Towards Agent-Based Evolutionary Planning in Transportation Systems*

Jarosław Koźlak, Marek Kisiel-Dorohinicki, and Edward Nawarecki

Institute of Computer Science
AGH University of Science and Technology, Kraków, Poland
{kozlak,doroh,nawar}@agh.edu.pl

Abstract. In this paper problems of planning in transportation systems based on Pickup and Delivery Problem with Time Windows (PDPTW) are discussed. The results of two variants of evolutionary algorithms illustrate the pros and cons of using different approaches, and their cooperation in hybrid island model indicates how they can help each other in achieving better solutions. This leads to the general idea of an agent-based cooperative system, in which many different techniques may be used simultaneously, exchanging the obtained solutions. Experimental study of such a system that uses evolutionary algorithms and tabu search concludes the work.

Keywords: transport planning and scheduling, Pickup and Delivery Problem with Time Windows, evolutionary algorithms, multi-agent systems.

1 Introduction

It is rather obvious that effective organisation of transportation systems allows companies to highly limit sustained costs and be more competitive on the market. This could hardly be achieved without adequate tools, which should support transport planning on the basis of acquired knowledge on available resources, incoming requests and road network structure. Critical situations analysis seems to be of vast importance for such planning. Yet, even though there is a wide selection of planning techniques, most of them assume a complete description of both resources and requests available a priori. Thus it is very difficult (or even impossible) to apply them to dynamic problems, and even more difficult, with unsure and incomplete knowledge.

The goal of the research partially reported in this paper is to create concepts and tools, that should manage planning in dynamic environments of multi-agent systems in the face of crisis, considering transportation systems as a particular case. Based on a general scheme of crises management in MAS, as well as preliminary results obtained in the field of transportation systems [7], several possible variants of planning-support techniques were already considered [2]. In this paper special attention is paid to evolutionary techniques as a tool for solving static transportation problems, moving to cooperative systems, which should be flexible enough to be used in dynamic environments.

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

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 687–696, 2007.

Still the paper does not touch dynamic problems, but rather shows how different techniques or different configurations of similar techniques can help one another, attaining better results than when used alone.

Section 2 introduces a particular transportation problem considered in the paper, that is Pickup and Delivery Problem with Time Windows (PDPTW). In section 3 there is a discussion of evolutionary algorithms dedicated to solving transportation problems, with special attention to the advantages and shortcomings of different approaches. Selected experimental results that illustrate the described algorithms are presented in the next section. Section 5 introduces the idea of a cooperative system that allows for exchanging of solutions between different algorithms solving transportation problems, and finally section 6 provides an experimental study of the system at work.

2 Research on Transportation Problems

Typical transportation problems are based on a set of requests being realised by a set of available vehicles. Vehicles are characterised by their capacity and speed, and requests by the required capacity of vehicles and a time period (known as *time window*) within which the pickup and delivery operations have to take place. In a more widely researched *Vehicle Routing Problem with Time Windows* (VRPTW) with each transport request only one location point (either pickup or delivery) is associated, but in *Pickup and Delivery Problem with Time Windows* (PDPTW) each request is characterised by both a pickup and delivery location. The quality of a solution depends on the number of vehicles used and the total distance travelled. Sometimes, to express the quality of a solution, a total travel time of vehicles and a total waiting time of vehicles before the start of any time window is also considered. In problems *with hard time windows*, it is absolutely necessary that pickup and delivery operations start in the given time window. In problems *with soft time windows*, pickup and delivery operations may start after the end of this time period, but in estimating the quality of a solution, a penalty for the delay may be taken into account (higher delay may result in higher penalties). The problems have numerous practical applications — for example in planning sea and air transport, different kinds of cargo services and transport services on demand (for example transport of handicapped for treatment) or taxi-share services.

A set of benchmark tests for VRPTW was proposed by Solomon and extended by Gehring and Homberger. Li and Lim proposed a similar set of benchmark problems to verify the quality of the algorithms for PDPTW [9]. Benchmarks are divided into different groups depending on the number of requests to be served and locations to be visited (about 100, 200, 400 locations etc.). For each group six classes of tests are distinguished: on one hand due to the characteristics of time windows (problems with small time windows and a short scheduling horizon — LR1, LC1, LRC1, as well as with large time windows and a long planning horizon — LC2, LR2, LRC2), on the other due to the spatial distribution of requests (request locations may be grouped into clusters — LC1, LC2, evenly distributed — LR1, LR2, and there are also mixed problems with some request locations in clusters and some randomly distributed — LRC1, LRC2).

Due to the complexity of the described transportation problems (mainly on account of many constraints) nowadays the most promising approximate solutions provide heuristic approaches (accurate solutions are not attainable because of NP-hardness of the problem). The majority of algorithms are based on the generation of an initial solution using some simple heuristics (like insertion heuristic, sweep heuristic or partition heuristic), which is optimised afterwards using some metaheuristics. The proposed algorithms for VRPTW are numerous and it would be difficult to list them here. However it is worth mentioning that when comparing different VRPTW solving algorithms [1], hybrid evolutionary approaches achieve the best results. The approach based on the tabu search and simulated annealing [5] provided the best solutions obtained so far for PDPTW. Many other interesting approaches to PDPTW are also based on tabu search, e.g. [6,4].

3 Evolutionary Approach to Transportation Problems

Evolutionary algorithms are based on iterative transformation of the *population of individuals* potential solutions of the given problem. Evolution consists on generating consecutive generations, using so called *genetic operators* (or *variation operators*) and the *selection* mechanism.

Most evolutionary algorithms for transportation problems use direct representation of solutions [1] – each individual consists of consecutive locations assigned to particular routes. Such representation assumes no *coding*, which results in genetic operators operating directly on solutions. This guarantees the generation of acceptable solutions, which is easily achieved introducing genetic operators based on existing optimisation algorithms dedicated to transportation problems (e.g. pointed out in the previous section). Also the initial population is not generated randomly, as for typical evolutionary algorithms, but by using some existing construction heuristics. Several criteria considered for transportation problems are often aggregated (e.g. as a weighted sum) into a single value, which may be used as the fitness of individuals.

The discussed approach [3] is based on GENEROUS algorithm, which uses direct representation as described above. Two recombination operators: based on sequence (SBX) and route exchange (RBX), allow an improvement of the total distance, yet can hardly reduce the number of vehicles. Thus two mutation operators: one level (1M) and two level (2M) exchange, aim at emptying (the shortest) routes. Third mutation operator works as a local optimiser based on *or-opt* technique. If the solution cannot be repaired (there are unserved locations), it is rejected and the whole process is repeated [8].

This algorithm was adapted for the PDPTW problem, leaving the same representation, as well as slightly modified SBX recombination and 1M mutation operators. Also, additional recombination operator for exchanging best routes and mutation operators based on the concept of ghost routes were introduced. The initial population is generated using a clustering technique in the first phase and a modified sweep heuristic [5] to fill in the population up to the assumed size in the second phase. Tournament selection was used with individual comparison based on three criteria: the number of routes, a total distance, and a total waiting time, considered one by one in the given order.

In general the process of evolution should tend to generate better individuals and finally to find the needed (usually approximate) problem solution, which quality depends on the operators used and the parameters of the algorithm. Yet, evolutionary computation often suffers from the loss of population diversity, which practically hinders further search. This means that the algorithm locates the basin of attraction of some local optimum instead of a global one. This is especially important considering transportation problems with direct representation, because of introduced constraints, which often eliminate many new individuals from the population.

That is why a second considered variant of evolutionary algorithm utilized a partial representation of the solution, which consisted of only pickup locations. For such representation genetic operators for travelling salesman problem might be used. Also the initial population could be generated randomly. An *insertion heuristics* [5] allowed for transformation of every individual into a feasible complete solution. It was chosen because of its low computational complexity (it must be used for every individual in every generation), yet unfortunately permitted different individuals to be transformed into the same, often weak solution. Selection and fitness evaluation was realised in the same way as for the previous algorithm.

4 Experimental Comparative Study of the Evolutionary Approaches

Various experimental studies were conducted in order to compare the performance of the above-described algorithms [3]. Below, only selected results allowing for drawing preliminary conclusions are presented. The results were obtained for 100-location problems with even distribution of request locations, with small (LR1) and large (LR2) time windows. Tables 1 and 2 show the benchmark results [9] and the best individual obtained averaged over 3 independent runs of each algorithm, with the population of 125 individuals evolving for 125 generations.

Table 1. Results obtained for problems with small time windows (LR1)

	benchmark		GEN1		GEN2		GEN1+GEN2	
problem	routes	distance	routes	distance	routes	distance	routes	distance
lr101	19	1650.8	19	1744.5	19	1650.8	19	1650.8
lr102	17	1487.6	17	1580.9	17	1575.1	17	1523.9
lr103	13	1292.7	14	1550.1	13	1421.2	13	1369.8
lr104	9	1013.4	10.7	1149.9	11	1244.7	9.5	1037.2
TOTAL	58	5444.4	60.7	6025.4	60	5891.7	58.5	5581.8

For problems with small time windows (table 1) the results were quite good – both algorithms could find solutions very close to the benchmark ones (but one must remember that one vehicle more in the obtained solution gives a considerable relative difference to the benchmark value). The situation was slightly different for large time windows (table 2) – even though the number of vehicles obtained by both algorithms was still comparable to the benchmark result, the total distance was worse for the algorithm with direct representation (GEN1), and much worse for the algorithm with partial representation (GEN2).

Table 2. Results obtained for problems with large time windows (LR2)

	benchmark		GEN1		GEN2		GEN1+GEN2	
problem	routes	distance	routes	distance	routes	distance	routes	distance
lr201	4	1253.2	4	1419.1	4	1923.5	4	1328.4
lr202	3	1197.7	4	1398.9	4	1734.9	4	1341.2
lr203	3	949.4	3	1224.4	3	1849.6	3	1115.3
lr204	2	849.1	3	1099.7	3	1494.1	3	1110.9
TOTAL	12	4249.4	14	5142.1	14	7002.1	14	4895.9

The reasons for the weak results obtained seem to be different for the algorithms discussed. As already suggested and illustrated by figures 1, the first algorithm suffered from the lack of diversity in the evolving population, which inhibited its search capabilities from ca. 40-50 generation. One may notice that the second algorithm maintained the diversity for the whole run. The reason for weak results in this case was the heuristics used to generate complete solutions, as suggested in the previous section.

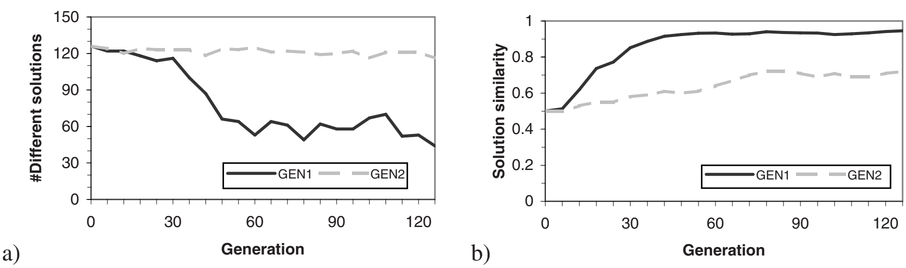

Fig. 1. The number of different solutions (a) and the similarity of solutions (b) in algorithms GEN1 and GEN2

The obvious conclusion drawn from these experiments was to use both algorithms simultaneously, allowing to exchange the solutions during the search. This meets the idea of the hybrid island model of parallel evolutionary algorithm, assuming that migration operator is responsible for conversion of the solutions between representations used by both algorithms. The results presented in the third part of tables 1 and 2 are quite promising and initially confirm the correctness of the approach.

5 Solution at a Cooperative Level

As it was illustrated in the previous section different algorithms applied to transportation problems have different strengths and weaknesses, e.g. some may be better suited to solving problems with small time windows and other for problems with large time windows. This is also confirmed by benchmark results – the best known solutions for a given test case are often obtained by different algorithms [9]. Preliminary results obtained for the discussed dual-population evolutionary algorithm indicated that cooperation of different approaches should allow to achieve more flexibility and produce better results for a variety of test cases.

That is why the environment was developed that facilitates the cooperation of different algorithms solving PDPTW (with both hard and soft time windows), by means of exchanging solutions or even parts of solutions (routes, requests served in the routes). The system model and architecture is based on a multi-agent approach and consist of several embedded sub-environments which contain computational agents [3]. There are two kinds of agent groups:

- standard groups – applying the algorithms proposed by [5] based on tabu search and simulated annealing,
- evolutionary groups - take advantage of algorithms presented above.

Of course the optimisation is performed simultaneously by different agents using different algorithms.

The quality function used has the following form:

$$f = \alpha N + \beta D + \gamma CD + \delta WT + \varepsilon P \tag{1}$$

where: NV – the number of vehicles, TD – total distance, SD – total service realisation time, WT – total waiting time, P – total lateness, α – weighing factor of the number of vehicles (in tests equals 5000000), β – weighing factor of the total distance (in tests equals 1000), γ – weighing factor of total service realisation time (in tests equals 1), δ – weighing factor of total waiting time (in tests equals 0.001), ε –weighing factor of penalty caused by lateness (in tests equals 100). The quality of the solution decreases with the increase of the value of the f function.

The important characteristic feature of the presented approach is a cooperative aspect of the computation process. Agents which represent different algorithms find the routes and requests having the worst influence on the quality of solution (have the highest impact on the quality function). Each agent is then informed by other agents about similar routes to their worst ones (identified by the central point calculated as the average of respective coordinates of request/delivery locations present in the given route), and about the routes, where the other agents placed the most costly requests that were analysed. On the basis of the obtained suggestions, the agent may modify its route or even construct a new one, selecting the request from the route being replaced or from other accessible routes and moving the other requests from the route being removed to other feasible positions in other routes.

6 Results of the Cooperative Approach

The goal of the tests performed was to compare the quality of solutions offered by the discussed cooperative algorithm with the quality offered by the considered metaheuristics used alone, as well as the quality of the best known solutions. Numerous tests were performed for different sets of Li-Lim benchmarks, but as the space in this paper is somewhat limited, only the most interesting results concerning the number of vehicles and total travel distance are presented.

To search a wide part of the solution space, different quality functions were applied in the particular agents. These differences are consequences of different weights (sometimes randomly generated) of particular elements of the quality function. Thanks to this approach, it was possible to take into consideration different kinds of solutions, for example the ones that attached greater significance to the number of vehicles used, a total distance or the arrival on time at service points. The difference between the results may also be influenced by the fact that the cooperative approach used soft time windows and thus allowed solutions with vehicles arriving late at service points, but their lateness was penalised by an important factor.

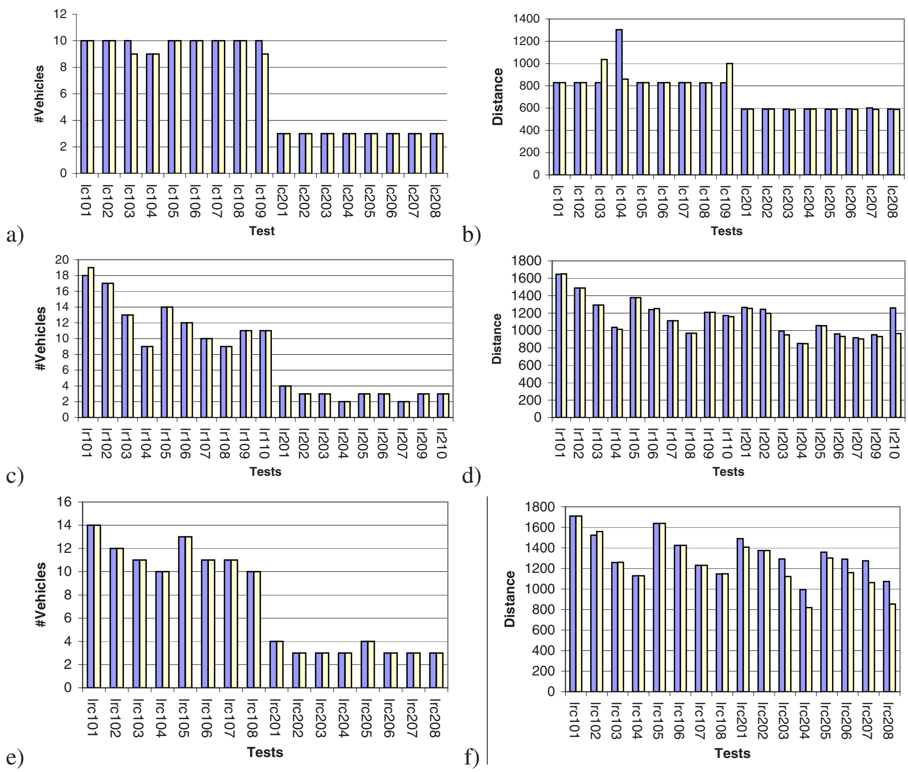

Fig. 2. Results for 100 locations: vehicles (a) and total distance (b) for LC1/LC2, vehicles (c) and total distance (d) for LR1/LR2 and vehicles (e) and total distance (f) for LRC1/LCR2; dark bars – results of our cooperative algorithm, fair bars – the best known solutions

The computational environment was composed of two groups of three agents of different types (tabu and evolutionary). If the basic algorithms were able to find the best known solution, the meta-algorithms were unable to find a better one, unless it accepted some lateness and penalty factor associated with it. In the situations when basic algorithms were not able to find the best known solutions, the meta-algorithm sometimes guaranteed an increase of solution quality. The best benefits of the introduced approach appeared in the problems with small time windows in the LR type problems. One can also notice that the worst results were obtained for LRC problems and for long time windows.

Figures 2 and 3 show the results obtained by the cooperative algorithm in comparison to the best known solutions for benchmark problems with 100 and 200 request locations. The figures include the results for cases with clusters, with even spatial distribution and mixed clusters/distributed. For each figure, the results for small and large time windows are presented. The number of used vehicles and a total travel distance for each group of tests are presented in separate figures.

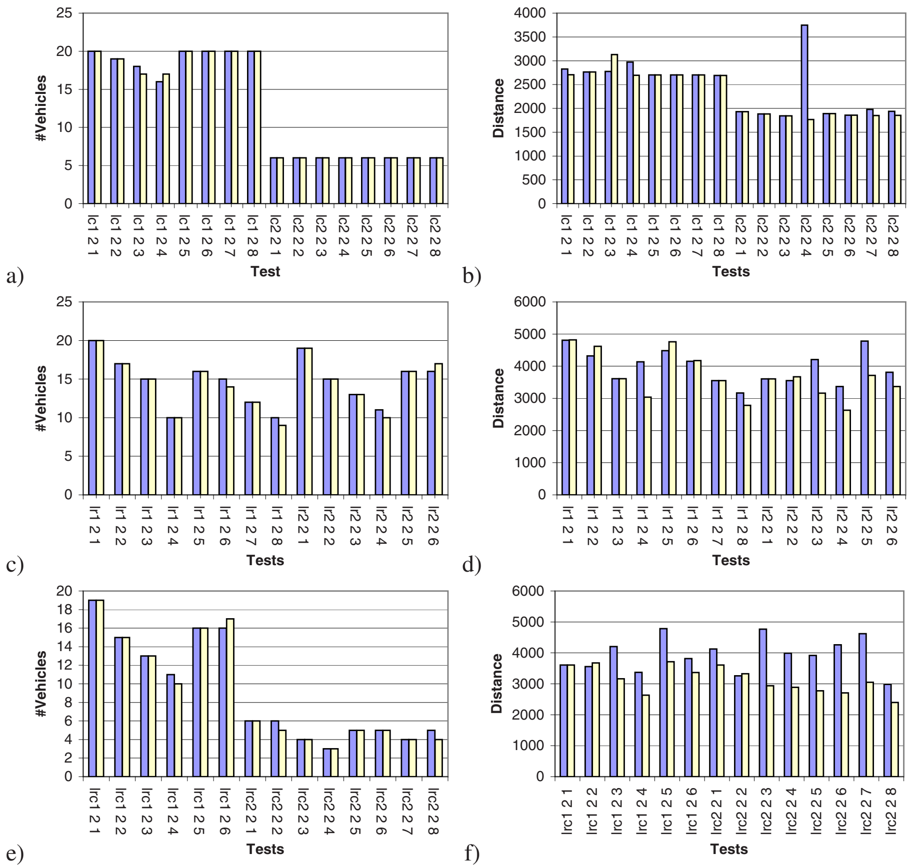

Fig. 3. Results for 200 locations: vehicles (a) and total distance (b) for LC1/LC2, vehicles (c) and total distance (d) for LR1/LR2 and vehicles (e) and total distance (f) for LRC1/LCR2; dark bars – results of the cooperative algorithm, fair bars – the best known solutions

In table 3 the results obtained for cooperative algorithm are compared with the results obtained using only evolutionary island model based on algorithms presented in the previous section and the multi-agent systems which uses only tabu-search heuristics. In the columns concerning the distances ”+” (”-”) means that cooperative algorithm obtained better (worse) results with respect the total travel distance in the given percentage of tests of the considered test class. In the columns concerning the number of routes additional information is provided. It concerns a difference between the number of routes used by the cooperative solution and basic solutions (”+” – less routes, ”-” – more routes).

Note that in the case of island model of evolutionary algorithm and multi-agent system consisting of tabu agents sometimes not all benchmark problems in the given class were solved. The table shows that mixing the solutions obtained from evolutionary and tabu algorithms using the cooperative algorithm in general gives better results.

Table 3. Results obtained for cooperative approach in comparison to evolutionary algorithm and agent-based tabu search

	cooperative/evolutionary		cooperative/agent-based tabu	
problem	routes	distance	routes	distance
100 LC1	(25%,-1)	(25%+),(25%-)		(11%+)
100 LC2		(50%+)		(25%+),(25%-)
100 LR1	(25%,+1)	(75%+),(25%-)	(10%,+1)	(60%+)
100 LR2	(25%,+1)	(100%+)		
100 LRC1	(25%,+1)	(100%+)	(12.5%,+1)	(75%+)
100 LRC2	(25%,+1)	(75%+), (25%-)		(37.5%+)
200 LC1	(25%,+1)	(75%+), (25%-)	(12.5,%+1)	(50%+), (12.5%-)
200 LC2		(50%+), (25%-)	(50%+)	(37.5%+), (12.5%-)
200 LR1	(25%,+2),(25%,+1)	(75%+)	(50%,+2)	(75%+), (25%-)
200 LR2	(25%,+1)	(50%+), (50%-)	—	—
200 LRC1	(25%,+2),(25%,+1)	(50%+),(50%-)	—	—
200 LRC2	(75%, +1)	(25%+),(75%-)	—	—

The final total results are as follows:

- 33% was equal to the best known solutions,
- 22% was better than the best known solution, after the application of soft-time windows and calculation of penalties,
- 14% was worse than the best know solutions obtained so far, considering the number of vehicles,
- 36% of results were worse then the best known solutions considering the total travel distance.

7 Concluding Remarks

In this paper two different approaches of growing complexity for solving transportation problems were presented. The results obtained using both systems do not differ significantly from the best known solutions for the existing set of benchmark problems. The cooperative approach not only allows to get slightly better results but also proves much more flexible. This is due to the use of soft time windows, because often it is not possible to strictly predict times of the particular activities (like travel time, pickup time, delivery time) or the definition of changes in the problem (due to breaking down of the cars or withdrawing of requests). In relation to this, considering the possibility of development of plans based on different definitions of quality function may make it possible to find solutions which are more resistant to critical situations. It also makes it possible to develop a set of plans which afterwards may be adapted to the current conditions with respect to new or unpredicted events arising. Additionally, it may constitute a basis for the development of solutions for dynamic problems (when new requests arrive simultaneously while the vehicles are serving the previously accepted requests), which will be the subject of further research.

References

1. Braysy, O.: Genetic algorithms for the vehicle routing problem with time windows. Arpakannus. Special issue on Bioinformatics and Genetic Algorithms 1, 33–38 (2001)
2. Dreżewski, R., Kisiel-Dorohinicki, M., Koźlak, J.: Agent-based and evolutionary planning techniques supporting crises management in transportation systems. In: Dolgui, A., Morel, G., Pereira, C.E. (eds.) Information Control Problems in Manufacturing 2006. Information Systems, Control and Interoperability, vol. 1, Elsevier, Amsterdam (2006)
3. Lalewicz, M., Wójcik, J., Rola, L., Srebro, M., Koźlak, J., Kisiel-Dorohinicki, M.: Evolutionary and hybrid approaches for PDPTW (in polish: Ewolucyjne i hybrydowe podejścia do PDPTW). Technical Report 5/2005, Department of Computer Science, AGH-UST, Kraków (2005)
4. Lau, H., Liang, Z.: Pickup and delivery with time windows : Algorithms and test case generation. In: Proceeedings of 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01), Dallas, USA (May 2001)
5. Li, H., Lim, A.: A metaheuristic for the pickup and delivery problem with time windows. In: Proceedings of 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01), Dallas, USA (2001)
6. Nanry, W., Barnes, J.: Solving the pickup and delivery problem with time windows using reactive tabu search. Transportation Research 34(Part B), 107–121 (2000)
7. Nawarecki, E., Koźlak, J., Dobrowolski, G., Kisiel-Dorohinicki, M.: Discovery of crises via agent-based simulation of a transportation system. In: Pěchouček, M., Petta, P., Varga, L.Z. (eds.) Multi-Agent Systems and Applications IV. LNCS (LNAI), vol. 3690, pp. 132–141. Springer, Heidelberg (2005)
8. Potvin, J.Y., Bengio, S.: The vehicle routing problem with time windows - part ii: Genetic search. INFORMS Journal on Computing 8, 165–172 (1996)
9. Benchmarks - vehicle routing and travelling salesperson problems. `http://www.sintef.no/static/am/opti/projects/top/`