Decentralized Evolutionary Optimization Approach to the p-Median Problem

Stephan Otto[1] and Gabriella Kókai[2]

[1] method park Software AG, Am Wetterkreuz 19b, Erlangen, Germany
stephan.otto@methodpark.de
http://www.methodpark.de
[2] University Erlangen-Nuremberg, Martensstr. 3, Erlangen, Germany
kokai@informatik.uni-erlangen.de
http://www2.informatik.uni-erlangen.de

Abstract. The facility location problem also known as p-median problem concerns the positioning of facilities such as bus-stops, broadcasting stations or supply stations in general. The objective is to minimize the weighted distance between demand points (or customers) and facilities. In general there is a trend towards networked and distributed organizations and their systems, complicating the design, construction and maintenance of distributed facilities as information is scattered among participants while no global view exists. There is a need to investigate distributed approaches to the p-median problem. This paper contributes to research on location problems by proposing an agent oriented decentralized evolutionary computation (EC) approach that exploits the flow of money or energy in order to realize distributed optimization. Our approach uses local operators for reproduction like mutation, recombination and selection finally regulated by market mechanisms. This paper presents two general outcomes of our model: how adaptation occurs in the number and strategies of agents leading to an improvement at the system level. The novelty of this approach lies in the biology-inspired bottom-up adaptation method for inherent distributed problems. It is applied to the uncapacitated p-median problem but is also intended to be general for a wide variety of problems and domains, e.g. wireless sensor networks.

1 Introduction

Today's IT systems like the internet, global supply chains, sensor networks or grid applications are large distributed systems with a huge number of elements working in collaboration in order to fulfill requirements from customers, service providers, organizations and other systems. These systems cannot be fixed in their structure, design and behavior in order to cope with a highly dynamic and unpredictable environment. For this reason effective positioning of facilities is crucial. Current approaches are not easily adapted to for distributed systems which have no central coordinator. This is based on two reasons. At first elements

M. Giacobini et al. (Eds.): EvoWorkshops 2008, LNCS 4974, pp. 659–668, 2008.

are limited by how much they can communicate and process [1]. The second reason is information hiding, which means, that not all information can be given to a central control due to intellectual property or security reasons [2,3,4]. Throughout this paper the term decentralized system for systems without central control is used.

From a general point of view, we understand a decentralized system as "distributed solution" to a distributed problem. The overall solution (system strategy) is scattered amongst participating system elements, each with part of the solution. In order to adapt distributed systems to distributed problems, both an on-the-fly system size and strategy adjustment is needed. In [5,6] an approach is presented to show decentralized coordination in distributed economic agent systems. The investigation in [7] proposes evolutionary algorithms with on-the-fly population size adjustment as a rewarding way of parameter regulating. As in natural environments population sizes of species change and tend to stabilize around appropriate values according to some factors such as resources provided by the carrying capacity of the environment. Technically seen, population size is used as the most flexible parameter in natural systems. As a combination of these two approaches we present in this paper an artificial life like optimization strategy using a decentralized economic-agent based evolutionary algorithm that offers both properties needed. First results of a prototypic implementation using the examples of two p-median problems are shown. This paper seeks an answer to the question whether it is possible and rewarding to solve distributed problems using a decentralized evolutionary algorithm.

The remainder of this paper is organized as follows. In section 2, we review previous work related to p-median problem and evolutionary computation and evolutionary multi-agent systems. In section 3, we present our model of a market-based agent system. Section 4, illustrates our decentralized optimization method by combining evolutionary computation and multi-agent systems. Using this model two emergent effects are specified that occur within such a distributed system formed by agents: adaptation of the number of agents and adaptation of agent's strategies on the p-median problem. We show how the number of agents is regulated by the flow of money and how resource efficient strategies dominate. In section 5, we describe our current implementation and present experimental results. Finally, the conclusion and future plans are mentioned in section 6.

2 Related Work

Consider a set of locations $L = \{1 \dots l\}$ for customers and a set of potential locations for facilities $P = \{1 \dots p\} \subseteq L$ and $l \times l$ matrix D_{ij} of transportations costs for satisfying the demands w_i of the customers from the facilities. The weighted distance matrix is $W_{ij} = w_i D_{ij}$. The p-median problem is to locate the p facilities at locations of L in order to minimize the total transportation cost for satisfying the demand of the customers. Each customer is supplied from the cheapest open facility according to W. The uncapacitated p-median problem can be expressed as follows:

$$minimize \sum_{j=1}^{l} \sum_{i=1}^{p} w_{ij} \qquad (1)$$

The p-median problem has attracted a great deal of interest over the last decades in evolutionary research. Based on our search, Hosage and Goodchild [8] published the first research, followed by [9,10] and many others. However, they use the classical evolutionary approach with a central optimization loop and central operators. Classical generation-based evolutionary algorithms with a nonoverlapping population model where the entire population is replaced at each generation is not generally desirable in adaptive applications, where a high level of on-line performance is needed [11]. Also the central control of traditional evolutionary algorithms is inappropriate for distributed problems. A decentralized EC-enabled multi-agent-system (MAS) framework is presented in [12] where only local selection occurs. Based on [12] Smith and Eymann [6,5] investigate negotiation strategies in a supply chain for the production of cabinets but concentrate merely on self-organizing coordination effects. Explicit optimization is not investigated which is not suitable to the given problem.

To the best of our knowledge, there is no approach that applies a distributed evolutionary agent model to the p-median problem. The flow of money is used to direct evolutionary search. This paper describes a scalable multi-agent approach without central control providing distributed optimization.

3 Model

This section describes the model and generic outcomes used throughout the paper. In order to model and study generic decentralized systems, the term multi-agent system is used and defined as follows.

3.1 Multi-Agent System

Definition 1 (Multi-Agent System (MAS)). *A multi-agent system* $MAS = (A, E)$ *consists of a finite set of agents* $A = \{a_1, ..., a_z\}$ *embedded in an environment* E.

An agent is defined as follows:

Definition 2 (Agent). *An agent* $a = (I, O, m, s, c)$ *consists of a finite set of sensory inputs* $I = \{i_1, \ldots, i_r\}$, *a finite set of effector outputs* $O = \{o_1, \ldots, o_q\}$, *a function* $m : I \rightarrow O$ *which maps sensory inputs to effector outputs, the strategy vector* $s = (s_1, s_2, \ldots, s_r)$ *determining or parameterizing* m, *and the agents current funds* $c \in \mathbb{R}$.

As a basic element of a decentralized system, an agent a is an autonomous sensor effector entity with function m_a. The strategy vector s_a represents a set of parameters that are assigned to a. We consider the agent's function m_a to be determined by s_a and thus, the strategy s_a of agent a determines the behavior. Access to a particular element in the strategy vector is given by the following convenience notation: $s_a(parameter)$ denotes *parameter*. Note that the dimension of s may vary between different agents.

3.2 Distributed Optimization Problem

The task of optimizing can be formulated as follows:

Definition 3 (Distributed Optimization Problem (DOP)). *A distributed optimization problem is given by:*

$$\begin{aligned} \textit{minimize} \quad & f(s_A), \\ & \textit{subject to } \gamma(s_A) \end{aligned} \tag{2}$$

where $s_A = (s_1^{a_1}, s_2^{a_1}, \ldots, s_t^{a_1}, \ldots, s_1^{a_z}, s_2^{a_z}, \ldots, s_u^{a_z}) \in S_A^*$ *is the strategy vector of MAS and* S_A^* *is the search space. The objective function is* f *and there are* v *constraints* $\gamma_i(s_A), i = 1, \ldots, v$ *imposed on* s_A*.*

The strategy vector s_A is distributed among A and the objective function f is unknown to A and therefore cannot be calculated by any single agent. The set $\gamma(s_A)$ contains constraints of the DOP distributed over A. The combination of local strategy vectors of all agents s_A forms the strategy vector and is also the distributed solution of the DOP. If constraints vary over time t, the DOP becomes even harder. The dimensions of s_A are not fixed, rather they may vary over time, as agents enter or leave the system.

3.3 Integrating an Economic Perspective

In our model we assume a discrete timeline, where all actions take place at consecutive steps. Further, agents have to pay a tax $\mathcal{T}_a(t)$ to the environment E at every time t for their actions and according to their strategy parameters s_a. Given the tax, we can calculate the profit π_a an agent a receives at time t by

$$\pi_a(t) = \mathcal{R}_a(t) - \mathcal{P}_a(t) - \mathcal{T}_a(t) \tag{3}$$

where $\mathcal{P}_a(t)$ denotes the payment a has to pay to other agents or the environment in order to follow its strategy s_a, $\mathcal{T}_a(t)$ denotes tax turned over to the environment and finally a may have receipts $\mathcal{R}_a(t)$ from previously executed actions. Based on the profit $\pi_a(t)$, an agent accumulates funds c_a over time expressed by:

$$c_a(t+1) = c_a(t) + \pi_a(t) \tag{4}$$

where $c_a(t+1)$ denotes the funds of agent a at time $(t+1)$, $c_a(t)$ denotes a's funds at time t and $\pi_a(t)$ is the profit of a at time t. Money cannot be 'created' by the agent, rather it is provided by the environment E representing the demand $\mathcal{D}$ of the market since it provides a specific amount of funds in return to services offered by the set of agents A.

4 Evolutionary Computation as Decentralized Adaptation Method

The aim of this paper is to show an decentralized optimization approach for distributed problems lined out in section 3.2. Our assumptions on a system that applies this method are 1) Agents have the ability to communicate and to sense their environment at least locally 2) There is no central manager, instead the DOP and its solution is distributed among A 3) There is a limited amount of money ($\mathcal{D}$) that will be provided by the environment E.

The previous section has shown a market-based multi-agent system that consists of economic agents who have to solve a distributed problem DOP collaboratively and fulfill previous prerequisites. In this section we re-use evolutionary algorithm theories of Holland [11,13] concurrently with a decentralized economic agent perspective given in [5] as well as our economic market perspective as we believe that a decentralized market mechanism can be a profound approach to replace the central fitness calculation in evolutionary algorithms. There are a variety of evolutionary computational models that have been proposed and studied that we will refer to as evolutionary algorithms [11,13]. In our agent based approach we turn the typical evolutionary algorithm software design on its head by breaking up any central instance and move the genetic representation as well as the operators to the agents respective individuals itself as proposed in [12].

Local reproduction by an agent includes all steps and operations necessary to produce new offspring, such as local search, recombination and mutation. We introduce an agent specific variable θ that serves as a threshold and enables agents to reproduce. Whenever the funds of an agent exceed this threshold ($c_a \geq \theta$) it will reproduce. Advantages of local versus global reproduction are discussed in [14] in detail. As we want to focus on emerging global effects of local reproduction, based on [15] an agents algorithm motivated by modeling ecologies of organisms adapting in environments is formulated as follows:

Algorithm 1. Main loop, every agent executes asynchronously

Require: s and c from parent
initialize m
loop
  get sensory inputs I
  execute m
  set effector outputs O
  calculate c using equation (4)
  if $c \geq \theta$ **then**
    reproduce and split c with child
  else if $c < 0$ **then**
    die
  end if
end loop

An agent a is set up with its strategy s_a and funds c from its parent. The initial population is created by assigning a random strategy and an initial reservoir of funds to every agent. Once started, an agent will continuously follow his strategy by executing m. Whenever c reaches θ the agent will reproduce (line 7-8) and create a new agent. In the opposite case, if an agent dies, his funds drop below zero ($c < 0$, line 9), construed to signify the bankruptcy of an inefficient agent. Therefore the notion of 'generation number' does not exist and instead a perpetual reproduction and replacement of agents takes place. This mechanism makes survivor selection unnecessary and population size an observable. This can be regarded as a combination of biological and economical ideas [5], because no rational economic agent would create its own competition. However, in biology, it is the case.

Given the model in chapter 3 and algorithm 1 it is clear that the distribution of money within the system differs among the agents. Any agent selfishly tries to exploit the limited money based on its own strategy; however we assume an agent has no particular intelligence about how this strategy relates to its own success. In the following sections a short description of the two basic properties of our model is given: adaptation in the number of agents and their strategy. A more extensive investigation is given in [16].

4.1 Adaptation of the Number of Agents

Following the statement in [7] in natural environments the population size tend to stabilize around the carrying capacity of the ecosystem we define the carrying capacity as demand $\mathcal{D}$. As outlined in [11] the system can only be stable for bounded input when $\mathcal{D}$ does not grow in magnitude of its own accord.

For further analysis we approximate the creation of agents from discrete by a continuous adaptation in population growth. For a particular agent a the ratio $\frac{\pi_a(t)}{\theta}$ is an expression of the estimated creation of new agents at time t. In fact, the number of agents increase/decrease stepwise and not continuously, depending on the distribution of $\Pi(t) = \sum_{a \in A} \pi_a(t)$ and the funds in A.

Dividing the overall profit $\Pi(t)$ by θ we get the average number of new/dying agents at time $t+1$ (again, averaged for sufficient long runs) in equation 5

$$|A(t+1)| = |A(t)| + \frac{\Pi(t)}{\theta} \tag{5}$$

Thus $|A(t+1)|$ vary by factor $\frac{\Pi}{\theta}$ compared to $|A(t)|$, where $|A(t)|$ is the number of agents in A at time t. If $\frac{\Pi}{\theta}$ is positive (negative), the number of agents will increase (decrease) depending on Π. This emergent behavior can be observed in real scenarios [2] where actors enter and leave the market. The rate at which agents enter or leave the market is directly correlated with the overall profit (see equation 5) and the market supply $\mathcal{S}$ and demand $\mathcal{D}$ in such a scenario. Even without any central control this effect can be observed.

4.2 Distributed Problem Optimization by Spread of Successful Strategies

Since in our model there is no central instance that can rank and compare the agents funds, no central selection based on fitness can be calculated and performed. Therefore the system must evolve fitter strategies in an emergent self-organizing way.

For the following discussion we assume that strategies were reproduced in a pure way without disturbance of mutation or recombination. The proportion of strategy s_a in $A(t)$ is $p_a(t) = \frac{1}{|A(t)|}$, where $|A(t)|$ denotes the number of agents. Based on evolutionary pressure the agent population profit Π reaches zero after a sufficient amount of time and for long running simulation we can set $\lim_{t\to\infty} \Pi(t) = 0$. According to equation 5 the number of agents in the next time step can be rewritten by $|A(t+1)| = |A(t)|$ and the expected proportion of s_a at time step $t+1$ is as follows:

$$p_a(t+1) = \frac{1 + \frac{\pi_a(t)}{\theta}}{|A(t)|} \tag{6}$$

where $1 + \frac{\pi_a(t)}{\theta}$ is the original strategy plus the estimated additional quantity of s_a that together forms the number of samples of strategy S_a in $A(t+1)$. It follows, that based on the proportion $p_a(t+1) > p_a(t)$ for a positive profit the part of s_a in A grows. In words, the proportion of a particular strategy s_a grows as the ratio of the average profit π_a of agent a. Strategies causing a positive/negative profit will receive an increasing/decreasing number of replications in the next time step.

It follows that strategies inducing a positive profit on the one hand may have to pay less tax compared to below zero profit strategies due to the usage of less resources or more efficient resource utilization. Both are needed to streamline logistic networks while simultaneously improving service to the customer. Based on equation 3 one can see that tax $\mathcal{T}$ is that part of an agents profit determining variable which is not explicitly related to the flow of goods. With tax the models basic conditions can be set respective controlled and the agent population will adapt to it, if tax is not too high. Therefore, an intrinsic property of systems using our model is the constant search for better resource utilization.

5 Case Study: The p-Median Problem

We include a case study that illustrates the successful application of decentralized EC-enabled economic agents. It consists in positioning facilities on a discrete map by using two p-median sample problem sets presented in [9,10]. The set of locations L is given by the problem set and facilities are represented by agents. The agents strategy vector consists two values: the location $s_a(location) \in L$ and the profit factor $s_a(profit) \in \mathbb{R}$. An agent a offers service to customer at location $j \in L$ according to $cost_{aj} = w_{s_a(location)j} * s_a(profit)$. Customers choose always the agent with lowest $cost$ and the selection is expressed as

$$\sigma_{aj}(t) = \begin{cases} 1 & \text{customer } j \text{ is served by agent } a \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

Multiple agents can offer their service at the same location l and the 'cheapest' agent with lowest *cost* at location l is defined as $\dot{a}^l$. The set of cheapest agents is defined as $\dot{A} \subseteq A$ with $s_{\dot{a}_i}(location) \neq s_{\dot{a}_j}(location), \dot{a}_i, \dot{a}_j \in \dot{A}$ and considered as the set of facilities P that form the solution to the p-median problem given in equation 1. For simplicity reasons the set of $\dot{A}$ is denoted with A, if not stated otherwise. The p-median problem specific DOP is given as

$$\text{minimize } f(s_A) = \sum_{j=1}^{l} \sum_{a=1}^{A} w_{s_a(location)j} \tag{7}$$

$$\text{subject to} \quad \pi_a(t) = \mathcal{R}_a(t) - \mathcal{P}_a(t) - \mathcal{T}_a(t)$$

$$\mathcal{R}_a(t) = \sum_{j=1}^{l} \sum_{a=1}^{A} cost_{aj}(t) * \sigma_{aj}(t) \tag{8}$$

$$\mathcal{P}_a(t) = 0 \tag{9}$$

$$\mathcal{T}_a(t) = 10 + \sum_{j=1}^{l} \sum_{a=1}^{A} w_{aj} * \sigma_{aj}(t) \tag{10}$$

According to equations 8 and 10 the profit $\pi_a(t)$ is basically dependent on $s_a(profit)$. The fixed tax of 10 currency units is necessary to slowly remove agents with no income, e.g. for $a \notin \dot{A}$, otherwise they would remain in the system for ever and consuming resources. As the fixed tax is negligible, a profit factor $s_a(profit) > 1$ is important and induces an evolutionary pressure on agents to evolve a strategy with $s_a(profit)$ slightly over 1. Otherwise an agent will be removed based on a negative profit as well as turning inactive ($a \ni \dot{A}$). There is no direct ressource transfer among the agents (eq. 9) in this particular application but money is transferred between customers and agents. Splitting threshold θ is dynamic calculated as the average income over the last two time steps since a fixed θ would need to be adjusted for every problem.

The initial setting is a population size of 20 agents, a mutation rate of 0.03, 10000 time steps and a profit factor of 2. It is important to set the initial profit factor high enough compared to expected payments in order to get a running system. During the simulation an asymptotic convergence of the profit factor to 1 is expected. The agents strategy is represented as an integer/float vector that can be evolved using the common variable-length binary valued crossover and mutation operators [11]. We use uniform crossover and an implementation of the Breeder Genetic Algorithm [17] for mutation throughout all runs. Selection is disabled and results are obtained by averaging 50 simulation runs.

In figure 1 the results of two different problem sets are shown and costs for different p values are compared to optimal values and random search. At the end of each independent run all pareto-optimal values explored during the run are averaged (The averaged values are not necessary pareto optimal). During simulation the agent population explores different values of p as the population

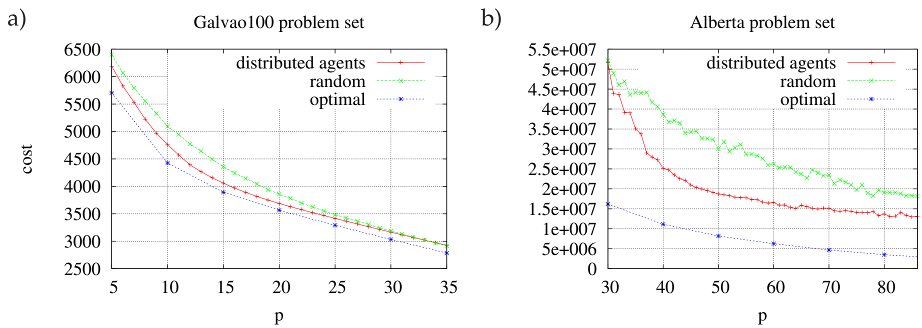

Fig. 1. Distributed agents approach compared with random search and optimal values for multiple p on two problem sets (Galvao: 100 nodes, Alberta: 316 nodes)

size is changing. The graphs show clear differences between distributed agents approach and random search. There are, however, significant differences in search space exploitation for different p values in the distributed agents approach that needs further investigations in different tax and starting conditions.

6 Conclusion and Further Directions

This paper has addressed the potentials of distributed evolutionary systems whereas a market-based multi-agent approach is used. All necessary steps of the evolutionary algorithm have been moved to the agents itself which allows a fully decentralized approach. We have shown with our approach, that even in a decentralized system without central control an adaptation occur, caused by two basic properties. First the number of agents $|A|$ adapt according to the demand provided from environment E and second, successful strategies spread in the population of agents which form together a distributed and optimized solution. The optimization is based on funds provided by E that spread within the system and allows local selection and reproduction by the agents. The two effects indicate that an agent population and their resource utilization adapt to a pre-defined demand and will constantly continue to adapt thereby satisfying the demand with lower resource utilization. We have applied our method on the p-median problem. Then, it was compared to random search. The results show, that the distributed approach is clearly outperforming random search and is also a very promising perspective for distributed problems.

Further research in this line will pursue the development of a generalized model of complex adaptive systems using an economics-enabled market-based evolutionary approach. One aspect that we have not addressed in this paper is the comparison of the results against classical and multi objective evolutionary algorithms. Next, we will look for relations to bi-level or multi-level optimization/adaptation. We also hope to address other useful application fields for this method, e.g. in the context of grid computing and agent systems, artificial immune systems, life-cycle management and other logistics problems.

References

1. Durfee, E.H., Lesser, V.R., Corkill, D.D.: Trends in cooperative distributed problem solving. IEEE Transactions on Knowledge and Data Engineering 1(1), 63–83 (1989)
2. Graudina, V., Grundspenkis, J.: Technologies and multi-agent system architectures for transportation and logistics support: An overview. In: International Conference on Computer Systems and Technologies - CompSysTech, Varna, Bulgaria (2005)
3. Cai, W., Turner, S.J., Gan, B.P.: Hierarchical federations: an architecture for information hiding. In: PADS 2001: Proceedings of the fifteenth workshop on Parallel and distributed simulation, pp. 67–74. IEEE Computer Society, Washington (2001)
4. Hohl, F.: A framework to protect mobile agents by using reference states. Technical Report 2000/03, Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR) (March 2000)
5. Smith, R.E., Bonacina, C., Kearney, P., Eymann, T.: Integrating economics and genetics models in information ecosystems. In: Proceedings of the 2000 Congress on Evolutionary Computation CEC 2000, La Jolla Marriott Hotel La Jolla, 6-9 2000, pp. 959–966. IEEE Press, California (2000)
6. Eymann, T.: AVALANCE - Ein agentenbasierter dezentraler Koordinationsmechanismus für elektronische Märkte. PhD thesis, Universität Freiburg (2000)
7. Eiben, A., Schut, M.: New Ways to Calibrate Evolutionary Algorithms. In: Advanced in Metaheuristics and Optimization, Springer, Heidelberg (2007)
8. Hosage, C.M., Goodchild, M.F.: Discrete space location-allocation solutions from genetic algorithms. Annals of Operations Research 6(2), 35–46 (1986)
9. Alp, O., Drezner, Z., Erkut, E.: An efficient genetic algorithm for the p-median problem. Annals of Operations Research 122(1-4), 21–42 (2003)
10. Galvão, R.D., ReVelle, C.: A lagrangean heuristic for the maximal covering location problem. European Journal of Operations Research 88, 114–123 (1996)
11. Goldberg, D.E.: Genetic Algorithms in search, optimization, and machine learning. Addison Wesley Longmann, Inc., Reading (1989)
12. Smith, R.E., Taylor, N.: A framework for evolutionary computation in agent-based systems. In: Looney, C., Castaing, J. (eds.) Proceedings of the 1998 International Conference on Intelligent Systems, pp. 221–224. ISCA Press (1998)
13. Holland, J.H.: Hidden Order: How Adaptation Builds Complexity, vol. 9. Addison Wesley Publishing Company, Reading (1996)
14. Menczer, F., Degeratu, M., Street, W.: Efficient and scalable pareto optimization by evolutionary local selection algorithms. Evolutionary Comp 8(3), 223–247 (2000)
15. Menczer, F., Belew, R.K.: Adaptive retrival agents: Internalizing local context and scaling up to the web. Mach. Learn. 39(2-3), 203–242 (2000)
16. Otto, S., Kirn, S.: Evolutionary adaptation in complex systems using the example of a logistics problem. International Transactions on Systems Science and Applications 2(2), 157–166 (2006)
17. Mühlenbein, H.: The breeder genetic algorithm - a provable optimal search algorithm and its application. In: Colloquium on Applications of Genetic Algorithms, IEEE, London, vol. 67, p. 5/1– 5/3, IEEE, London (1994)