Constraint Analysis for Security Policy Partitioning Over Tactical Service Oriented Architectures

Vasileios Gkioulos and Stephen D. Wolthusen

Abstract Tactical networks are typically of an ad-hoc nature operating in highly restricted environments and constrained resources. The frequent presence of communication disruptions and network partitioning must also be expected and managed, while core functionalities must be maintained, providing asynchronous invocation and access to services in a distributed manner. Supporting the required functionalities of the contemporary tactical environment, requires the dynamic evaluation of security policies, incorporating semantic knowledge from various network layers, together with facts and rules that are defined axiomatically a priori. However, the required basis for such policy decisions can be excessively extended and dynamic. Thus, it is desirable to locally minimize the scope of the policy maximizing efficiency. In this paper, we therefore analyze criteria and optimization goals for the a priori distribution and partitioning of security policies, ensuring the continuous support of the required capabilities, given the operational tasks of each deployed actor.

Keywords Ad Hoc network · Distribution · Security · Security policies · Tactical network · Partitioning

1 Introduction

Tactical networks refer to mobile networks, with characteristics similar to Ad-Hoc and mesh structures. They are typically adjusted and deployed to serve the specifics of a particular operation, with characteristics known partially in advance.

V. Gkioulos (✉) · S.D. Wolthusen
Norwegian Information Security Laboratory, Norwegian University of Science and Technology, Trondheim, Norway
e-mail: vasileios.gkioulos@ntnu.no

S.D. Wolthusen
e-mail: stephen.wolthusen@ntnu.no

S.D. Wolthusen
School of Mathematics and Information Security, Royal Holloway, University of London, London, UK

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_9

Consequently, the study, evaluation and realization of globally suitable security mechanisms, must be able to dynamically adapt to the versatile and diverse nature of tactical operations. The tactical environment is continuously studied, both in terms of operational analysis and technical evaluation [1–5], allowing the extraction of valuable information regarding their nature, characteristics and requirements.

The deployed assets for a specific operation should be expected to operate over distinct platforms, with diverse capabilities and requirements, including the ability to operate in coalition environments. Additionally, due to resource limitations and the dynamically evolving topologies, no safe assumptions can be made regarding continuous connectivity, since a tactical network may degrade to the point of partitioning. For the same reasons, communication failures, uncertain service delivery and extensive delays must be expected and properly addressed. Within this environment tactical networks must be able to provide reliable and secure service delivery and communication. Hence, the realized security mechanisms have to be distributed across the deployed assets, since no centralized security dedicated entity can be assumed, due to inability of reassuring a continuously available link towards it.

In addition to the aforementioned constraints, the introduction and increasing requirement of supporting Network Enabled Operations (NEO) and Network Centric Warfare (NCW), formulated a new set of requisite features regarding the functionalities of contemporary tactical networks [6–8]. Thus, mechanisms based on the Service Oriented Architecture (SOA) paradigm emerged as the most suitable mediators for the realization of these requirements, within the deployed C4I (Command, Control, Communication, Computers and Intelligence) and C4ISR (Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance) systems [9–15].

Securing tactical SOA requires not only the accomplishment of general information protection goals (such as confidentiality, availability, authenticity and control) but also the dynamic protection of communication, data at rest and processing, within the aforementioned restrictions imposed by their nature. The realization of suitable security mechanisms requires the conceptualization of the multitudinous semantic attributes available across the network. Such elements rise among others from services, terminals, information, communication links and subjects, alongside their relations and interactions.

Well known mechanisms (Such as WS-Security, Ponder [16], SAML [17], XACML [18], RT [19], cassandra [20], Peer-Trust [21], Tulip [22], ROWLBAC [23], REI [24], KAOS [25], Kolter et al. [26]) have been extensively studied and found to be unsuitable for the contemporary tactical environment for a variety of reasons. Some face limitations in capturing and expressing the required semantics, others are relatively heavyweight regarding their computational and communication requirements, or lack the ability of decentralized operation. Furthermore, some are not rigorous and flexible enough in expressing and reasoning over security policies, face scalability limitations or a combination of these reasons. These studies (Including but not limited to [23, 27–34]) promoted the use of ontologies for the definition of general purpose security policies, due to their expressive power and ability to overcome the aforementioned constraints.

For the same reasons in our previous study [35] we proposed a framework for the realization of an ontologically defined security infrastructure, with the use of Web Ontology Language (OWL), suitably adjusted to the constraints and high level functional requirements of tactical SOA. Yet, although ontologies can provide the required extended scope over the existing semantic attributes, the aforementioned inability to rely on a centralized security dedicated entity requires the distribution of the defined mechanisms across the deployed tactical nodes. However, due the functional limitations of tactical nodes (e.g. computational capacity, storage capacity, bandwidth availability), mere replication of those mechanisms across the network is inefficient and commonly infeasible.

In this paper we present our findings regarding the partitioning and distribution of ontologically defined security policies, suitably adjusted to the specifics of tactical SOA, aiming to maximize efficiency by minimizing the local scope of the policy. We approach this topic by identifying the criteria rising from the nature of tactical SOA, seeking a reliable limitation to a problem similar in nature to a 0-1 multiple knapsack problem, therefore subject to existing mechanisms of discrete optimization. Furthermore, we identify suitable elements in order to minimize the complexity by reducing the number of instances, maintaining the complete set of functionalities supported by the defined security policies.

2 Ontologically Defined Security Policies for Tactical SOA

An ontologically defined security policy dedicated to the specifics of tactical SOA must be able to provide the dynamic protection of communication, data at rest and processing, alongside the general information protection goals. Such a mechanism requires the conceptualization of the assorted semantic attributes, within a robust yet flexible mapping between the involved elements. These elements comprise of the defined *Domains* (Including but not limited to planning, protection, diligence, detection and response), the required *Capabilities* (Similar to NATO Architecture Framework/NATO Capability View (NAF/NCV) [36], including but not limited to core, application, communication and inter-domain), the available *Actions* and a set of governing *Rules* for each action, each of which incorporates a varying set of the involved *Conditions* (Which correspond to the aforementioned dynamic and static semantics). An outline of the security policy structure, including the overlaying relations, is presented at Fig. 1.

These elements are defined as OWL classes, which are populated according to the requirements of each tactical operation. The Security_Core is the anchor of the policy structure similar to owl: Thing of ontologies, incorporating all the other elements as subclasses. Furthermore, the Security_Core is the gateway towards the TSI_common (Tactical Service Infrastructure common core ontologies) and additional ontologies that are required to be linked with the security infrastructure. Thus, through the Security_Core the security policy can monitor the functionality of the enabled capabilities, within each tactical domain. This is achieved by the on-line

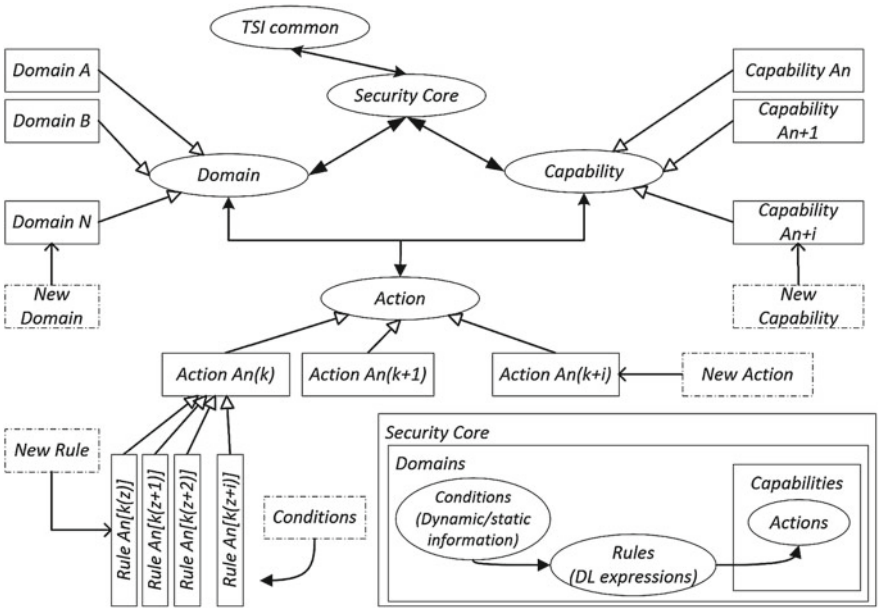

Fig. 1 Outline of security policy structure

evaluation of the environmental conditions, through the set of governing rules established for each action.

This framework permits the multi-domain and cross-layer implementation of security policies. Making use of the expressive power of description logic, complex relations can be established between the defined elements. Thus, actions within a specific capability can be linked to trigger the conditions evaluation of a rule established over a different domain. Additionally, conditions collected from various layers can affect decisions on other layers. Namely, a condition within the physical layer can affect a decision regarding the application layer.

The conceptualization of the policy framework is achieved by the use of unary and binary predicates, which are utilised to define the various network entities (data, services, users, terminals) and the relationships among them. Thus, a complete representation of the network can be achieved by defining the distinct constituting elements and their relations, as part of the tactical terminology. The tactical terminology is constructed within the T-Box with unique and acyclic concept definition, while the A-Box is used for instance identification with the use of concept and role assertions.

A detailed procedure for the ontological definition of security policies dedicated to tactical SOA was described earlier [35].

Table 1 Governing parameters for the distribution of security policies

Security policy distribution		
Ontology	Tactical nodes	Dynamism
1-Syntactic complexity	3-Operational specialization	6-Dynamic attributes
2-Structural complexity	4-Functional specialization	7-Dynamic policy evaluation
	5-Operating features	8-Tactical decision cycle

3 Constraint Analysis for the Distribution of Security Policies

Limiting the local scope of the security mechanisms in each tactical node, requires the identification of the parameters enabling the partitioning and distribution of security policies, within the context of tactical SOA. In the following sections, we present our findings regarding the identified parameters of critical impact, as they are presented in Table 1.

Our study over the functional characteristics of tactical SOA and the operation of ontologically defined security policies, promoted three main categories of governing parameters, regarding the attainment of the required horizontal and vertical security policy distribution. The first category refers to the evaluation of the policy, constructed based on the framework described in Fig. 1, regarding its overall and local complexity. The second category refers to the evaluation and categorization of the deployed tactical nodes, based on their expected functional and operational specialization, alongside their presumably known operating features. The last category refers to the sufficient integration of dynamism, emerging from the aforementioned characteristics of the tactical environment.

3.1 Complexity Inducing Components of Tactical Ontological Constructs

As highlighted earlier, the definition of the ontological security policy is unique for each tactical operation, constructed over an overlaying common framework (Fig. 1).

Regarding the syntactic complexity, OWL is provided in three increasingly expressive subsets that can be used for the definition of suitable security policies, namely OWL-Lite (Exp-time complete complexity), OWL-DL (NExp-time complete complexity) and OWL-Full (Undecidability). OWL-Lite supports simple constraint features and basic classification hierarchies. OWL-DL supports increased expressiveness, maintaining guaranteed computational completeness. Finally, OWL-Full provides maximum expressiveness and syntactic capabilities similar to RDF, yet reasoning is not reassured. A summary of the available constructs within OWL-Lite and OWL-DL is presented in Table 2 [37–39]. Furthermore, OWL 2 provides a

Table 2 Summary of available constructs within OWL-Lite and OWL-DL

OWL-Lite	
Category	Constructs
Constructors	Class, subClassOf, Property, subPropertyOf, domain, Individual
Restrictions	Restriction, allValuesFrom, someValuesFrom, intersectionOf
Equality	EquivalentClass, equivalentProperty, sameAs, differentFrom
Cardinality (0 or 1)	MinCardinality, maxCardinality
Properties	ObjectProperty, inverseOf, Datatype, Transitive, Symmetric, Functional, InverseFunctional
OWL-DL (In addition to the aforementioned)	
Values	HasValue
Cardinality (No limitation)	MinCardinality, maxCardinality
Class axioms	DisjointWith, equivalentClass, complementOf, subClassOf, unionOf, intersectionOf

wide set of subset profiles, supporting assorted accommodation between expressive power and reasoning efficiency. For instance, OWL 2 QL (NLogSpace complete complexity) is dedicated to efficiently supporting extensive instance data and database queries, OWL 2 RL (NP-time complete complexity) is optimized for scalable reasoning without fully utilizing the available expressive power, while OWL 2 EL (P-Time complete complexity) is suitable for large scale definition of properties and classes.

Regarding the structural complexity of the defined security policy, a variety of metrics with significant impact have been identified through our study. Their additive complexity overhead must be contemplated during the initial construction of the security policy, while they can be classified as:

1. **Vocabulary size**: The amount of the defined classes, individuals and properties.
2. **Impurity**: The deviation of the ontological structure from a pure tree form, as a result of the defined rdfs: subClassOf axioms.
3. **Mean inheritance**: The mean overall distance between the defined ancestor classes to the corresponding root classes.
4. **Connectivity**: A measurement of the connection density within the security policy, defined as the average number of connections for each of the defined elements (Classes and individuals).

Additionally, estimating the significance of individual classes over the overall functionality of the security policy, is pivotal for the identification of crucial distribution links within the policy structure. Such an estimation is possible with the use of the following metrics, for each of the defined classes.

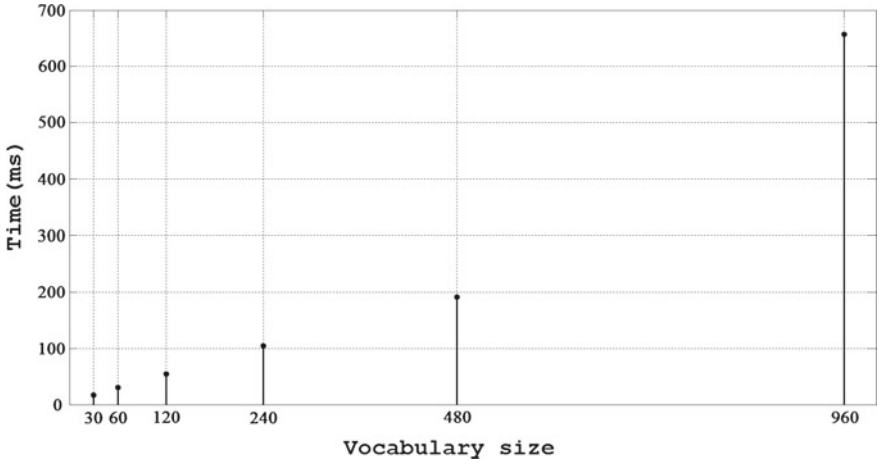

Fig. 2 Reasoning time escalation in relation to vocabulary size

1. **Direct inheritance**: The number of direct ancestors for each defined class. Meaning the number of subclasses defined based on a specific class and affected by changes within it.
2. **Inheritance exponentiation**: The depth of the most distant ancestor of a given class. It can be used as a measure of information inheritance within classes that belong to the same policy branch.
3. **Individual connectivity**: A connection density measure, referring to a specific class, calculated as the sum of the defined relations from and towards this class.

A representation of how these parameters affect the complexity of the security policy and the time required for reasoning over it, is provided in Fig. 2. In this set from our executed simulations, the Pellet reasoner is used over a basic ontological construct, structured using the ALC(D) fragment, in order to isolate and measure the impact of the value of the Vocabulary_size parameter. Furthermore, Fig. 3 provides an illustration of the global complexity estimation, based on the aforementioned combination of the propagating syntactic and local structural complexities.

3.2 Classification and Management of Tactical Nodes

Tactical nodes refer to a plethora of mobile platforms, with restricted operational characteristics and distinct requirements. Achieving a viable security policy distribution, requires the identification and incorporation of their influential attributes, for which we can attain a priori awareness. Our study over the characteristics of tactical nodes and the nature of tactical operations promoted three elements, of significant impact, as presented in Table 1.

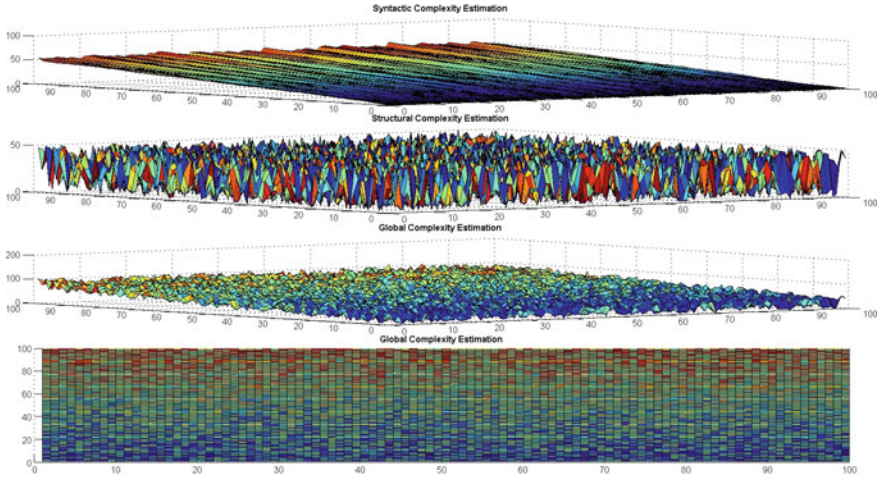

Fig. 3 Complexity estimation of tactical ontological constructs

The first two elements represent the operational and functional specialization of tactical nodes, rising through the initial operational and contingency planning of a tactical operation. The operational specialization refers to the identification of distinct operational groups among the entirety of the deployed assets, based on their particular strategic objectives. Additionally, functional node specialization, occurs due to the distinct roles of each node within the initial categorization into operational groups (e.g. Assuming a tactical team, the hand-held device of a medic, has distinct service/security requirements from the hand-held device of the team leader or a rifleman).

Hence, the defined operational and functional node specializations can provide an initial classification of nodes, in discrete groups with distinct yet entangled security requirements. This classification can form the basis for the horizontal (In terms of Domain/Capability groups) or vertical (In terms of Action/Rule groups), distribution of security policies, incorporating the operational perspective. A representation of the aforementioned procedure is presented in Fig. 4, based on our executed simulations. In this scenario, ten tactical nodes are organised in two operational groups (OG1-square, OG2-circle), while three functional groups (FG1-green(—), FG2-red(|), FG3-blue(\)) are globally defined.

An additional element that can significantly affect the distribution of security policies, within tactical SOA, is the presumably known operating features of tactical nodes. Tactical nodes refer to a variety of platforms, which may differ in various terms affecting their performance (Grouped afterwards as Computational Capacity). These elements can be classified as:

1. Computational power
2. Environmental limitations
3. Physical limitations
4. Resolution/accuracy limitations

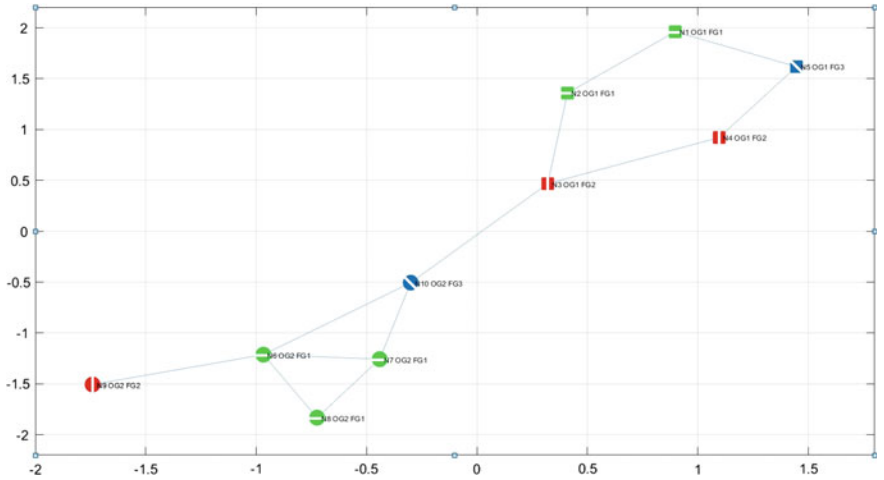

Fig. 4 Node classification based on operational and functional specialization

5. Input/output limitations
6. Range/coverage limitations
7. Network interconnection limitations

The knowledge of these parameters and their incorporation within the policy distribution decisions, can be used to enhance the network performance, in terms that include communication latency, service delivery/discovery and autonomy in case of partitioning, since they are correlated with the elements presented at Sect. 3.1.

3.3 Incorporation of Dynamism

The aforementioned characteristics of the tactical ecosystem, describe a highly dynamic and continuously evolving environment. Thus, the notion of dynamism has to be embodied, not only within the definition of the security policy, but also through the distribution mechanisms. For this reason, the realised security components must incorporate the available dynamic attributes across the network elements/domains, but also allow for the dynamic security policy evaluation, as presented at Sect. 2.

For the purpose of this study, achieving the efficient security policy distribution, also relies on the incorporation of a suitable tactical decision cycle. John Boyd's OODA (Observe, Orient, Decide, Act), is a decision cycle developed and used by military strategists, primarily within the strategic domain and the first two stages (Preparation, Execution) of combat operations, with additional applications to the third stage (Debrief/Evaluation). Evaluating the various suggested iterations of the OODA loop [40], the NCW targeted OODA model, proposed by Smith [41], emerged as the most suitable solution for tactical SOA, despite its complexity. Our decision

was promoted by the fact that this model can coincide with suitably adjusted ontologically structured security policies, into the representation of complex and dynamic systems, providing in addition an enhanced level of granularity.

Similarly to the implementations within the strategic domain, the distinction between the involved processes (Observe, Orient, Decide, Act) and further segmentation to the defined domains (Physical, Information and Cognitive in Smith's model), can be eminently beneficial towards the technical implementation of a suitable distribution mechanism, within the tactical domain. Thus, the execution of the distinct processes of the decision cycle, can be delegated and distributed within the nodes of each operational group, allowing them to cooperatively reach the attainment of each objective, while dispensing the computational and overall cost. Additionally, the distribution of the involved processes, dispenses the required resources and time for the achievement of the optimality point, within the *Time Cost of Information* and *Decision Confidence/Quality* function, as described by Harrison [42].

4 Accommodation of the Defined Constraints for Security Policy Distribution

Having defined the overall security architecture and the critical parameters, for the distribution of security policies over tactical SOA, it is necessary to reconstruct the framework presented in Fig. 1, in accordance to the aforementioned criteria. This will allow the required minimization of the local policy scope in each tactical node, maintaining all the requisite functionalities. Additionally, this procedure will provide a transformation into a problem similar in nature to a 0-1 multiple knapsack problem, therefore subject to existing and widely studied optimization mechanisms. Furthermore, the incorporation of the identified elements, prior to the implementation of these mechanisms, will significantly increase the computational efficiency, due to the induced minimization of the number of instances.

Aiming to continuously support the required functionalities, within the defined security mechanisms:

1. Capabilities may span across various domains.
2. Actions may span across various capabilities.
3. A specific action within the context of different capabilities or domains, may be governed by a distinct set of rules.

Thus, a three dimensional space is required, in order to represent all the possible combinations of domains, capabilities and actions. The multitude of these ordered triplets constitutes the overall security policy of the tactical network, as presented in Fig. 5, while every individual action can be represented by a vector:

$$Action : A'_m = (D\hat{i} + C\hat{j} + A\hat{k}),\ where\ \hat{i}, \hat{j}, \hat{k}\ are\ unit\ vectors. \tag{1}$$

as presented in Fig. 6

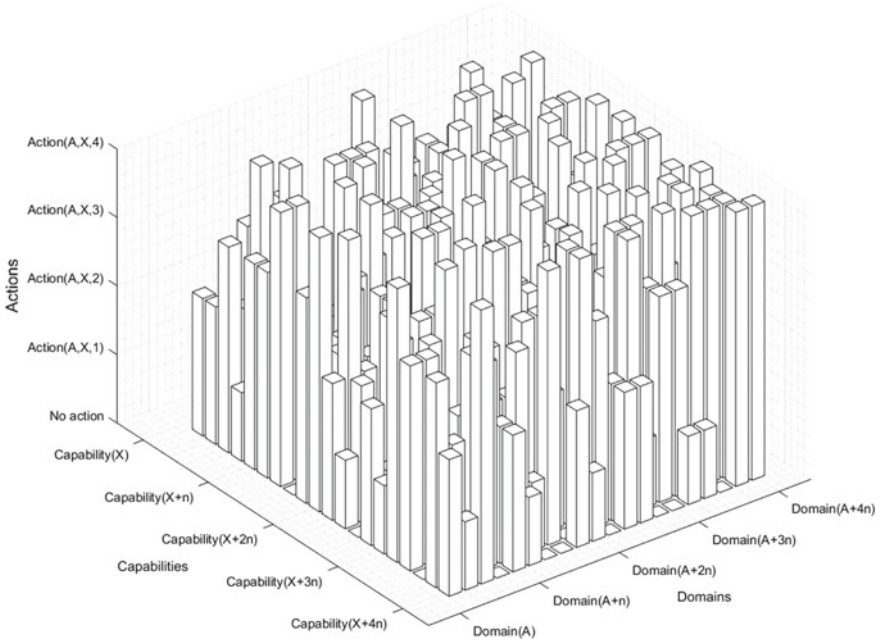

Fig. 5 Visualisation of a simplified security policy

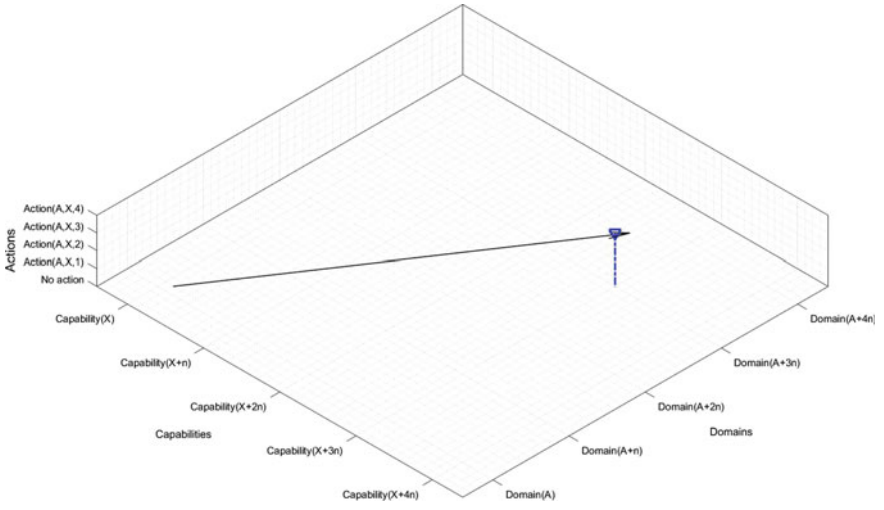

Fig. 6 Visualisation of a distinct action within the security policy

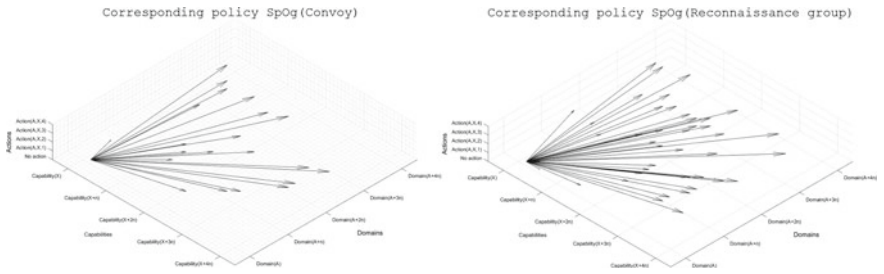

Fig. 7 Specimen security policy vector sets for convoy and reconnaissance operational groups

Due to the aforementioned constraints, mere replication of the entire security policy across all the deployed nodes is not sufficient. The incorporation of node operational specialization (Third identified element—Table 1), can provide an initial filtering, towards the minimization of the distributed policy branches. Thus, the specific operational contexts of the various deployed groups of nodes, correspond to a distinct set of basic vectors (Linearly independent), in the form:

$$Security\ policy\ :\ SpOg_{(x)} = \{A'_m, A'_{m+1},\ \ldots, A'_{m+n}\} \tag{2}$$

This mapping is based on the required/estimated actions of each operational group, within each tactical operation, while it can be constructed a priori and automatically recalled when needed. For instance, a convoy operation may incorporate various operational groups including but not limited to the convoy, multiple protection groups and a medical evacuation group. The structure of the corresponding security policies, for each operational group, has a form similar to those presented in Fig. 7.

Yet, policy replication within an operational group is not the optimal solution, due to the node functional specialization (Fourth identified element—Table 1). Thus, the distinction between the functional groups of nodes across each given operational group, allows for further partitioning of the security policy as:

$$SpOg_{(x)} = SpFg_{(y)} \cup SpFg_{(y+1)} \cup \ldots \cup SpFg_{(y+n)} \tag{3}$$

Hence, the security policy of a given operational group is defined as the union of the security policies of the functional groups that constitute it. This allows for the defined subsets ($SpFg_{(y)}$), to collectively compose or address distinct dimensions of the given $SpOg_{(x)}$. Yet, a given vector (Action: $A'_m = (D\hat{i} + C\hat{j} + A\hat{k})$) can span various subsets ($SpFg_{(y)}$) or be unique to one of them. A calculation of the sets intersections (e.g. $SpFg_{(y)} \cap SpFg_{(y+1)}$) and the sets differences (e.g. $SpFg_{(y)}/SpFg_{(y+1)}$), can provide a direct mapping between each action vector and the functional groups, across which it can be distributed, as:

$$\begin{array}{lll} SpFg_{(y)} = \{A'_1, A'_2, A'_3\} & & A'_1 : Fg_{(y)}, Fg_{(y+1)} \\ SpFg_{(y+1)} = \{A'_1, A'_3\} & \succ & A'_2 : Fg_{(y)}, Fg_{(y+2)} \\ SpFg_{(y+2)} = \{A'_2, A'_3, A'_4\} & & A'_3 : Fg_{(y)}, Fg_{(y+1)}, Fg_{(y+2)} \\ & & A'_4 : Fg_{(y+2)} \end{array}$$

As presented in the defined security policy framework (Fig. 1), each vector $A'_m = (D\hat{i} + C\hat{j} + A\hat{k})$ corresponds to a set of governing rules, distinct for each individual action, enabling the dynamic adaptation of the security policy to alterations of the environmental conditions:

$$A'_m = \{R_{(z)}, R_{(z+1)}, \ \ldots, R_{(z+n)}\} \tag{4}$$

Each rule is constructed making use of the expressive power of description logic, in order to incorporate the available static and dynamic attributes (Sixth identified element—Table 1) across the network, into the defined security policy decisions. Furthermore, as presented at Sect. 3.1, each rule caries an inherited complexity based on the values of the presented metrics, as a function of its syntactic and structural complexities (First and second identified elements—Table 1). Thus:

$$Vector\ complexity\ :\ CA'_m = \sum_{z=1}^{n} CR_{(z)} \tag{5}$$

Consequently, based on the operational features of the tactical nodes constituting each functional group (Fifth identified element—Table 1), suitable metrics incorporating their computational capacity (e.g. $CCFg_{(y)}$) can be constructed. Hence, given the aforementioned scenario, it is possible to construct a corresponding set of equations among the defined CA'_m and $CCFg_{(y)}$, as:

$$\begin{aligned} CA'_1 &= a * CCFg_{(y)} + b * CCFg_{(y+1)} \\ CA'_2 &= c * CCFg_{(y)} + d * CCFg_{(y+2)} \\ CA'_3 &= e * CCFg_{(y)} + f * CCFg_{(y+1)} + g * CCFg_{(y+2)} \\ CA'_4 &= h * CCFg_{(y+2)} \\ a + c + e &= 1 \\ b + f &= 1 \\ d + g + h &= 1 \end{aligned} \tag{6}$$

If the evaluation of the occurring equations is not feasible or a simplification of the process is required, assumptions can be made regarding the values of the variables, given the incorporation of the two additional identified elements of our study, namely:

1. Dynamic policy evaluation (Seventh identified element—Table 1): Meaning that the most suitable of the *available* rules, is dynamically selected to govern an action.
2. Decision cycle (Eighth identified element—Table 1): Meaning that (i) Gathering/storing the required rule inputs, (ii) Selecting the most suitable rule, (iii) Evaluating the selected rule, (iv) Enforcing the rule outcome, can be further distributed among the nodes constituting each functional group.

Thus, allowing for some additional flexibility regarding the exact values.

The utilization of the identified elements, as presented in this section, significantly limits the scale of the security policy distribution requirement, by identifying the maximum set of nodes responsible for a given set of actions (Equivalently: Minimizing the set of actions each node is responsible for). Having introduced the notions of CA'_m and $CCFg_{(y)}$, this has been limited to a problem similar in nature to a 0–1 knapsack problem in the following form.

Given for an action vector $A'_m = \{R_{(1)}, R_{(2)}, ..., R_{(n)}\}$ a finite set of rules, defined so $CR_{(1)} \leq CR_{(2)} \leq, ..., \leq CR_{(n)}$, and $SpFg_{(y)} = \{SpFg_{(1)}, SpFg_{(2)}, ..., SpFg_{(k)}\}$ a finite set of functional groups of tactical nodes with fixed capacities $CCFg_{(y)} = \{CCFg_{(1)}, CCFg_{(2)}, ..., CCFg_{(k)}\}$ (Calculated earlier as a percentage of their overall CC, dedicated to this action) and fixed 'k'. Assign each element of A'_m across the elements of $SpFg_{(y)}$ so:

1. The capacity of no element of $SpFg_{(y)}$ is exceeded.
2. No element of A'_m is duplicated within any given element of $SpFg_{(y)}$.
3. Duplicates of the elements of A'_m with minimum complexity, are allowed across the elements of $SpFg_{(y)}$, to increase redundancy.

Thus, given that:

1. $pR_{(j)}$ = Profit form $R_{(j)}$ (Requirement for a specific subset of rules).
2. $CR_{(j)}$ = Complexity of $R_{(j)}$.
3. $CCFg_{(i)}$ = The calculated percentage of each CC dedicated to this action.

Then maximize:

$$D = \sum_{i=1}^{k} \sum_{j=1}^{n} pR_{(j)} * X_{ij} \tag{7}$$

Subject to:

$$\sum_{j=1}^{n} CR_{(j)} * X_{ij} \leq CCFg_{(i)}, \quad i = [1, ..., k] \tag{8}$$

$$\sum_{j=1}^{n} X_{ij} = 1, \quad i = [1, ..., k] \tag{9}$$

$$X_{ij} = 1 \; or \; 0, \quad i = [1, ..., k], j = [1, ..., n] \tag{10}$$

where:

$$X_{ij} = \begin{cases} 1 \text{ if } R_{(j)} \text{ is selected for } Fg_{(i)}, \\ 0 \text{ if not} \end{cases}$$

A variety of exact and heuristic algorithms has been developed for the attainment of optimal/near optimal solutions for this type of problems [43–54]. The average solution time of these algorithms is directly correlated to the number of instances [55, 56], which with the incorporation of the defined parameters, has been limited to a minimum set of rules for each node, maintaining at the same time support of all the required functionalities within a tactical operation.

It must also be stated that the described procedure is executed at the mission preparation stage, facing no computational, time, communication or other type of limitations. In this manner, we can achieve a mapping between the required and the available computational power achieving optimal policy partitioning and distribution, incorporating all the corresponding elements of significant impact.

5 Conclusions

Through this article, the findings of our study regarding the parameters governing the partitioning and distribution of security policies within tactical SOA, have been presented. Evaluating the characteristics of tactical networks and utilized actors, the involved elements of critical impact, have been identified and analysed. Furthermore, a suitable mechanism has been suggested, accommodating the identified parameters, for the optimum partitioning and distribution of security policies within the mission preparation stage.

Our future plans include the further refinement and evaluation of the proposed mechanism for the mission preparation stage and its extension within the mission execution stage, in the presence of additional constraints, such as connectivity and bandwidth availability. More precisely the utilisation of hierarchical structures within the defined rule sets, governing the individual actions, and the constrained optimization for online distribution of both security policies and governing conditions. Furthermore, we intent to identify suitable mechanisms for the reconciliation of security policies, adjusted to the dynamics of tactical SOA.

Acknowledgements The results described in this work were obtained as part of the European Defence Agency project TACTICS (Tactical Service Oriented Architecture). The TACTICS project is jointly undertaken by Patria (FI), Thales Communications & Security (FR), Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie FKIE (DE), Thales Deutschland (DE), Finmeccanica (IT), Thales Italia (IT), Norwegian University of Science and Technology (NO), ITTI (PL), Military Communication Institute (PL), and their partners, supported by the respective national Ministries of Defence under EDA Contract No. B 0980.

References

1. Horne, G., Leonardi, M.: Maneuver warfare science 2001 (2001)
2. Bar-Noy, A., Cirincione, G. Govindan, R. Krishnamurthy, S., LaPorta, T., Mohapatra, P., Neely, M., Yener, A.: Quality-of-information aware networking for tactical military networks. In: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 2–7, Mar 2011
3. Burbank, J., Chimento, P., Haberman, B., Kasch, W.: Key challenges of military tactical networking and the elusive promise of manet technology. IEEE Commun. Mag. **44**, 39–45 (2006)
4. Elmasry, G.: A comparative review of commercial versus tactical wireless networks. IEEE Commun. Mag. **48**, 54–59 (2010)
5. Shi, V.: Evaluating the performability of tactical communications networks. IEEE Trans. Veh. Technol. **53**, 253–260 (2004)
6. Moffat, J.: Adapting Modeling & Simulation for Network Enabled Operations (2011)
7. Alberts, D.S., Hayes, R.E.: Power to the Edge (2003)
8. Smith, E.A.: Complexity, Networking, and Effects-Based Approaches to Operations (2006)
9. Lund, K., Eggen, A., Hadzic, D., Hafsoe, T., Johnsen, F.: Using web services to realize service oriented architecture in military communication networks. IEEE Commun. Mag. **45**, 47–53 (2007)
10. Johnsen, F., Bloebaum, T., Schenkels, L., Fiske, R., Van Selm, M., de Sortis, V., van der Zanden, A., Sliwa, J., Caban, P.: Soa over disadvantaged grids experiment and demonstrator. In: Communications and Information Systems Conference (MCC), 2012 Military, pp. 1–8, Oct 2012
11. Suri, N.: Dynamic service-oriented architectures for tactical edge networks. In: Proceedings of the 4th Workshop on Emerging Web Services Technology, WEWST '09, pp. 3–10. ACM, New York, NY, USA (2009)
12. IST-090 Task Group, Service oriented architecture (SOA) challenges for real time and disadvantaged grid (IST-090). https://www.cso.nato.int/Activity_Meta.asp?ACT=1830, Apr 2014
13. IST-118 Task Group, SOA recommendations for disadvantaged grids in the tactical domain (IST-118). https://www.cso.nato.int/ACTIVITY_META.asp?ACT=2293
14. Maule, R., Lewis, W.: Security for distributed soa at the tactical edge. In: Military Communications Conference, 2010—MILCOM 2010, pp. 13–18, Oct 2010
15. Mayott, G., Self, M., Miller, G.J., McDonnell, J.S.: Soa approach to battle command: simulation interoperability (2010)
16. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The ponder policy specification language. In: Sloman, M., Lupu, E., Lobo, J. (eds.) Policies for Distributed Systems and Networks, Lecture Notes in Computer Science, vol. 1995, pp. 18–38. Springer, Berlin (2001)
17. OASIS, Oasis security services (saml) tc
18. Ramli, C.D.P.K., Nielson, H.R., Nielson, F.: The logic of XACML. Sci. Comput. Prog. **83**, 80–105 (2014)
19. Li, N., Mitchell, J., Winsborough, W.: Design of a role-based trust-management framework. In: Proceedings. 2002 IEEE Symposium on Security and Privacy, 2002, pp. 114–130 (2002)
20. Becker, M., Sewell, P.: Cassandra: distributed access control policies with tunable expressiveness. In: Proceedings. Fifth IEEE International Workshop on Policies for Distributed Systems and Networks, 2004. POLICY 2004, pp. 159–168, June 2004
21. Nejdl, W., Olmedilla, D., Winslett, M.: Peertrust: Automated trust negotiation for peers on the semantic web. In: Jonker, W., Petkovi, M. (eds.) Secure Data Management, Lecture Notes in Computer Science, vol. 3178 pp. 118–132. Springer, Berlin (2004)
22. Czenko, M., Doumen, J., Etalle, S.: Trust management in p2p systems using standard tulip. In: Karabulut, Y., Mitchell, J., Herrmann, P., Jensen, C. (eds.) Trust Management II, FIP The International Federation for Information Processing, vol. 263 pp. 1–16, Springer, US (2008)
23. Finin, T., Joshi, A., Kagal, L., Niu, J., Sandhu, R., Winsborough, W.H., Thuraisingham, B.: ROWLBAC—representing role based access control in OWL. In: Proceedings of the 13th

Symposium on Access control Models and Technologies, ACM Press, Estes Park, Colorado, USA, June 2008
24. Kagal, L., Finin, T., Paolucci, M., Srinivasan, N., Sycara, K., Denker, G.: Authorization and privacy for semantic web services. IEEE Intell. Syst. **19**, 50–56 (2004)
25. Uszok, A., Bradshaw, J.M., Johnson, M., Jeffers, R., Tate, A., Dalton, J., Aitken, S.: Kaos policy management for semantic web services. IEEE Intell. Syst. **19**, 32–41 (2004)
26. Kolter, J., Schillinger, R., Pernul, G.: Building a distributed semantic-aware security architecture. In: Venter, H., Eloff, M., Labuschagne, L., Eloff, J., von Solms, R. (eds.) New Approaches for Security, Privacy and Trust in Complex Environments, IFIP International Federation for Information Processing, vol. 232 pp. 397–408. Springer, US (2007)
27. Ferrini, R., Bertino, E.: Supporting RBAC with XACML + OWL. In: Carminati, B., Joshi, J. (eds.) Proceedings of the 14th ACM Symposium on Access Control Models and Technologies (SACMAT '09), pp. 145–154. ACM Press, Stresa, Italy, June 2009
28. Ben Brahim, M., Chaari, T., Ben Jemaa, M., Jmaiel, M.: Semantic matching of ws-securitypolicy assertions. In: Pallis, G., Jmaiel, M., Charfi, A., Graupner, S., Karabulut, Y., Guinea, S., Rosenberg, F., Sheng, Q., Pautasso, C., Ben Mokhtar, S. (eds.) Service-Oriented Computing-ICSOC 2011 Workshops, Lecture Notes in Computer Science, vol. 7221, pp. 114–130. Springer, Berlin (2012)
29. Helil, N., Rahman, K.: Extending xacml profile for rbac with semantic concepts. In: 2010 International Conference on Computer Application and System Modeling (ICCASM), vol. 10, pp. V10–69–V10–74, Oct 2010
30. Blanco, C., Lasheras, J., Valencia-Garcia, R., Fernandez-Medina, E., Toval, A., Piattini, M.: A systematic review and comparison of security ontologies. In: 3rd International Conference on Availability, Reliability and Security, 2008. ARES 08, pp. 813–820, Mar 2008
31. Souag, A., Salinesi, C., Comyn-Wattiau, I.: Ontologies for security requirements: a literature survey and classification. In: Bajec, M., Eder, J. (eds.) Advanced Information Systems Engineering Workshops, Lecture Notes in Business Information Processing, vol. 112 pp. 61–69. Springer, Berlin (2012)
32. Nguyen, V.: Ontologies and information systems: a literature survey 6 (2011)
33. Kolovski, V., Parsia, B., Katz, Y., Hendler, J.: Representing web service policies in owl-dl. In: International Semantic Web Conference (ISWC), pp. 6–10 (2005)
34. Trivellato, D., Zannone, N., Glaundrup, M., Skowronek, J., Etalle, P.S.: A semantic security framework for systems of systems. Int. J. Coop. Inf. Syst. **22**, 1–35 (2013)
35. Gkioulos, V., Wolthusen, S.D.: Enabling dynamic security policy evaluation for service-oriented architectures in tactical networks. In: Accepted for presentation at Norwegian Information Security Conference 2015 (NISK-2015) (2015)
36. Nato Architecture Framework, NATO Capability View, NAF v3 NCV-2, June 2013
37. W3C Recommendation, OWL2-OVERVIEW OWL 2 Web Ontology Language Document Overview, 2nd Edn. http://www.w3.org/TR/2012/REC-owl2-overview-20121211/. Dec 2012
38. Schneider, P.F.P., Hayes, P., Horrocks, I.: OWL-SEMANTICS OWL Web Ontology Language Semantics and Abstract Syntax. http://www.w3.org/TR/2004/REC-owl-semantics-20040210/. Feb 2004
39. Motik, B., Grau, B.C., Horrocks, I., Wu, Z., Fokoue, A.: OWL2-PROFILES OWL 2 Web Ontology Language Profiles, 2nd edn. http://www.w3.org/TR/2012/REC-owl2-profiles-20121211/. Dec 2012
40. Breton, R., Rousseau, R.: The future of c2 the c-ooda: a cognitive version of the ooda loop to represent c2 activities. Topic: C2 process modelling
41. Smith, E.A.: Effects Based Operations. Crisis, and War. Center for Advanced Concepts and Technology, Applying Network Centric Warfare in Peace (2002)
42. Harrison, F.: The Managerial Decision-Making Process-5th edn. South-Western College Pub (1998)
43. Veni, K.K., Balachandar, S.R.: Int. J. Math. Comput. Phys. Electr. Comput. Eng. **4**(7), 1044–1048 (2010)

44. Balas, E., Glover, F., Zionts, S.: An additive algorithm for solving linear programs with zero-one variables. Oper. Res. **13**(4), 517–549 (1965)
45. Frville, A.: The multidimensional 01 knapsack problem: an overview. Eur. J. Oper. Res. **155**(1), 1–21 (2004)
46. Ohkura, K., Igarashi, T., Ueda, K., Okauchi, S., Matsunaga, H.: A genetic algorithm approach to large scale combinatorial optimization problems in the advertising industry. In: Proceedings. 2001 8th IEEE International Conference on Emerging Technologies and Factory Automation, 2001, vol. 2, pp. 351–357, Oct 2001
47. Chu, P., Beasley, J.: A genetic algorithm for the multidimensional knapsack problem. J Heuristics **4**(1), 63–86 (1998)
48. Balas, E., Martin, C.H.: Pivot and complement heuristic for 0–1 programming. Manag. Sci. **26**(1), 86–96 (1980)
49. Gavish, B., Pirkul, H.: Efficient algorithms for solving multiconstraint zero-one knapsack problems to optimality. Math. Program. **31**(1), 78–105 (1985)
50. Gilmore, P.C., Gomory, R.E.: The theory and computation of knapsack functions. Oper. Res. **14**(6), 1045–1074 (1966)
51. Osorio, M., Glover, F., Hammer, P.: Cutting and surrogate constraint analysis for improved multidimensional knapsack solutions. Ann. Oper. Res. **117**(1–4), 71–93 (2002)
52. Magazine, M., Oguz, O.: A heuristic algorithm for the multidimensional zero-one knapsack problem. Eur. J. Oper. Res. **16**(3), 319–326 (1984)
53. Volgenant, A., Zoon, J.A.: An improved heuristic for multidimensional 0–1 knapsack problems. J. Oper. Res. Soc. **41**(10), 963–970 (1990)
54. Weingartner, H.M., Ness, D.N.: Methods for the solution of the multidimensional 0/1 knapsack problem. Oper. Res. **15**(1), 83–103 (1967)
55. Caccetta, L., Kulanoot, A.: Computational aspects of hard knapsack problems. In: Proceedings of the Third World Congress of Nonlinear Analysts Nonlinear Analysis: Theory, Methods & Applications, vol. 47, no. 8, pp. 5547–5558 (2001)
56. Pisinger, D.: Where are the hard knapsack problems? Comput. Oper. Res. **32**(9), 2271–2284 (2005)