AGENT-BASED AND EVOLUTIONARY PLANNING TECHNIQUES SUPPORTING CRISES MANAGEMENT IN TRANSPORTATION SYSTEMS

Rafał Dreżewski, Marek Kisiel-Dorohinicki, Jarosław Koźlak

Institute of Computer Science
AGH University of Science and Technology
Al. Mickiewicza 30, Kraków, Poland
e-mail: {drezew,doroh,kozlak}@agh.edu.pl

Abstract:
The paper deals with a particular approach to crises management in transportation systems. The considerations are based on a layered reference architecture dedicated to monitoring and management of multi-agent systems. In the contribution agent-based and evolutionary approaches are proposed as a basis for the simulation layer realisation. Several variants are shortly discussed and illustrated by selected experimental results.

Keywords: transportation systems, multi-agent systems, crisis management.

1. INTRODUCTION

During the last decade the idea of an intelligent autonomous agent gains more and more interest both in academic community and in industry. A constantly increasing number of computer systems are being analysed and designed in terms of agents. Agents play a key role in integration of AI sub-disciplines, which is often necessary to design and build modern intelligent systems. Agent technology is used in various domains, providing concepts and tools for development of complex, distributed and decentralised systems.

The systems under consideration may both be designed from scratch as multi-agent ones (operating in the virtual world, e.g. network information services, virtual enterprises), as well as function in the reality as a set of cooperating autonomous subsystems of whatever origin (e.g. transportation systems, industrial complexes). Acceptance of the agent-based approach opens possibility for solving many problems that until now has been tractable only with respect to tightly coupled centralized systems. Some of these problems are risk and critical situations (states) analysis (Wu and Soo, 1999; Collins *et al.*, 1999).

The problem critical situations analysis with respect to transportation systems will be of our special interest here. Obviously effective construction of a transport planning allows companies to highly limit sustained costs and be more competitive on the market. Therefore, an important challenge is to create tools, which support development of such planning on the basis of acquired knowledge on available transport resources, incoming transport requests and road network structure. One of the applied approaches is simulation research, which facilitates selection and configuration of transport planning algorithms.

Based on a general scheme of crises management in multi-agent systems (MAS), as well as preliminary results obtained in the field of transportation systems (Nawarecki *et al.*, 2005), a variety of possible variants of planning-support techniques are considered in the paper. Section 2 introduces fundamental concepts

[1] This work was partially sponsored by State Committee for Scientific Research (KBN) grant no. 3 T11C 025 27

of the approach—a reference architecture for crises management for MAS. Section 3 presents selected approaches to solving transportation problems, with special attention to evolutionary algorithms and agent-based systems. And finally section 4 describes several realisations and selected results of performed experiments.

2. MANAGEMENT OF CRITICAL SITUATIONS IN MULTI-AGENT SYSTEMS

Multi-agent systems (both real and virtual) are marked by the possibility of arising critical situations that can be caused by both outer (e.g. undesirable interference, the forces of nature) and inner (e.g. resource deficit, local damages) factors. *Crisis* is interpreted here as a threat of loss (partial or complete) of the system functionality. Based on the principal assumptions of MAS operation an architecture was proposed, which seems to be general enough to be used as a reference one for describing crises management activities (Nawarecki *et al.*, 2005).

The architecture is actually a four-layer one as presented in fig. 1. The bottom layer (MAS) constitutes the system under consideration. The directly higher layer (Monitoring) consists of agents that are assigned to gathering information about the subject system by inquiring and observing done according to the agent paradigm (Kisiel-Dorohinicki, 2005). An agent-based simulation constitutes the next layer (vMAS), which aims at foreseeing future states of the system based on the monitoring data. The main purpose of the upper monitoring layer is the evaluation of situations (states) arising in the course of simulations carried out using vMAS. It is assumed that selected results can be applied as a direct management or influence on mechanisms (e.g. organization) of the system. Thus the agents of the upper layer may be equipped with the ability of decision making and, in turn, have an effect on the system. This may create a loop of semi-automatic prevention of crises in the proposed architecture.

The agent-based model of the considered system situated at the third layer (vMAS) is of our special interest here. Its agents try reconstruct future states of the

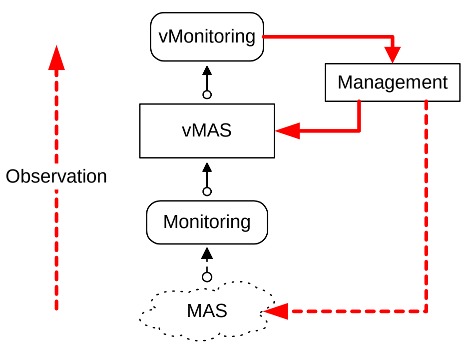

Fig. 1. MAS management structure

system, simulating the possible behaviour of the real one using the monitoring data, in various aspects and different time horizons. These scenario-based studies are carried out aiming at critical situations detection and search for an anti-crisis policy. The upper layer is designed to monitor vMAS and the main purpose of its agents is the evaluation of situations (states) arising in the course of simulations. The idea is that the investigation of a reach enough bunch of scenarios leads to finding the strategy of avoiding the crisis in the system or, at least, reducing its effects. Of course the evaluation may evoke a need of communication among the agents.

The simulation layer may have different structure and may employ a variety of techniques accordingly to the particular application area. A review of different possibilities in case of transportation systems is given in the next section.

3. SOLVING TRANSPORTATION PROBLEMS

Transportation problems like Vehicle Routing Problem with Time Windows (VRPTW) or Pickup and Delivery Problem with Time Windows (PDPTW) are defined by a set of known transport requests to be performed with the least resources available and time consumption (Desaulniers *et al.*, 2000; Mitrowic-Minic, 1998). Time windows restrict the acceptable time periods of service realization at visited points. The maximum capacities of vehicles cannot be exceeded by the overall load. The routes of vehicles should start and finish at a given depot point. In VRPTW it is assumed that each transport request is described by one location which should be visited by a vehicle respecting given constraints. The loads are delivered either from a starting depot to different destinations or from different starting locations to one destination. In PDPTW each transport request is expressed by two locations, which are visited by the same vehicle: a pickup and delivery point.

For both VRPTW and PDPTW the solution consists of a set of routes associated with particular vehicles. Each route contains a list of visited request points with information concerning the time of arrival and departure. Solving *static* problems consists of calculating the optimal routes so as to service all requests from a fixed set. Because of high computational complexity of exact methods, approaches based on various heuristics are often used. During the first phase of computation, initial solutions are built using different versions of construction heuristics, which insert subsequent request points into the routes (they differ mainly in the order of insertions). After the construction phase, the optimisation occurs, which is done e.g. by tabu search, evolutionary algorithms or simulated annealing. Operations of solution modification are based on emptying some routes (by moving request points to other routes), different versions of request exchange among routes, or changing the order of requests within a route. In *dynamic* problems it is assumed that requests may also arrive while the system is running, which makes dynamic modification of vehicle routes necessary. Thus the modelling of vehicle location also has to be considered. In this case some modifications of the above algorithms are used, e.g. in (Gendreau *et al.*, 1998) a parallel system using tabu search with adaptive memory is presented.

Evolutionary algorithms are based on iterative transformation of the *population of individuals* representing the set of potential solutions of the given problem. Evolution consists on generating consecutive generations, using so called *genetic operators* (or *variation operators*) and the process of *selection*. The process of evolution should tend to generate better individuals and finally to find the needed (usually approximate) problem solution. Yet, evolutionary algorithms often suffer from the loss of population diversity, often resulting in the premature convergence, which means locating the basin of attraction of local optima instead of a global one. This is especially important considering transportation problems like VRPTW or PDPTW, because of introduced constraints, which practically eliminate many new individuals from the population.

Niching methods are aimed at forming and stably maintaining subpopulations (species) throughout the search process, thereby allowing to locate multiple basins of attraction of local minima (Mahfoud, 1995). Various techniques have been proposed that allow for species formation via the modification of the mechanism of selecting individuals for new generation (*crowding*), or the parent selection mechanism (*fitness sharing*). *Sexual selection* also applies to the parent selection mechanism and is based on individuals' preferences (Ratford *et al.*, 1997). Another possibility is the restricted application of selection and/or recombination mechanisms e.g. by introducing the environment with some *topography* within which the individuals are located (Cantú-Paz, 1998).

In *co-evolutionary algorithms* the fitness of each individual depends not only on the quality of solution to the given problem but also (or solely) on other individuals' fitness. Co-evolutionary techniques are aimed at overcoming limited adaptive capacity of evolutionary algorithms resulting from the loss of useful diversity of population. As the result of ongoing research quite many co-evolutionary techniques have been proposed. Generally, each of these techniques belongs to one of two classes: competitive or cooperative (Paredis, 1995).

A multi-agent approach is based on the cooperation of intelligent, autonomous elements, called agents. Each agent constructs a plan so as to accomplish its goal interacting with other agents. This should lead to the accomplishment of system goals. Multi-agent systems for transport planning and scheduling offers additional features in comparison to meta-heuristic approaches. A model of the problem is developed for the purpose of being as close to real life as possible. For example, the components of the vehicles, the available drivers, and the loading and unloading process specifications for different kinds of cargoes are considered. Some elements (like drivers) need a degree of autonomy which favours the multi-agent approach (e.g. they should rest if they feel very tired, or take decisions concerning detours if necessary). The other important feature is a possibility of keeping information by agents representing competing companies or other bodies confidential. Below is a review of some existing multi-agent systems for transport planning and scheduling.

MARS system (Fischer *et al.*, 1996) is realized to simulate planning and scheduling for a society of shipping companies. Agents represent transportation companies and vehicles. Protocol Contract Net (Smith, 1980) as well as its extension is applied to assign requests to particular vehicles. Simulated trading (Bachem *et al.*, 1994) is used for dynamic re-planning or for optimization of the current solution. The MARS system also makes possible to take into consideration modifications of travel time because of traffic jams.

TeleTruck (Burckert *et al.*, 1998) is a distributed multi-agent system to support dispatch officers in shipping companies. The architecture is based on the concept of holonic agents, agents may build temporary, complex structures composed of several agents, and closely cooperate loosing to some degree their autonomy. There are several types of agents (driver, truck, trailer, container, chassis) which possess and manage specific types of resources (driving time, motor, chassis, loading space). Such an approach makes it possible to consider different aspects of transport problems like storing, transportation, management of cargoes, drivers, trucks etc.

4. REALISED SYSTEMS AND OBTAINED RESULTS

There exists a number of benchmarks for testing the quality of algorithms solving static problems, both PDPTW, prepared by Li and Lim, as well as VRPTW, prepared by Solomon and by Gehring and Homberger (*Benchmarks - Vehicle Routing and Travelling Salesperson Problems*, 2004). The benchmarks differ in the number of customers (transport requests) to be served, spatial distribution of requests (positioned randomly—LR, in clusters—LC or with mix of random and clustered positions—LRC), scheduling horizon (the number of requests, which should be served by a vehicle) and the size of time windows. Dynamic problems were generated on the basis of static benchmarks in such a way that the given percentage of requests from a static problem was added randomly during simulation.

Several techniques based on the evolutionary and multi-agent approach were utilized in the realised implementations of the simulation layer (vMAS) dedicated for the crises management in transportation systems. As it was already stated, *static* approaches do not consider the previously elaborated patterns of management, and define the strategy to be applied from scratch, based on the actual state of the system. Thus they may utilize a variety of existing methods and tools, but their computational complexity must be taken into account. Yet, which is of vast importance here, in static problems, the possibility of researching crisis situations is rather limited. One can introduce into the algorithm some different conditions/constraints which, if fulfilled, improve the security level offered by the solution and make it easier to find an alternative solution e.g. when a car is out of order. It is possible to consider, as an example, a condition which promotes balanced distances among vehicles in each time point or make slacker routes schedules, which allows the realization of future additional requests.

In the **island model of parallel evolutionary algorithm** developed for static PDPTW solving a high diversity of the population is guaranteed thanks to the introduction of islands, on which two different evolutionary algorithms are applied. There are a number of differences among these algorithms. First of all they concern a different solution representation method (a traditional representation, containing points of pickup and delivery, or a limited representation, which contains only pickup points, the delivery points only being added when the value of the quality function is calculated). They also differ in the application of a virtual route, that temporarily stores those requests that highly deteriorate the quality of other routes.

Experiments were performed using the different example problems. 25 percent of the test results are equal to the best solutions obtained so far, and 40 percent of test results had equal route numbers but slightly higher total distances. In the fig. 2 selected results (number of vehicles used in the solution and total travel distance) of tests for problems with 200 customers are presented, results obtained for tests with 100 customers were better.

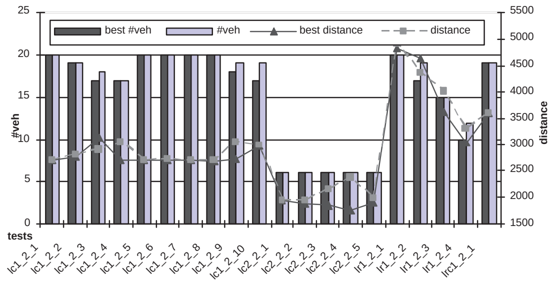

Fig. 2. Island model for static PDPTW: solutions of 200-customers problems

Another approach utilizes a **multi-agent system**, which serves as an environment where many optimization processes are run concurrently, solving the PDPTW transport problem. This approach makes it possible to explore different parts of a solution space. Parts of solutions are exchanged among agents and agents which provide solutions estimated as valuable, are rewarded.

In the constructed multi-agent system, each agent represents a solution of the problem, which is then subjected to an optimization process. The agents may apply different optimization algorithms, but in the current realization they perform tabu search algorithms, based on solutions presented in (Li and Lim, 2001) but with the utility functions expressed by different equations.

Obtained results do not differ from the best know solutions. For 46 percent of the 100 customer tests and 35 percent of the 200 customer tests, the results were equal to the best solutions obtained so far, and for the rest, an average diversity regarding the best solutions is equal to 6 percent. In the fig. 3 are presented obtained solutions (number of vehicles used in the solution and total travel distance) for the problems with 200 customers.

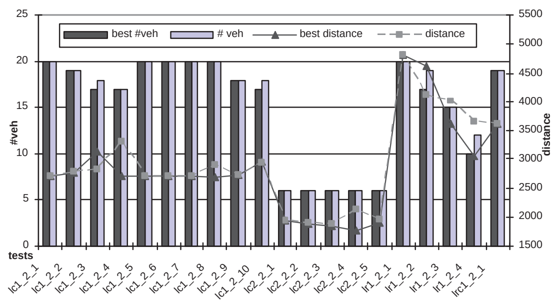

Fig. 3. MAS for static PDPTW: solutions of 200-customers problems

The agents may examine different test examples (benchmarks) concurrently. Some of them may be modified so as to represent expected crisis situations. The choice of different utility functions may promote a lower number of vehicles or shorter distances. The latter may accept an increased number of vehicles whose routes are short, so that these vehicles have a higher chance of serving additional requests when other vehicles are out of order. The important factor may be to promote a construction of special route configurations e.g. the ones where two vehicles are not far one from another at any given point in time, which may make it easier to take over the requests in cases of breakdown.

On the contrary to static approaches, *dynamic* ones try to modify the current management policy, and thus they need specific mechanisms yet should be computationally less complicated. But first of all they form a natural means for development and analysis of anti-crises strategies.

Co-evolutionary algorithm with island model proved especially useful in the case of dynamic vehicle routing problems, where the configuration of clients is changing during simulation. Such techniques can much improve solutions generated by evolutionary algorithm by strengthening their adaptive capabilities.

The co-evolutionary algorithm used is based on two populations: *A* and *B* (Machado *et al.*, 2002). The individuals from the first one have the numbers of clients in each route coded in their genotypes. The individuals from the second one have clients coded in their genotype. The complete solution is the pair of individuals in which each of them comes from different population. The fitness value of individual from the *A* population a^A is the average fitness of pairs containing individual a^A and individuals coming from a group of individuals selected from the population *B*. The individuals from population *B* are evaluated in analogical way. The fitness function for the pair of individuals is given by: $\varphi(a^A, a^B) = penalty \cdot nroutes + totdist$, where $nroutes$ is the number of vehicles used, and the $totdist$ is the total distance. Tournament, roulette wheel and deterministic selection mechanism are used. The partially mapped crossover (PMX) operator is used in both populations, shift mutation is used in population *B*, and uniform mutation in population *A* (detailed description of operators may be, for example, found in (Michalewicz, 1996)). Also some experiments with the use of local optimization technique were carried out.

In order to introduce additional population diversity the island model (Cantú-Paz, 1998) is used. In each island there is different fitness function: fitness of individual *a* is the maximal fitness of pairs within which it was evaluated or average fitness described above is used. Also, the *penalty* parameter have different value on each island. The best individuals are allowed to migrate between islands.

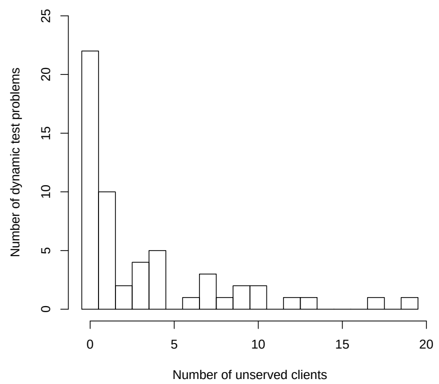

Fig. 4. Frequencies of the number of unserved clients in all dynamic benchmark problems

In the case of dynamic VRPTW problems there is no commonly used test problems and quality of solution measures, so the number of unserved clients was used as the measure of how good the solution found by co-evolutionary algorithm is. In the case of dynamic problems generated on the basis of *C* Solomon's problems there was about 6% of unserved clients, in the case of *R* problems there was about 2.2% of unserved clients, and in the case of *RC* problems there was about 1.6% of unserved clients. The frequencies of the numbers of unserved clients for all Solomon's problems can be seen in fig. 4.

Another approach uses evolution strategy with niching technique based on **sexual selection** mechanism. Niching technique based on sexual selection can make evolutionary algorithm easily adapt to constantly changing conditions in the case dynamic vehicle routing problems. In the considered case each individual's genotype is composed of two chromosome's. The first one represents a solution and the second one is composed of evolution operator's parameters (number of mutations performed during mutation phase and the indicator of the goal of mutation operators: minimization of vehicles or minimization of total distance), as well as the sex marker (female or male). During the reproduction phase individuals are sorted (within each sex separately) on the basis of their fitness values. Individuals are then paired in such a way that best male individual is paired with best female individual, second male with second female and so on. The offspring is produced with the use of recombination (RBX and SBX) and mutation (LSM) operators (Michalewicz, 1996).

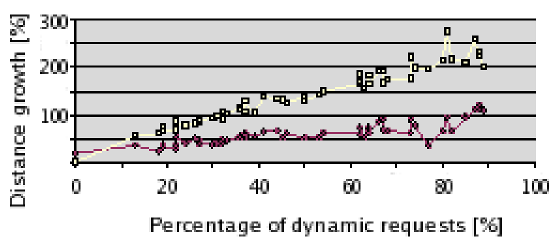

Fig. 5. Percentage of distance growth (as compared to the best known solution for static problems) for different percentages of dynamic requests—the average values for all benchmark problems. Comparison of algorithms with (darker line) and without (brighter line) sexual selection

Evolutionary algorithm with sexual selection was compared to the evolutionary algorithm without any niching technique. On the average, the algorithm with sexual selection generated better solutions (about 50% shorter total distance) than evolutionary algorithm without niching technique. Also, when the percentage of dynamic requests grows the total distance grows much slower in the case of evolutionary algorithm with sexual selection than in the case of evolutionary algorithm without any special mechanism (see fig. 5).

5. CONCLUDING REMARKS

The article is concerned with the application of agent and evolutionary approach to the problem of management of critical situations in transportation systems. Several techniques are discussed that solve classically defined VRPTW and PDPTW problems used for evaluation of plans that concern future behaviour of the system in consideration.

Future work will concentrate on the applications to crises management in real transportation enterprises of different organisation. The implementation of crisis situations simulator is foreseen, allowing for elaborating different scenarios (for example, changing travel time between two clients, priorities of clients, shifting time windows, and vehicle's fault). More detailed comparison of the algorithms in the case of crisis situations in VRPTW and PDPTW will follow, in order to justify and deepen solutions and conclusions elaborated so far.

Acknowledgements. We would like to thank the students of Computer Science at AGH-UST for the participation in the implementation of the systems descibed in this paper.

REFERENCES

Bachem, A., W. Hochstattler and M. Malich (1994). Simulated trading a new parallel approach for solving vehicle routing problems. In: *Proc. of Int. Conf. on Parallel Computing: Trends and Applications*.

Benchmarks - Vehicle Routing and Travelling Salesperson Problems (2004). Technical report. SINTEF.

Burckert, H.-J., K. Fischer and G. Vierke (1998). Transportation scheduling with holonic MAS – the teletruck approach. In: *Third Int. Conf. on Practical Applications of Intelligent Agents and Multiagents (PAAM'98)*.

Cantú-Paz, E. (1998). A survey of parallel genetic algorithms. *Calculateurs Paralleles, Reseaux et Systems Repartis*.

Collins, J., M. Tsvetovas, R. Sundareswara, J. van Tonder, M. Gini and B. Mobasher (1999). Evaluating risk: flexibility and feasibility in multi-agent contracting. In: *Proc. of the Third Int. Conference on Autonomous Agents (Agents'99)* (O. Etzioni, J.-P. Müller and J. M. Bradshaw, Eds.). ACM Press.

Desaulniers, G., J. Desrosiers, A. Erdmann, M. M. Solomon and F. Soumis (2000). The vrp with pickup and delivery. Les Cahiers du GERAD.

Fischer, K., J.P. Muller and M. Pischel (1996). Cooperative transportation scheduling: an application domain for dai. *Applied Artificial Intelligence* pp. 1–33.

Gendreau, A., F. Guertin, J. Potvin and R. Séguin (1998). Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. Technical report CRT-98-1, University of Montreal.

Kisiel-Dorohinicki, M. (2005). Monitoring in multi-agent systems: Two perspectives. In: *Monitoring, Security, and Rescue Techniques in Multi-Agent Systems* (B. Dunin-Kęplicz, A. Jankowski, A. Skowron and M. Szczuka, Eds.). Springer-Verlag.

Li, H. and A. Lim (2001). A metaheuristic for the pickup and delivery problem with time windows. In: *Proc. of 13th IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI'01), Dallas, USA*.

Machado, P., J. Tavares, F. B. Pereira and E. Costa (2002). Vehicle routing problem: Doing it the evolutionary way. In: *Proc. of the 2002 Genetic and Evolutionary Computation Conf.* (W. B. Langdon, Ed.). Morgan Kaufmann.

Mahfoud, S. W. (1995). Niching methods for genetic algorithms. PhD thesis. University of Illinois at Urbana-Champaign.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag.

Mitrowic-Minic, S. (1998). Pickup and delivery problem with time windows: A survey. Technical Report 1998-12. SFU CMPT TR.

Nawarecki, E., J. Koźlak, G. Dobrowolski and M. Kisiel-Dorohinicki (2005). Discovery of crises via agent-based simulation of a transportation system. In: *Multi-Agent Systems and Applications IV* (M. Pěchouček, P. Petta and L.Z. Varga, Eds.). Vol. 3690 of *LNAI*. Springer-Verlag.

Paredis, J. (1995). Coevolutionary computation. *Artificial Life*.

Ratford, M., A. L. Tuson and H. Thompson (1997). An investigation of sexual selection as a mechanism for obtaining multiple distinct solutions. Technical Report 879. Univ. of Edinburgh.

Smith, R. G. (1980). The contract net protocol: high-level communication and control in a distributed problem solver. *IEEE Transactions on Computer* pp. 1104–1113.

Wu, S. and V. Soo (1999). Risk control in multi-agent coordination by negotiation with a trusted third party. In: *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99-Vol1)* (D. Thomas, Ed.). Morgan Kaufmann Publishers.