but this motion appears as a straight motion without depth information and caused misclassification.

The exercise recognition module successfully recognized 96.25 % of the 240 exercise sessions by using both gray-level and depth videos. The general misclassification error between straight and circular pendulum exercises was greatly reduced by using depth videos. As a baseline method [2] achieved 80.8 % recognition rate on HPTE dataset as the mean of the gray-level and depth sequences.

Repetition count estimator module estimated the repetition count of the 211 exercise sessions correctly with 88.0 % accuracy rate by using the ground truth labels (manually labeled). Using the obtained labels from exercise recognition module (with depth and gray-level videos), our module estimated the repetition count of 204 exercise sessions correctly with 85.0 % accuracy rate. The majority of incorrect estimation of repetition counts were observed in towel and circular pendulum exercise sessions.

8 Conclusions

In this paper, we propose a monitoring system for HPTE by using gray-level and depth videos obtained from the Microsoft Kinect sensor. The experimental results showed that the proposed system can effectively recognize the exercise in the given exercise session. We also observed that the monitoring system estimates the repetition count of the exercises in the given exercise sessions with encouraging results. To the best of our knowledge, the proposed system is the first system to monitor home-based exercises that includes objects by using a low-cost Microsoft Kinect sensor.

References

1. Ar, I., Akgul, Y.S.: A framework for combined recognition of actions and objects. In: International Conference on Computer Vision and Graphics, Warsaw, 2012
2. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. IEEE TPAMI (2001). doi:10.1109/34.910878
3. Fei-Fei, L.: Bag of words models: recognizing and learning object categories. In: CVPR Short Courses, Minnesota, 2007
4. Fitzgerald, D., Foody, J., Kelly, D., Ward, T., Markham, C., McDonald, J., Caulfield, B.: Development of a wearable motion capture suit and virtual reality biofeedback system for the instruction and analysis of sports rehabilitation. In: Proceedings of the 29th Annual International Conference of the IEEE EMBS, Lyon (2007)
5. Jung, Y., Kang, D., Kim, J.: Upper body motion tracking with inertial sensors. In: Proceedings of the 2010 IEEE International Conference on Robotics and Biomimetics, Tianjin (2010)
6. OpenNI, www.openni.org
7. Soutschek, S., Kornhuber, J., Maier, A., Bauer, S., Kugler, P., Hornegger, J., Bebenek, M., Steckmann, S., Stengel, S.V., Kemmler, W.: Measurement of angles in time-of-flight data for the automatic supervision of training exercises. In: 4th International Conference on Pervasive Computing Technologies for Healthcare, Munich (2010)
8. Xia, L., Chen, C.C., Aggarwal, J.K.: Human detection using depth information by kinect. In: Workshop on Human Activity Understanding from 3D Data in Conjunction with CVPR, Colorado Springs (2011)
9. Zhou, H., Hu, H.: Human motion tracking for rehabilitation-A survey. Biomed. Signal Process. Control (2008). doi:10.1016/j.bspc.2007.09.001

On the Use of Parallel Programming Techniques for Real-Time Scheduling Water Pumping Problems

David Ibarra and Josep Arnal

Abstract Most of the energy consumed by a water company is used for pumping systems. The electricity has a hourly pricing policy, therefore finding in real time the optimal schedule to operate those systems drastically reduces the power bill. Scheduling pumping problem with three main elements: pumping system, tank and water demand to be satisfied is analyzed. The mathematical programming model and techniques used to solve the problem, considering as known the water demand are exposed. Parallel programming paradigm is proposed to solve this problem when introducing stochastical programing techniques (scenario tree evaluation) and multi site problem. Mixing classical mathematical programing techniques and parallel tools (MPI and OpenMP), numerical experiments on parallel computers are designed and completed. As a result parallel programing strategy is experimentally proved as a useful technique to improve the real-time pumping scheduling problem solving.

Keywords Parallel systems · Water distribution systems · Pump scheduling · Optimization · Stochastic programming · Model predictive control · Mixed integer linear programing

1 Introduction

The drinking water supply in urban areas is usually divided by zones called district metered areas (DMA). These areas have their inputs and outputs pipes monitored in order to simplify management tasks such as leak isolation or water demand

D. Ibarra · J. Arnal (✉)
Departamento de Ciencia de la Computación e Inteligencia Artificial,
Universidad de Alicante, 03071 Alicante, Spain
e-mail: arnal@dccia.ua.es

D. Ibarra
Aquology Aquaambiente Servicios Integrales, Barcelona, Spain
e-mail: dibarra@aqualogy.net

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_51, © Springer-Verlag London 2013

prediction [1]. Usually DMA water supply is provided by a water tank placed in a higher place, this tank is filled using a pumping system that uses electricity which has a hourly pricing policy. The real-time scheduling pumping problem consists of decide in real-time the timetable for the pumping system for the next day/week, bearing in mind the pricing policy and other constraints in order to reduce electricity bill. The most important constraints are:

- to meet water demand,
- to respect tank levels for a given maximum and minimum,
- to extend pumping system lifespan. The pumping timetable should avoid turning on the pump for short periods of time.

Furthermore, as this problem should be solved on line and the decision system must be able to react to contingencies, it's required to solve this problem at least every hour to confirm the timetable.

Often this kind of decision system is found in literature as an engineering control problem, optimal control or Model Predictive Control (MPC) [2]. After checking different MPC software and realizing that they will fail to achieve real-life requirements, an expert system, including demand forecast and real-time optimization, was developed to achieve demands of experienced water network operator. The modeling scheme used is a direct approach according to decision variables [3]. The model is presented in the next section as Mixed Integer Program (MIP), however it goes further using short (minutes) scheduling periods, special integer variables and special operator constraints in contrast to simplified modeling of MPC. Moreover it's been operating on line since 2010, reducing 20 % of electricity costs. The used integer modeling techniques are found in general books about Operations Research [4–6].

Moreover, present paper is about evaluating computational tools to tackle, in real-time, Stochastical scenario tree evaluation [7] and multi-site problem using parallel programing paradigm. Experiments were performed in distributed memory multiprocessors and shared memory multiprocessors. Experiments were conducted in high performance machines with Unix operating systems but also in Windows OS environments. This was due to the fact that almost every real life available server to host a software to solve this problem at location uses Windows OS systems, and then it is necessary to build a software prototype portable to a Windows OS machine.

This paper has been organized as follows: The mathematical programming formulation of the problem, real-time needs and the use of the parallel programming paradigm are justified for both stochastic and multi problem settings in Sect. 2. Experimental results are shown in Sect. 3, explaining the mathematical programing libraries, parallel tools (OpenMP and MPI), particular problem and methods used for the experiments. Finally the conclusions are presented in Sect. 4.

2 Mathematical Programming Model and Parallel Settings

The scheduling problem, given a deterministic water demand, is modeled using mixed integer programing (MIP). On the next lines the main variables and equations of the problem are explained.

Let $T = \{1, 2, \ldots, t\}$ be the set of all scheduling periods. Let $x_i = 1$ if the pump is powered on the period i and $x_i = 0$ in other case, where $i \in T$. Let $c_i,\ \forall i \in T$ be the energy cost of the pump in the period i. This cost depends on the required power to power on the pumping system and the pricing police. Then the objective function of the problem is defined by

$$Min \sum_{i \in T} c_i x_i. \tag{1}$$

Let $V_{\max}$, $V_{\min}$ be the maximum and minimum allowed operational tank volume. Let V_{initial} the last known tank volume. Let $D_i, i \in T$ be the water demand at i period. Let Q the amount of water that the pumping system is able to pump in one period. Then the constraints to cover the demand (2) and to respect tank limits (3) are defined.

$$V_{\text{initial}} + Q \sum_{i<j} x_i - \sum_{i<j} D_i \geq V_{\min}, \quad \forall j \in T. \tag{2}$$

$$V_{\text{initial}} + Q \sum_{i<j} x_i - \sum_{i<j} D_i \leq V_{\max}, \quad \forall j \in T. \tag{3}$$

In order to accomplish the third condition using x_i, two special sets of **binary** variables $P_i, A_i, i \in T$ are defined to model the start and stop events. x_0 is also defined to represent the current status of the pumping system. These variables must satisfy

$$x_{i-1} - x_i = P_i - A_i, \quad \forall i \in T. \tag{4}$$

The left side of the equation will be 1 if a stop event happens and it will be -1 if a start event happens. On the right hand side the result is separate into the positive part P_i and the negative part A_i. As both variables must be binary, P_i models the stop event and A_i models the start event. Using these variables it's possible to reduce the start and stop events of the pumping system using equations like

$$P_i + A_i \leq 1, \quad \forall i \in T_j, \tag{5}$$

where T_j is a set of adjacent periods of time (e.g. if the periods are 5 min to reduce the number of start and stop events in periods of 15 min $T_j = \{j, j+1, j+2\}$). The number of periods in T_j depends on $|T|$ (the cardinality of T) and the mechanical characteristics of the pumping system. Furthermore, it is possible to include these variables in the objective function to reduce start and stop events.

The parallel programing paradigm is purposed for two different settings:

1. Stochastic Programing: $D(i)$ is an stochastic process but nowadays a single estimation is used. Real-time constraint does not allow to deal with different scenarios when using sequential programming. Parallel programming makes scenario tree evaluation possible. As a result it is possible to build a robust solution.
2. Solution server (Software As A Service, SaaS): One machine or HPC computer must find a timetable for several tanks in different water networks.
   Real-Time constraint: to find a solution for every scenario in minutes.

3 Numerical Results

With the object of solving the proposed set of MIP problems, COIN-OR [8] libraries were used. The reason is that these non-commercial libraries are available in C++ source code which is portable both to Unix and Windows. In addition solvers from COIN-OR were used successfully in the past to solve deterministic MIP problems like the one proposed.

Usually the methodology used to solve a MIP problem begins with a presolve process and ends with a search algorithm, like branch and bound/cut (B&B). Presolve process includes applying cutting planes that reduce the feasible region and consistently the search time. Heuristics techniques are also used during presolve phase in order to quickly produce a feasible solution.

A real-life problem including particular demand pattern D_i, initial values for $V_{\max}$, $V_{\min}$, V_{initial}, $|T|$ and start and stop constraints was chosen. This problem served as pattern to be solved in parallel up to 720 times per test, so the size of the problem is defined as the number of original problems to be solved. The original problem matrix has 55.000 non zero elements. The COIN-OR solver was set to avoid the preprocess and to start directly with the B&B process.

Numerical experiments were performed on both shared memory multiprocessors and distributed memory multiprocessors. The experiments started with shared memory computers (multicores) on Windows/Linux environments using OpenMP [9]. Afterwards an MPI [9] implementation was coded to be executed on distributed memory computers.

Specific experiments and development were carried out using up to four different machines and software settings which are included in the following list:

- Multicore 1: Intel Core Duo T7500 (2 cores), 2.2 GHz, 3 GB RAM memory, Windows XP. Microsoft Visual Studio 10.
- Multicore 2: Intel Core 2 QUAD 9400 (4 cores), 2.66 GHz, 3.5 GB RAM memory, Windows XP. Microsoft Visual Studio 10.
- Multicore 3: Intel Xeon CPU E5320 (8 cores), 1.86 GHz, 8 GB RAM, Linux Ubuntu 8.04.1. GCC compiler.

Table 1 Solution time in seconds

Multiprocessor	Problem size	Sequential	2 cores	4 cores
Multicore 1	2	794	570	
Multicore 2	4	1144	712	610

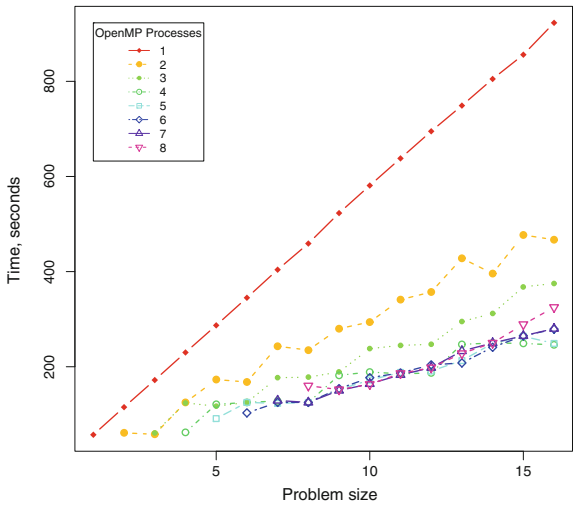


Fig. 1 Time to solve using Multicore 3

- Cluster: 26 Nodes: HP Proliant SL390s G7, 2 processor Intel XEON X5660 (12 cores per node), 2.8 GHz, 48 GB RAM memory per node, Linux CENTOS 5.6. GCC and Intel Compilers.

The first three are shared memory environments. The fourth environment could be used both as shared (single node use, up to 12 cores) or distributed memory machine.

Due to the practical issues exposed in the introduction, the first experiments were conducted using Windows and OpenMP with problem size 2 and 4. The results of these experiments are shown in Table 1 where the absolute time in seconds is presented for the sequential and parallel implementations on Multicore 1 and Multicore 2. The best relative time reduction is 46.7 % for Multicore 2 with a problem size of 4.

Figure 1 shows experiments on Multicore 3 using OpenMP and varying the size of the problem up to 16 scenarios and the number of cores used to 8.

Finally different experiments on the Cluster were completed using MPI. Figure 2 presents the results obtained using one node of the Cluster. Figure 3 presents the time needed to solve a problem of size equal to 720 with Cluster acting as a distributed memory system using between 13 and 239 cores (up to 20 nodes). Figure 4 presents time consumed by each MPI process for the same case of Fig. 3. Fixing solution time constraint to 900 s (for Windows), it's clear from the experiments that it's possible and appropriate to solve problems using parallel programing techniques. OpenMP allows to easily take advantage from a multicore processor.

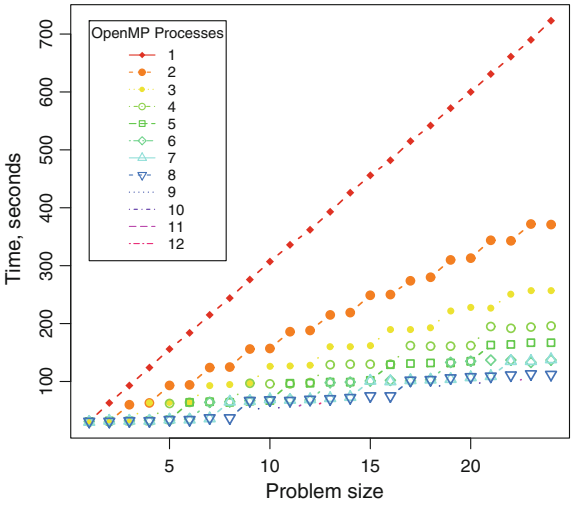


Fig. 2 Results from cluster: used as shared memory multicore, a single node from the cluster was used

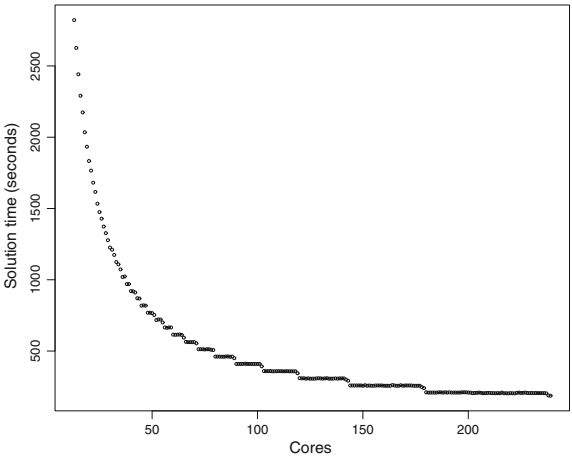


Fig. 3 Results from cluster: time to solve a 720 problem size

Table 1 illustrates that it's feasible to use Windows OS even with a laptop processor, although the use of Linux and server processors is the most suitable methodology (if possible).

Figure 1 reveals an unexpected behavior when using 7 or 8 cores on Multicore 3. It's slightly slower to solve a 16-Size problem using less cores. Nevertheless for Cluster (Fig. 2), which has more RAM memory, this doesn't happen.

Figure 4 reveals the importance of a good load balance policy among MPI processes. The more loaded the more time takes to complete the tasks. The best cases are obtained when the size of the problem, 720, is divided by the number of processes. In this case the amount of sub-problems is the same for every MPI process.

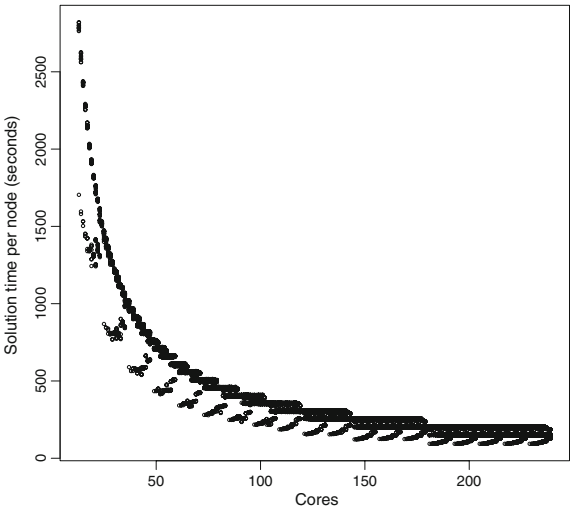


Fig. 4 Results from cluster: time to solve the assigned task for every MPI process for a 720 problem size

A real life issue is that for some problems it's not possible to find the optimal solution in the amount of time given for the real-time limitation, besides due to the use of B&B it's possible to give the best feasible solution when the time expires. This is because, after the preprocessing phase, a feasible solution is usually found in few seconds, but indeed it's not guaranteed that it's always found. In spite of this fact for this particular class of problems, given a 15 min time limit, a feasible solution was always found during more than a year of continuous use of the sequential version of the software.

4 Conclusions

We have proposed a mathematical programming model and techniques to solve the scheduling water pumping problem. We have introduced and evaluated parallel computational tools to tackle this model, solving, in real-time, stochastical scenario tree evaluation and multi-site problem. Experiments have been successful, and as a consequence it's expected to develop and integrate these techniques on the existing sequential software presented on the introduction. Furthermore the use of these techniques allows the implementation of the whole system avoiding commercial solver costs. Besides, results using Linux OS are clearly a good reason to use this OS instead of Windows for a solution server (SaaS).

Acknowledgments This work was funded by the Spanish Ministry of Science and Innovation (Project TIN2011-26254).

References

1. Farley, M., Trow, S.: Losses in Water Distribution Networks. IWA Publishing, London (2007)
2. Brdys, M.A., Ulanicki, B.: Operational Control of Water Systems: Structures, Algorithms, and Applications. Prentice Hall, Lebanon (1994)
3. Ormsbee, L.E., Lansey, K.E.: Optimal control of water pumping systems. J. Water Resour. Plann. Manag. **120**(2), 237–252 (1994)
4. Bazaraa, M.S., Jarvis, J.J., Sherali, H.D.: Linear Programming and Network Flows. Wiley, New Jersey (2011)
5. Eiselt, H.A., Sandblom C.L.: Integer Programming and Network Models. Springer, Berlin (2000)
6. Taha H.A.: Operations Research: An introduction. Prentice Hall, New Jersey (1997)
7. Birge, J.R., Louveaux, F.: Introduction to Stochastic Programming. Springer, New York (1997)
8. COmputational INfrastructure for Operations Research (COIN-OR). http://www.coin-or.org/
9. Quinn, M.J.: Parallel Programming in C with MPI and OpenMP. McGraw-Hill, Boston (2004)

Map Generation for CO_2 Cages

Dominique Barth, Boubkeur Boudaoud, François Couty, Olivier David, Franck Quessette and Sandrine Vial

Abstract This paper presents and proves a polynomial algorithm to generate particular planar maps (i.e., planar embeddings of planar graphs). These planar maps are constructed from a set of building blocks. Each building block is a molecule and the constructed planar maps are larger molecules that may have the properties of cages. Cages are molecules that can absorb smaller molecules such as Carbon dioxide.

1 Introduction

Carbon dioxide, as well as methane can be absorbed by large organic cages [4, 7]. These discrete architectures are formed by spontaneous assembly of small organic molecules bearing different reacting centres. The prediction of the overall shape of

Authors are supported by the Labex CHARM$_3$AT, the working group MODIMO and the CNRS GDR RO.

D. Barth, B. Boudaoud, F. Quessette and S. Vial (✉)
PRiSM—UMR CNRS 8144, University of Versailles, Versailles, France
e-mail: sandrine.vial@prism.uvsq.fr

B. Boudaoud
e-mail: boubkeur.boudaoud@uvsq.fr

F. Quessette
e-mail: franck.quessette@uvsq.fr

D. Barth
e-mail: dominique.barth@uvsq.fr

F. Couty · O. David
ILV—UMR CNRS 8081, University of Versailles, Versailles, France
e-mail: francois.couty@chimie.uvsq.fr

O. David
e-mail: olivier.david@chimie.uvsq.fr

E. Gelenbe and R. Lent (eds.), *Computer and Information Sciences III*,
DOI: 10.1007/978-1-4471-4594-3_52,

the cage that will be obtained by mixing of the starting blocks is rather difficult, especially because a given set of reacting partners can in principle lead to various architectures. It is hence crucial to have an operating tool that is capable of generating the many architectures potentially accessible from predetermined molecular modules facing to the combinatorial explosion of related graph generation problems. Thus, the aim of this paper is the generation of all planar maps (i.e., planar embeddings of planar graphs) representing possible molecules obtained from a set of elementary starting motifs. In this generation process, to reduce the combinatorial explosion, the objective is not to generate all possible maps, possibly many times each and then to make a selection using isomorphism detection. Our goal is to insure that each possible map is generated only once by using intermediate detections, to avoid duplication, based on a polynomial algorithm for isomorphism of planar maps. Tutte [8] in the 1960s had proposed the first works on planar graphs. After that, most of the works were non-constructive tools dealing with planar maps [1, 3].

There exist numerous works on enumeration and generation of planar maps [5, 8], but none of them deals with the generation of planar maps built with a set of starting motifs. In Sect. 2, we will define the objects we deal with, and we will present our isomorphism algorithm for our planar maps and in the last section we will present some numerical results.

2 Materials and Methods

In this section, we define *motif* and *map*, the objects we deal with. We also define the basic operations **concat**, **fold** that will be applied on maps to construct the maps we need for the chemical modeling. Maps are planar graphs and will be called planar maps [2, 3].

A *motif* m is the planar embedding of a star (i.e. a graph being a tree with one node called root connected to each other nodes called leaves). Leaves are labelled by letters in an alphabet $\mathcal{A}$. Alphabet $\mathcal{A}$ has $2n$ letters and is based on an alphabet with n letters and n complementary letters of each of these letters. In the following, we will denote by a a letter and $\overline{a}$ the complementary letter of a. In this context, $\overline{\overline{a}} = a$. So, $\mathcal{A} = \{a_1, a_2, \ldots, a_n\} \cup \{\overline{a_1}, \overline{a_2}, \ldots, \overline{a_n}\}$.

Considering the planar embedding of a motif m, we define a *clockwise order* $V_{out} = [u_0\ u_1\ ...\ u_{n-1}]$ of the leaves of m starting from an arbitrarily chosen leaf u_0. With this definition $[abc] = [bca] = [cab] \neq [cba] = [bac] = [acb]$. For a simple notation, the vertices will be named by their label, unless specified.

Motifs are the building blocks of the planar maps we will construct. A *planar map* is constructed from one motif or is obtained using two basic operations: the **concatenation** of two planar maps and the **folding** of one planar map to itself. Let us start by the definition of a planar map of one motif.

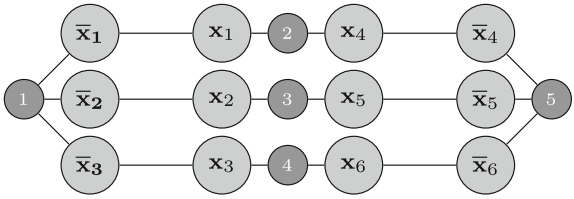


Fig. 1 Example of a planar map constructed from $\mathcal{M} = \{[x\ x], [\overline{x}\ \overline{x}\ \overline{x}]\}$

Definition 1 Let $C = (\mathcal{A}, M, V_{out}, E)$ be a *planar map* of one motif m such that $\mathcal{A}$ is the alphabet of m, $M = \{m\}$, V_{out} is the set of clockwise ordered labelled vertices of the motif and $E = \emptyset$.

E seems to be useless but will be meaningful in the two operations below. The operation of **concatenation** consists in connecting two planar maps.

Definition 2 We define the *concatenation* of two planar maps $C^{(1)} = (\mathcal{A}^{(1)}, M^{(1)}, V_{out}^{(1)} = [u_0 u_1 \dots u_l], E^{(1)})$ and $C^{(2)} = (\mathcal{A}^{(2)}, M^{(2)}, V_{out}^{(2)} = [v_0 v_1 \dots v_k], E^{(2)})$ on a couple of vertices (u_0, v_0), where $u_0 = \overline{v_0}$. We define $C = (\mathcal{A} = \mathcal{A}^{(1)} \cup \mathcal{A}^{(2)}, M = M^{(1)} \cup M^{(2)}, V_{out} = [v_1 \dots v_k u_1 \dots u_l], E = E^{(1)} \cup E^{(2)} \cup \{(u_0, v_0)\})$ the concatenation of $C^{(1)}$ and $C^{(2)}$ on (u_0, v_0). We note $C = \mathbf{concat}(C^{(1)}, u_0, C^{(2)}, v_0)$.

Note that the labelled vertices of a planar map have zero or one edge. A labelled vertex with no edge appears in V_{out} and not in E. A labelled vertex with one edge appears in E and not in V_{out}.

The operation of **folding** adds a new edge between two consecutive labelled vertices of the clockwise order V_{out}.

Definition 3 We define the *folding* of a planar map $C = (\mathcal{A}, M, V_{out} = [u_0 u_1 \dots u_{k-1}], E)$ on a vertex u_0, where $u_0 = \overline{u_1}$. We define $C' = (\mathcal{A}' = \mathcal{A}, M' = M, V'_{out} = [u_2 \dots u_{k-1}], E' = E \cup \{(u_0, u_1)\})$ as the folding of C on u_0 and we note $C' = \mathbf{fold}(C, u_0)$

The folding operation reduces the number of non connected labelled vertices in V_{out} by two and adds one edge to E.

The molecules we are interested in are represented by a planar map with all the labelled vertices connected and is simply defined by saturated planar map, i.e. a planar map with $V_{out} = [\ ]$. The final goal is to construct **all** the saturated planar maps made of a given number of motifs. The main difficulty is to detect isomorphic planar maps obtained by two different sets of operations. The algorithm given in this section is proven to construct all non-isomorphic saturated planar maps. The aim of this algorithm is to generate all the saturated planar maps made from a given number of *motifs* from a set $\mathcal{M}$. A planar map can use as many copies of motifs as needed.

For example, assume two motifs, named by their V_{out}, in $\mathcal{M} = \{[x\ x], [\overline{x}\ \overline{x}\ \overline{x}]\}$ we can construct the planar map of Fig. 1 with five motifs using $[x\ x]$ three times and $[\overline{x}\ \overline{x}\ \overline{x}]$ two times. The label have indices to distinguish them but all the x_i are equal to x and all the $\overline{x}_j$ are equal to $\overline{x}$.

The detection of isomorphic planar maps is solved by using signatures of planar maps. The definitions, properties and algorithms of isomorphism and signature are given in Sect. 3. Before we give the proof in Sect. 3, we claim that two different signatures identify two different planar maps.

$\mathcal{C}_n$ is the set of all the saturated and non saturated planar maps with n motifs. The GENERATE_MAP algorithm below generates $\mathcal{C}_n$ from $\mathcal{C}_{n-1}$. In $\mathcal{C}_n$ the planar maps and the signatures are stored. To manipulate the planar map already stored, we assume the following functions exist. **is_new**(S) returns true if the signature S is NOT already stored and false otherwise; **get_signature**(C) returns the signature of the stored planar map C.

If $n = 1$ GENERATE_MAP only consists in adding the planar map made of one motif to $\mathcal{C}_1$. If $n > 1$, first all the possible concat of a motif are tested and stored if this concatenation does not already exist, second all the possible folding are tested and stored by calling FOLD_MAP.

Let us now give the FOLD_MAP algorithm.

ALGORITHM FOLD_MAP (C : planar map with n motifs)
BEGIN
FOR ALL ($u_0 \in V_{out} = [u_0 u_1 ... u_k]$ of C such that $u_0 = \overline{u_1}$) DO
⌈$C' :=$ **fold**(C, u)
|$S' :=$ SIGNATURE(C')
|IF (**is_new**(S')) THEN
| ⌈$\mathcal{C}_n = \mathcal{C}_n \cup \{(C', S')\}$
| ⌊IF ($V'_{out} \neq [\,]$) THEN FOLD_MAP(C')
END

This recursive algorithm stops when V_{out} is empty. Since the recursion comes after the folding operation and since the folding operation reduces the size of V_{out}, it proves that this algorithm always ends.

3 Isomorphism and Signature

We will now define isomorphism and signature of planar maps, give an algorithm to compute signature, prove the uniqueness of signature and prove the polynomial complexity of this computation.

Definition 4 Two planar maps $C^{(1)}$ and $C^{(2)}$ are isomorphic if there exists a bijection f between the vertices of $C^{(1)}$ and $C^{(2)}$ such that:

- $\mathcal{A}^{(1)} = \mathcal{A}^{(2)}$
- $M^{(1)} = M^{(2)}$;
- if $u \in C^{(1)}$ is a root of a star, $f(u)$ is a root of a star in $C^{(2)}$;
- if ${V_{out}}^{(1)} = [u_1 u_2 ... u_k]$, ${V_{out}}^{(2)} = [f(u_1) f(u_2) ... f(u_k)]$;
- $(u, v) \in E^{(1)}$ if and only if $(f(u), f(v)) \in E^{(2)}$.

To deal with the isomorphism, the standard method is to compute a signature for planar maps that satisfies the following properties:

Property 1 Let $C^{(1)}$ and $C^{(2)}$ be two planar maps and $S^{(1)} = \text{SIGNATURE}(C^{(1)})$, $S^{(2)} = \text{SIGNATURE}(C^{(2)})$:

$$S^{(1)} = S^{(2)} \text{ if and only if } C^{(1)} \text{ and } C^{(2)} \text{ are isomorphic.}$$

In general this computation is, in the worst case, non polynomial [6]. Since we deal with a restricted family of graphs, planar maps constructed from particular motifs, we prove that signature computation can be made in polynomial time in the worst case. First we need to give a specific graph search.

Given a planar map C, a motif m of C and a labelled vertex u of m, the following ordered depth first search algorithm gives an ordered list of pairs (motifs, labelled vertex).

```
ALGORITHM ODFS(C : planar map, m : motif of C, u : labelled vertex of m) :
                                        ordered list of pairs (motif, labelled vertex)
BEGIN
P = empty stack
L = empty ordered list of pairs
stack in P all the labelled vertices of m, in the order of V_out, starting with u
append (m, u) to L
WHILE (P not empty) DO
        ⌈x = unstack P
        |IF (x is not in V_out of C) THEN
        |   ⌈y = unique labelled vertex neighbor of x
        |   |m_y = motif of y
        |   |IF (m_y not in L) THEN
        |   |   ⌈stack in P all the labelled vertices of m_y, in the order of V_out of
        |   |   |      m_y, starting with y
        |   ⌊   ⌊append (m_y, y) to L
        ⌊
RETURN L
END
```

The ordered pair list L computed by Ordered Depth First Search algorithm contains each motif of C once. It gives an order on the motifs of C. Each labelled vertex in L gives a starting vertex for clockwise ordering of each motif. Thus, it gives a total order on the labelled vertices of C. L satisfies the following property.

Property 2 Given C, m and u, the list $L_{C,m,u}$ obtained by ODFS algorithm is unique.

Proof Since the stacking in P is done according the order of V_{out}, the search order in a motif is fixed. Since a labelled vertex in a motif has at most one labelled neighbor in another motif, the search order between motifs is also fixed. Thus the whole algorithm is deterministic and L is unique. □

From an ordered pair list $L_{C,m,u}$, we can define a signature of an ordered pair list denoted SIGNATURE($L_{C,m,u}$) by:

Definition 5 Let $L_{C,m,u}$ be an ordered list computed by ODFS algorithm. The signature of $L_{C,m,u}$ is twofold and we note SIGNATURE($L_{C,m,u}$) = (MM, PP):

1. Construct an ordered list MM of V_{out} of model motifs. Each motif in $L_{C,m,u}$ corresponds to a V_{out} in MM. The order in MM is the same with the one in $L_{C,m,u}$.
2. Construct an adjacency matrix PP of the labelled vertices of C with rows and columns order given by $L_{C,m,u}$:
   - for all (m_x, x) before (m_y, y) in $L_{C,m,u}$ all the rows and columns indices of vertices of m_x are before those of m_y in PP.
   - for all (m_x, x) in $L_{C,m,u}$ with $V_{out} = [x, x_1, x_2, \ldots, x_k]$, the rows and columns indices are in the order $x, x_1, x_2, \ldots, x_k$ in PP.
   - for all vertex with index i in PP labelled by x connected to a vertex with index j in PP labelled by $\overline{x}$, $PP[i][j] = x$ and $PP[j][i] = \overline{x}$;
   - if two vertices of indices i and j are not connected $PP[i][j] = PP[j][i] = 0$.

Assuming a total order on the model motifs of $\mathcal{M}$ and a total order on the label of the alphabet $\mathcal{A}$ we can define a total order on the SIGNATURE of ordered pair list of the same planar map C:

Definition 6 Let $L^{(1)} = L_{C,m^{(1)},u_1}$ and $L^{(2)} = L_{C,m^{(2)},u_2}$ two ordered pair lists of a planar map C. Let $(MM^{(1)}, PP^{(1)}) =$ SIGNATURE($L^{(1)}$) and $(MM^{(2)}, PP^{(2)}) =$ SIGNATURE($L^{(2)}$). An order on the models motifs of $\mathcal{M}$ gives an order between $MM^{(1)}$ and $MM^{(2)}$. An order on the label of the alphabet $\mathcal{A} \cup \{0\}$ gives an order between $PP^{(1)}$ and $PP^{(2)}$ by reading the matrices row by row.

- if $MM^{(1)} < MM^{(2)}$ then SIGNATURE($L^{(1)}$) < SIGNATURE($L^{(2)}$)
- if $MM^{(1)} = MM^{(2)}$ and $PP^{(1)} < PP^{(2)}$ then SIGNATURE($L^{(1)}$) < SIGNATURE($L^{(2)}$)
- if $MM^{(1)} = MM^{(2)}$ and $PP^{(1)} = PP^{(2)}$ then SIGNATURE($L^{(1)}$) = SIGNATURE($L^{(2)}$)

This order is a total order since it is based on total orders.

Definition 7 A *signature* of a planar map C is defined by SIGNATURE(C) = $\min_{m \in C, u \in m}$ {SIGNATURE($L_{C,m,u}$)}

Property 3 For all planar map C, SIGNATURE(C) is unique.

Proof For a given motif m of C and a given labelled vertex u of m, $L_{C,m,u}$ is unique. Since ODFS algorithm is deterministic SIGNATURE($L_{C,m,u}$) is unique. Since there exists a total order on all the signatures of all the ordered pair list L of C, the minimum is unique. □

Note that we may have $L_{C,m^{(1)},u_1} \neq L_{C,m^{(2)},u_2}$ but $\text{SIGNATURE}(L_{C,m^{(1)},u_1}) = \text{SIGNATURE}(L_{C,m^{(2)},u_2})$. In this case, it means that there exists an automorphism in the planar map.

Property 4 For all planar map $C^{(1)}$ and $C^{(2)}$ on the same alphabet $\mathcal{A}$, $\text{SIGNATURE}(C^{(1)}) = \text{SIGNATURE}(C^{(2)})$ if and only if $C^{(1)}$ is isomorphic to $C^{(2)}$.

Proof Let $C^{(1)}$ and $C^{(2)}$ be isomorphic. Let (m, u) be a starting vertex of ODFS algorithm that gives minimum signature for $C^{(1)}$. The signature of a planar map is the minimum over all the possible starting vertex for ODFS algorithm. In $C^{(2)}$ the ODFS algorithm will also be run starting on (m, u) and will give the same signature.

Let $C^{(1)} = (\mathcal{A}, M^{(1)}, {V_{out}}^{(1)}, E^{(1)})$ and $C^{(2)} = (\mathcal{A}, M^{(2)}, {V_{out}}^{(2)}, E^{(2)})$. If $\text{SIGNATURE}(C^{(1)}) = (MM^{(1)}, PP^{(1)}) = \text{SIGNATURE}(C^{(2)}) = (MM^{(2)}, PP^{(2)})$. We have $MM^{(1)} = MM^{(2)}$ and then $C^{(1)}$ and $C^{(2)}$ are constituted with the same motifs and $M^{(1)} = M^{(2)}$. Since $PP^{(1)} = PP^{(2)}$, and since PP matrices store the connected labelled edges, we have $E^{(1)} = E^{(2)}$. Each labelled vertex is connected to 0 or 1 other labelled vertex. If a labelled vertex is connected, it is not in V_{out}, if not, it is in V_{out}. Since $E^{(1)} = E^{(2)}$, the labelled vertices in ${V_{out}}^{(1)}$ and ${V_{out}}^{(2)}$ are the same. For any signature, the order of the labelled vertices in V_{out} is the same with their relative order in PP since in the construction of PP, the clockwise order of each motif is preserved. Thus, ${V_{out}}^{(1)} = {V_{out}}^{(2)}$. □

Property 5 For all planar map C, $\text{SIGNATURE}(C)$ is computed in polynomial time.

Proof If we note E the number of labelled edges in a planar map C, computing an ordered list with ODFS algorithm starting from any (m, u) goes through each edge at most two times. Computing a signature from an ordered list needs to compute the PP matrix of size E^2. Computing a signature for a planar map needs to compute an ordered list starting from each labelled vertex. The comparison between signature needs E^2 comparisons for the comparison of the PP matrices. Thus the global computation is at worst in $\mathcal{O}(E^3)$.

4 Numerical Results and Conclusion

In this section, we summarize some of our results. For example, we use two models of motifs ($[x, x, x]$ and $[\overline{x}, \overline{x}, \overline{x}, \overline{x}]$), and we fix the maximum number of motifs. The number of saturated maps generated is 6 with 13 motifs and 401 with 14 motifs.

In Fig. 2, we show the number of saturated and non saturated planar maps generated in function of the number of motifs used. Note that the number of non saturated maps is linear in logscale. In this last case, we use three models of motif : $[xx]$, $[\overline{x}yy]$ and $[\overline{xyy}]$.

We have proved a polynomial algorithm to generate all the saturated maps made of a given number of motifs. Nevertheless, as numerical results show, we need to construct an exponential number of non-saturated maps. Future works are dedicated

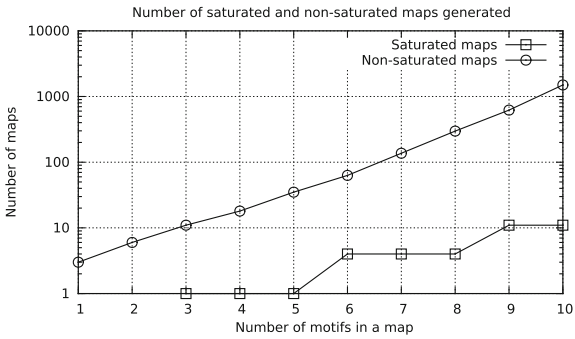


Fig. 2 Number of planar maps generated in function of the number of motifs

to reduce this number of non-saturated maps. Two main directions arise, taking into account chemical properties and amelioration on algorithms. From a chemical point of view, the planar maps must be fold on a sphere and since each motif have a particular shape, the concatenation of too numerous flat motifs may be of no use. Since the concatenation and folding operation applied on a map may be inverted to lead to the same result, it may be interesting to prove such general properties and enhance our algorithms.

References

1. Arquès, D.: Les hypercartes planaires sont des arbres très bien étiquetés. Discrete Maths **58**(1), 11–24 (1986)
2. Bouttier, J., Di Francesco, P., Guitter, E.: Planar maps as labeled mobiles. Elec. J Comb. **11**, 477–499 (2004)
3. Cori, R., Vauquelin, B.: Planar maps are labelled trees. Can. J. Math. **33**(5), 1023–1042 (1981)
4. Holst, J., Trewin, A., Cooper, A.: Porous organic molecules. Nat. Chem. **2**, 915–920 (2010)
5. Liskovets, V.: Enumeration of nonisomorphic planar maps. Sel. Math. Soviet. **4**, 304–323 (1985)
6. McKay, B.: Practical graph isomorphism. In: Numerantium, C. (ed.) 10th Manitoba Conference on Numerical Mathematics and Computing, vol. 30, pp. 45–87 (1981)
7. Tozawa, T., Jones, J., Swamy, S., Jiang, S., Adams, D., Shakespeare, S., Clowes, R., Bradshaw, D., Hasell, T., Chong, S., Tang, C., Thompson, S., Parker, J., Trewin, A., Bacsa, J., Slawin, A., Steiner, A., Cooper, A.: Porous organic cages. Nat. Mater. **8**, 973–978 (2009)
8. Tutte, W.: A census of planar maps. Canad. J. Math. **15**, 249–271 (1963)