Multi-agent Models for Transportation Problems with Different Strategies of Environment Information Propagation

Jarosław Koźlak, Sebastian Pisarski, and Małgorzata Żabińska

AGH University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
{kozlak,zabinska}@agh.edu.pl, pisarski.sebastian@gmail.com

Abstract. The aim of the work is to show a solution of dynamic transportation problem Pickup and Delivery Problem with Soft Time Windows, taking into account variable travel times between locations. The information about changes is obtained thanks to an algorithm that generates and propagates traffic jams or collects data from the traffic simulator. The model of such a multi-agent system is presented. Different types of propagation of information about changes of travel times for agents constituting a fleet of vehicles for the realisation of transportation requests and different methods of prediction of future actual travel times on road sections are taken into consideration.

Keywords: Transportation Problems, Dynamic Environment, Information Propagation, Prediction.

1 Introduction

A substantial amount of work on agent systems is targeted on building these systems to function properly in the dynamic environment, while adapting themselves to changing situations. The most important problem emerging in such systems is: to gain information about changes as well as accuracy and currentness of the information and quality of the models created to anticipate future changes. Many models of the multi-agent systems were developed considering these aspects (for example [5]), but in our work we are aiming to analyse a transportation problem from this point of view, such a kind of evaluation has not yet been performed according to our knowledge. We are focusing on Pickup and Delivery Problem with Soft Time Windows (PDPSTW) with changing times of travel caused by propagating traffic disturbances. This transport problem belongs to NP-hard problems. PDPTW consists in serving a set of transport requests by a fleet of available vehicles. Particular requests are described by points of pick up and delivery, time windows, when pick up and delivery should occur and required capacity (load). The function of the solution quality is dependent on solution parameters as: number of vehicles, the total distance, the total time

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 145–156, 2013.

of travel, times of waiting (stops) and the penalties, in cases of delay, so the solution of the problem may be considered as multi-criterial. Actions of all agents have influence on the final obtained result. The agents express their cooperative nature by solving imposed tasks and by exploring the environment. It is possible to evaluate the algorithms grading the scale of disturbances. There is also a capability to control the threshold of acceptable level of penalty from the side of each agent, by defining maximal acceptable delays in relation to time or total distance of travel.

Since these solutions may be mapped on different types of problems, e.g. related to tasks scheduling, the obtained conclusions may also be referred to other types of computational problems [4].

In this paper, we extended our solutions worked out so far, with elements related to the propagation of information about changes of environment and implementing appropriate methods of disturbances propagation. In [14] the influence of closure of particular roads on the quality of obtained results was examined, whereas in [8] - the influence of other crisis situations, as well. The problems connected with considering information and reaction on disturbances were examined in [9], where we examined the influence of probabilistic data concerning the spatial distribution of requests on modifications on vehicle routes upon the final solutions gained.

2 Research Domain Overview

Transportation Problems and Heuristic Methods of Their Solving. Transportation problems of the type vehicle routing problem, describing transfer of goods between locations, are widely examined with the use of different heuristic algorithms. In our work we will especially concentrate on the problem with time windows, i.e. Pickup and Delivery Problem with Time Windows (PDPTW), as well as on its special case – Vehicle Routing Problem with Time Windows (VRPTW). For the static PDPTW all requests are known at the moment of the vehicle journey start. The set of test benchmarks was proposed to evaluate the quality of solution of such problems ([13,1]). In the work we particularly concentrate on the dynamic transportation problems, where a request comes during vehicles movement/travel. A review of heuristic solutions of the problem is available in [15].

Multi–agent Models for Solving Transportation Problems. Many multi-agent systems for transport planning have come into existence. One of the best known, and at the same time presenting solutions, which we also refer to in our work, is MARS System [6]. Similarly to meta-heuristic solutions which comprise two phases – constructive heuristics and optimisation heuristics, also in this approach two steps are used: contract-net protocol [16], for initial assigning requests to vehicles and "simulated trading" algorithm [3] for later optimisation of the solution, allowing agent to accept or to get rid of requests which realisation becomes costly for them, chains of such exchanges with the participation of many agents may be constructed in this process.

Perception of Environment Dynamics, Information Propagation and Value. Obtaining and interpreting of information gained from the environment is a substantial domain in examining multi-agent approach and distributed artificial intelligence from the beginning of 1980's. In [12] a concept of the environment consisting of distributed sensor network monitoring of moving vehicles is presented. The problems of application of multi-agent systems for creating distributed networks of sensors was then widely examined, the review of works can be found in [11,17]. Optimal frequency of information exchange, in reference to the policy of performed actions in cooperative multi-agent system was examined in [7].

3 System Model

3.1 Main Model Elements

Our model comprises a dynamic environment and a set of agents of two types. The environment is described by a graph representing a transport network (the chosen essential locations and crossroads, as well as road sections between them) with the current times of travel by particular road sections with information about currently active (reported but not loaded yet) transportation requests, available for loading at a given point. Dynamics of an environment on the one hand - consists in changes of the set of active requests – disappearing loaded requests and the new incoming ones – but on the other hand - in generated changes of times of travel by road sections.

The Dispatcher is responsible for collecting transportation requests and assigning them to Transportation Units, identification of the general situation and adjustment of general working schema to the current situation. Transportation Units representing vehicles, are responsible for request realisation and in some scenarios, also for collecting information about the current environment state. Agents of the level of the model in the system are described by indexes $j\colon j \geq 0 \wedge j \leq n$, where agent $j : j = 0$ is a Dispatcher agent, but agents $j\colon j > 0$ are Transportation Units.

3.2 Model of Dynamic Environment

The environment model, used by the Dispatcher and Transportation Units, includes information about the structure of the transportation network, current, historical and predicted travel times across the road sections, and data about transportation requests, not assigned or picked up, in the system.

Therefore, a dynamic environment in time step t may be described as follows:

$$DynEnv(t) = (\mathbb{G}(t), ReqsA(t), ReqsP(t)) \quad (1)$$

where:

- $\mathbb{G}(t)$ – total state of the transportation network in the time step (t), contains not only the current state in the time step t – $Gr(t)$, but also information

about the previous states of the network, used for prediction, and future predicted states of network, marked $Gr(k)_{(t)}$, where in the case of historical data $k = t - h_{max}, t - h_{max} + 1, \dots t - 1$, where h_{max} – assumed size of history, and in the case of predicted future changes of network state, $k = t + 1, t + 2, \dots t + p_{max}$ where p_{max} – the maximum range of the predicted steps,
- $ReqsA(t)$ – a set of active requests, which arrived but still are not picked up by transportation units,
- $ReqsP(t)$ – set of requests picked up by vehicles, which still need to be delivered to destination locations,

The given agents j may have only partial and not up to date knowledge, regarding the environment, $DynEnv^j(t)$, in practice it may not include up to date knowledge about travel times across the road sections, incomplete historical information or disturbed prediction results.

Changes of the environment include:

- new request coming: a new request i comes in the time step t: $ReqsA'(t) := ReqsA(t) \cup Reqs_i$, $Reqs_i$ –new request coming in time step t, $ReqsA(t)$ – old set of active requests, $ReqsA'(t)$ – new modified set of active requests,
- picked up request $Reqs_i$ in time step t, a request ceased to be active and is loaded to the vehicle: $ReqsA'(t) := ReqsA(t) \setminus Reqs_i, ReqsP'(t) := ReqsP(t) \cup Reqs_i$
- delivery of request $Reqs_i$ in time step t, a request ceases to be loaded and is not considered anymore in the current state of the system: $ReqsP'(t) := ReqsP(t) \setminus Reqs_i$
- changes of travel time through the road sections: $Gr'(t) = Change(Gr(t))$

3.3 Fundamental Model

Dispatcher Dispatcher (DA). Agent Dispatcher is responsible for sending requests to the vehicles. The representation of dispatcher in time stamp t is expressed by equation 2.

$$DA(t) = (G^0(t), S^0(t), K^0(t), A^0) \tag{2}$$

where:

$G^0(t)$ – a value of goal function for time step t, which depends on the number of vehicles, total travel distance and total penalty for delays,

$S^0(t)$ – state of dispatcher, $S_0 = \{LocAgent(t), ReqsP(t), ReqsA(t)\}$, where $LocAgent(t)$ – set of locations of all transportation unit agents j, defined as follows: $LocAgent_j(t) = \{LocAgent_j(t) \colon j = 1 \dots m\}$, – estimated location of all agents j, m – number of transportation units, $ReqsP(t)$ – set of sets of requests assigned to and picked up to given transportation units j, $ReqsP^{(t)} = \{ReqsP^j(t) \colon j \in [1, m]\}$, $ReqsA(t)$ – set of sets of requests assigned to given transportation units j, but not picked up yet, $ReqsA(t) = \{ReqsA^j(t) \colon j \in [1, m]\}$:

$K^0(t)$ – knowledge of dispatcher, $K^0(t) = \{Env^0(t)\}$, where $Env^0(t)$ - information about transport network in time t, $Env^0(t) = (Gr(t), HGR^{(t)}(LpU))$, $Gr^{(t)}$ – state of transport network in time t and $HGR^{(t)}(LpU)$ – history of previous states of transport network during last LpU changes, $HGR^{(t)}(LpU) = \{Gr^{(k)} : k \geq t - LpU \wedge k \leq t \wedge k \in \mathbf{N}\}$

A^0 - allowed actions of dispatcher.

The allowed actions of dispatcher are as follows:

$$A^0 = \{Allc^0, GtPos^0, SndPos^0, PU^0\} \tag{3}$$

where:

$Allc^0$ – auction of a request and its allocation to a Transportation Unit,

$$Allc^0 : (Rec_i, ReqsV(t), Reqs(t), t) \rightarrow (ReqsV(t) = ReqsV(t) \backslash Rec_i, \\ ReqsA^j(t) = ReqsA^j(t) \cup Rec_i)$$

where j – agent which got a request for realisation, $ReqsV(t)$ – set of not assigned requests, $ReqsA^j(t)$ – set of all requests assigned to given transportation units j

$GtPos^0$ – get info about positions of transportation units, $GtPos^0 : (LocAgent(t)) \rightarrow (LocAgent(t) = \bigcup_{0<j\leq m} Loc^j)$

$SndPos^0$ – send info about agent positions to transportation units,

PU^0 – operation of prediction (described in more detail in section 3.5).

Transportation Unit. Transportation unit TU^j is expressed as follows:

$$TU^j = (G^j(t), P^j(t), K^j(t), S^j(t), A^j), \tag{4}$$

$G^j(t)$ – the goal function, which depends on total travel distance and total penalty,

$P^j(t)$ – a plan represented as a sequence of operations of movements, pickup and delivery operations, acquiring information about change of travel time for given road section,

$K^j(t)$ - knowledge of $j-th$ transportation unit $K^j = \{Reqs^j(t), DynEnv^j(t)\}$ comprising $Reqs^j$ - set of requests, $DynEnv^j(t)$ - information about dynamic environment (see 3.2). environment (transport network).

$S^j(t)$ – state of the j-th transportation unit, $S^j = \{Loc^j, ReqsA^j, ReqsP^j\}$, where Loc^j, – location, $ReqsA^j$ – allocated requests, $ReqsP^j$ – picked-up requests,

A^j - allowed actions of j-th transportation unit.

Allowed actions of the transportation unit j are as follows:

$$A^j = \{Allc^j, Mv^j, Ld^j, Dlv^j, Inf^j, PU^j\} \tag{5}$$

$Allc^j$ - participation in auctions of transport requests
$Allc^j : (ReqsA^j(t), Reqs_i, t) \rightarrow (ReqsA^j(t) = ReqsA^j(t) \cup Reqs_i)$
Mv^j – transfer between locations,
$Mv^j(N_p, N_q) : (Loc^j = N_p) \rightarrow (Loc^j = N_q)$
Ld^j - loading,
$Ld^j : (ReqsA^j(t), ReqsP^j(t), Reqs_i, t) \rightarrow (ReqsP^j(t) = ReqsP^j(t) \cup Reqs_i,$
$ReqsP^j(t) = ReqsP^j(t) \backslash Reqs_i)$
Dlv^j - unloading,
$Dlv^j : (ReqsP^j(t), Reqs_i, t) \rightarrow (ReqsP^j(t) = ReqsP^j(t) \backslash Reqs_i)$
Inf^j : – gather information about travel times for current road section, in time t (q – current edge/road section),

$Inf^j : (T_{E_q^j(t)}, q, t) \rightarrow Env^j = Env^j \cup (T_{E_q^j(t)})$
PU^j – operation of prediction, which produces the predicted values of travel times for road sections for next p_{max} steps, $PU^j : (Env^j) \rightarrow (Env^j)'$

3.4 Model of Information about Propagation of Environment Changes

In the realised environment, there are several methods of propagation of information about disturbances taking place or being noticed, this information reaches Dispatcher and Transportation Units. The model of the information propagation about dynamic environment is expressed as follows $DynEnvInfo(t) = (DynEnv(t), p)$, where $DynEnv(t)$ is the model of dynamic environment described in section 3.2 and p is an assumed mode of information propagation, which determines the method, how agents become acquainted with changes of travel times associated with given road sections.

The following modes p of information propagation are currently considered:

- *immediately* – (i) – dispatcher and all transportation units obtain globally immediate information about every change of travel times in any of the road sections,
- *after noticing* – (an) the information about change is disseminated to all agents only when a transportation unit discovers it (it means drives the road section with changed travel times),
- *after time* – (pX) – changes are disseminated to all agents after the given time period X, if a transportation unit notices the change earlier, it takes it into consideration in its plan, but does not inform other agents.

3.5 Models of Prediction of Environment Changes

The Dispatcher and Transportation Units may either not predict the future state of the environment, considering current and historical information, or predict them using one of several prediction methods. If prediction algorithms are used, the considered size of historical changes may influence the results. Predicting agents are equipped with a prediction unit PU which is represented as follows: $PU = PU(L_{PU}, A_{PU})$, where

- L_{PU} – number of remembered last changes of the road graph,
- A_{PU} – a prediction algorithms, the following solutions are implemented: *Standard* (lack of prediction), *Average* (basic mean value), *MovingAverage* (exponential moving average).

4 Realisation

The main elements of the system are presented in fig. 1.

Simulation control (Test Agent) – a component responsible for reading input files and generating the output ones. Its task is a simulation initialisation and sending simulation progress time markers. It also performs a function of the "outer world" from the point of view of other agents, it transfers incoming requests, as well as takes care of propagation of changes in the road network graph.

Data about agents in the system (InfoAgent) – the agent, which deals with creation (with the use of JADE [2] platform tools) of component agents of Transportation Units. It also stores identifiers of created agents.

Distribution of requests (Dispatcher) – it receives incoming requests from the control unit. Then they are queued in the optimal sequence (different policies) in such a way to increase probability of optimal routes creation after assigning requests to Transportation Units. Each request is sent to the existing Transportation Units with a query about costs of realisation. The Unit offering the lowest cost is chosen. When the request cannot be realised using the current set of vehicles, it initiates the process of negotiation (level 2 optimisation process), based on our implementation of a Simulated Trading [3] algorithm. It gathers schedules of all Transportation Units, modifies them, and sends them back - updated. If it is still not possible to serve the request, it orders the creation of a new Transportation Unit.

Transportation Units – they serve requests assigned by the Dispatcher. After receiving the new request they calculate the cost of its realisation and send it back to the Dispatcher. Additionally Transportation Unit transfers information on its state to the simulation control module, and to the Dispatcher. Apart from this, it has also the capability to change its route construction strategy during the course of simulation. After allocating new requests, the Transportation Units start an optimisation process (level 1), they offer a given number of requests allocated to them, that mostly increase the cost of their route, to other Transportation Units. If there is another agent which can serve a request with lower costs than its current owner, the request is moved to its route.

In the system, two types of disturbance generation were realised.

Simple generator of disturbances. It is based on the concept of virtual vehicles, which move along graph edges, carrying disturbances by themselves. The disturbance generator functions in the following way:

- initialisation – parameters of each vehicle are drawn from available values set during the initialisation of the generator. They are: departure time, worsening of traffic factor, the path to be travelled by the given vehicle,

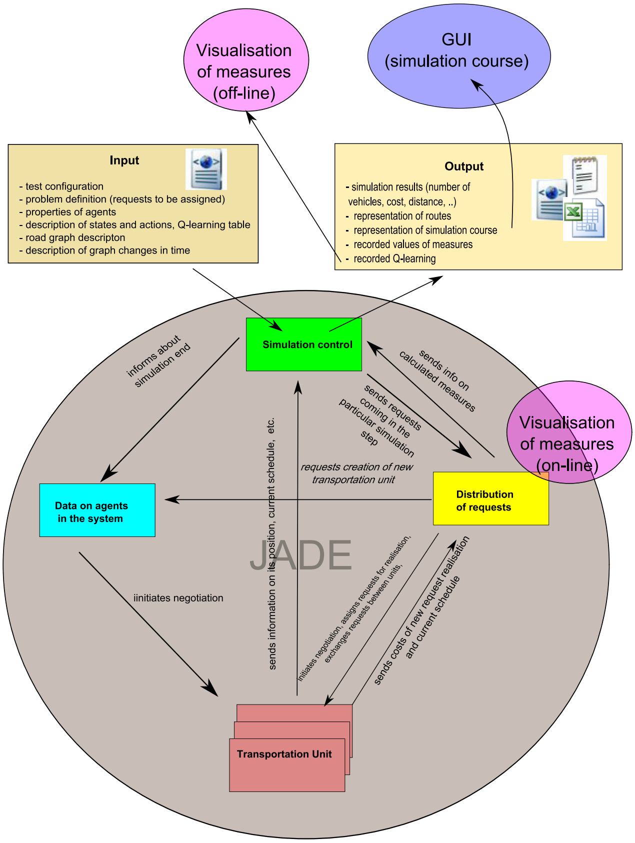


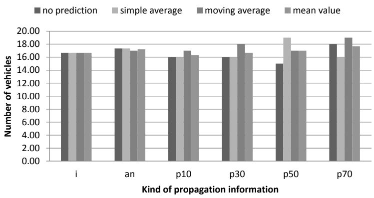
Fig. 1. General architecture of the system

– vehicles are moved in the loop until the deadline iterations, when the vehicle enters the given road section, it modifies the time of travel using its worsening factor, the change remains until the vehicle leaves the road section,
– the state of the graph with current travel times through the road sections is recorded every given number of time steps.

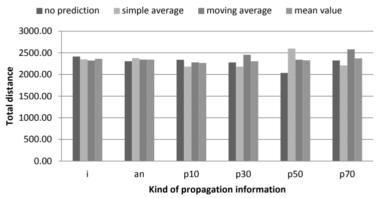
Use of Configuration Data Generated by Multi-agent Traffic Simulator. The second method of gathering information on the changes of traffic states and changes of travel times through road sections are obtained from the multi–agent traffic simulator, described in [10]. It uses extended multi-lane Nagel-Schreckenberg traffic model, and adaptive algorithm for traffic lights on intersections.

5 Results

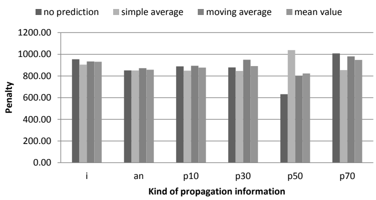
Experiments were carried out for different types of graphs as well as different grades of disturbances. For such defined conditions tests were carried out with the use of different types of information on propagation and different prediction algorithms. One group of tests (*tests1*) was performed with disturbances generated by virtual vehicles, the second for travel times generated using traffic modelling system (*tests2*).



(a) Vehicles



(b) Total distance



(c) Total penalty

Fig. 2. Results for high disturbances. Different methods of propagation information about changes and different prediction algorithms.

Table 1. Average values for different strength of traffic disturbances

disturbance strength	vehicles	distance	penalty	non-performed requests
veh100	16.94	2351.95	888.19	0.00
veh200	19.28	2770.52	1014.67	0.00
veh300	19.17	2732.52	892.58	3.00
veh400	17.25	2499.25	1034.25	7.58
veh500	18.58	2354.41	974.83	11.00

For *tests1* average values of vehicle numbers were shown, as well as the total distance and penalties gained for several simulation runs, for each considered methods of information propagation and predictions (fig. 2).

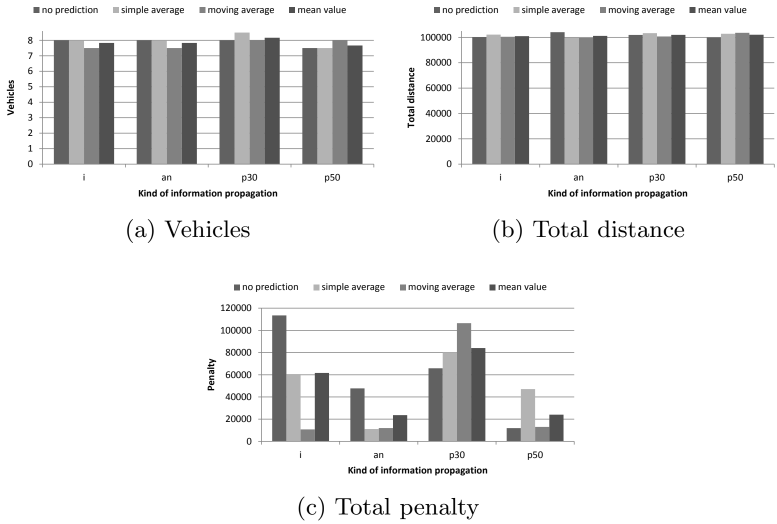


Fig. 3. Results obtained for different information propagation and different prediction algorithms using data from road traffic simulation

The following methods of information propagation about changes of travel times were used: immediate (*i*), after noticing by vehicle which entered the given edge (*an*), or with delay of 10, 30, 50 and 70 time units (*pXX*). For each method of information propagation the results are given for the following situations: *no prediction*, prediction using the following *simple average* and *moving average* methods, and *mean value* obtained using all methods.

One can see that an average number of used vehicles usually slightly increases with the decrease of information accuracy. One of the prediction methods, especially *simple average* in most cases enables obtaining better results than in case without prediction. Thus we can say, that although the results are to some extent burdened by random generation of disturbances, usually more accurate information slightly favours better solutions.

Analysing obtained results from the point of view of two criteria viz. (i) solution quality or vehicles number, then the distance and (ii) penalty from delays, one can distinguish non-dominated solutions making Pareto front. They are: simple average and moving average for immediate information flow, simple average for information obtained after 10, 30 and 70 time units, no prediction for information obtained after 50 time units, all three predicting methods for the information obtained after the vehicle visits of the edge. So, the use of methods of predicting changes, even despite their simplicity (lack of consideration e.g. identification of patterns recognised in history) usually allows us to obtain better solutions than in cases without prediction.

The important question is also how the increase of the level of the traffic disturbances influence the average results obtained by agents (tab. 1). We considered the disturbances generated by 100, 200, 300, 400 and 500 virtual vehicles. For each level of disturbances average values were calculated, considering the problems with all kinds of propagation of information about changes and three considered prediction strategies. For disturbances generated by 100 and 200 vehicles all requests are performed, for 200 vehicles the number of vehicles and total distance increase. For cases with 300, 400 and 500 vehicles appear and increase requests which are not possible to be served in a given time horizon.

For the *tests 2* one can notice that differences for different methods of information propagation are small. It is caused by more realistic and steady changes of the traffic characteristic. Considering the non-dominated solutions for each kind of information propagation, it is a moving average for immediate information propagation, moving and simple average for information obtained after vehicle visits the edge, moving average and no prediction for information obtained after 30 time units and no prediction for obtained after 50 time units. It is worth noting that for the information obtained after much longer time period – 100 time units – the number of vehicles for no prediction increased to 9, while for the both prediction methods it remained at the level of 8 vehicles. Also the total distance for no prediction increased to 112 thousands, remaining at about 100 thousands and 103 thousands for simple and moving average respectively. The penalty for no prediction case increased to a very large value, more than 300 thousands, while for the prediction cases, it remained at the usual low level.

6 Conclusions

Within frames of our works the model and environment to solve dynamic transport problems with changes of travel times were developed and different methods of propagation of information about changes and travel time prediction, for different strengths of disturbances were evaluated.

Analysing the results for more natural evolution of traffic conditions (*test2*) one can notice, that for the majority of cases the impact of the choice of prediction methods and information propagation methods was weak, but for very accurate and very delayed information results obtained without prediction were significantly worse. It seems that for up-to-date information about traffic conditions, prediction may help to obtain better results allowing to fit the plans better to the future real situation. For very inaccurate information, prediction often allowed reduction of differences between the real traffic conditions and the conditions considered during planning, while for the no prediction technique this gap remained very large.

For stronger changes of traffic conditions (*test1*), the results (number of vehicles and distance) are usually better for cases with prediction, especially when the information is up-to-date (i, p10, p30). When the level of disturbances exceeds a given threshold (for the tests it was 300 virtual vehicles causing increases in travel times) it is no longer possible to handle all requests, the number of non-served requests increases together with the further increase of disturbances.

Further work will comprise extension of the set of available optimisation algorithms and their configuration. Another direction of work will be applying more complex regression methods for predicting traffic conditions, such as clustering and regression trees techniques.

References

1. Benchmarks - Vehicle Routing and Travelling Salesperson Problems, http://www.sintef.no/Projectweb/TOP/Problems/PDPTW/Li--Lim-benchmark/
2. Java Agent DEvelopment Framework, http://jade.tilab.com/
3. Bachem, A., Hochstättler, W., Malich, M.: The simulated trading heuristic for solving vehicle routing problems. Discrete Appl. Math. 65, 47–72 (1996)
4. Beck, J.C., Prosser, P., Selensky, E.: Vehicle routing and job shop scheduling: What's the difference? In: Proc. of the 13th International Conference on Automated Planning and Scheduling, pp. 267–276 (2003)
5. Cetnarowicz, K., Gruer, P., Hilaire, V., Koukam, A.: A formal specification of M-agent architecture. In: Dunin-Keplicz, B., Nawarecki, E. (eds.) CEEMAS 2001. LNCS (LNAI), vol. 2296, pp. 62–72. Springer, Heidelberg (2002)
6. Fischer, K., Muller, J., Pischel, M.: Cooperative Transportation Scheduling: an Application Domain for DAI. Applied Artificial Intelligence, 1–33 (1996)
7. Goldman, C.V., Zilberstein, S.: Optimizing information exchange in cooperative multi-agent systems. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2003, pp. 137–144. ACM, New York (2003)
8. Konieczny, M., Koźlak, J., Żabińska, M.: Multi-agent crisis management in transport domain. In: Allen, G., Nabrzyski, J., Seidel, E., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2009, Part II. LNCS, vol. 5545, pp. 855–864. Springer, Heidelberg (2009)
9. Koźlak, J., Créput, J.C., Hilaire, V., Koukam, A.: Multi-agent approach to dynamic pick-up and delivery problem with uncertain knowledge about future transport demands. Fundamenta Informaticae: Annales Societatis Mathematicae Polonae 71(1) (2006)
10. Koźlak, J., Dobrowolski, G., Kisiel-Dorohinicki, M., Nawarecki, E.: Anti-crisis management of city traffic using agent-based approach. Journal of Universal Computer Science 14(14), 2359–2380 (2008)
11. Lesser, V., Tambe, M., Ortiz, C.L. (eds.): Distributed Sensor Networks: A Multiagent Perspective. Kluwer Academic Publishers, Norwell (2003)
12. Lesser, V., Corkill, D.: The Distributed Vehicle Monitoring Testbed: A Tool for Investigating Distributed Problem Solving Networks. AI Magazine 4(3), 15–33 (1983)
13. Li, H., Lim, A.: A Metaheuristic for the Pickup and Delivery Problem with Time Windows. In: Proc. of 13th IEEE Int. Conf. on Tools with Artificial Intelligence, ICTAI 2001 (2001)
14. 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.) CEEMAS 2005. LNCS (LNAI), vol. 3690, pp. 132–141. Springer, Heidelberg (2005)
15. Pillac, V., Gendreau, M., Guérat, C., Medaglia, A.L.: A review of dynamic vehicle routing problems. Tech. Rep. CIRRELT-2011-62, CIRRELT (October 2011)
16. Smith, R.G.: The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Transactions on Computer, 1104–1113 (December 1980)
17. Vinyals, M., Rodriguez-Aguilar, J.A., Cerquides, J.: A survey on sensor networks from a multiagent perspective. Comput. J. 54(3), 455–470 (2011)