Holonic-Based Environment for Solving Transportation Problems

Michał Gołacki, Jarosław Koźlak, and Małgorzata Żabińska

Department of Computer Science,
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
{kozlak,zabinska}@agh.edu.pl, michal@golacki.com.pl

Abstract. In this paper, a system based on the concept of holons for solving transportation problems is presented. Such a system focuses on the possibility of comparing the quality of solutions offered by the system with the results obtained from classical algorithms for commonly used sets of test problems. These problems were modified in order to take into consideration the fundamental features offered by the holonic approach.

1 Introduction

The application of methods which efficiently realise transport requests has a crucial significance for the competitiveness of companies. For the sake of the high complexity of algorithms for solving transport problems, different heuristic approaches are used. Widely examined classic transport problems such as VRP (Vehicle Routing Problem) and its different variants (e.g. *Vehicle Routing Problem with Time Windows* - *VRPTW*, *Pickup and Delivery - PDPTW, Pickup and Delivery with Time Windows* - *PDPTW*) operate on a simplified model of the world, which does not take into consideration essential problems that transport companies experience in reality. It would be especially necessary to take into account heterogeneity of transport units (trucks, trailers, and drivers). It seems to be useful to apply approaches based on holons because such a complex unit may be treated as a single element at most of its actions, having at the same time the capability of reorganising, removing and replacing its elements.

The goal of a proposed system is to solve transport problems starting from the static problem, hitherto tasks of the dynamic problem extended with the use of holonic agents. The classic problem consists in finding a solution for a set of transport requests, when the whole pool of requests is known at the start of computations. The solution makes a set of routes for transport units. All units are the same, and a transport network is a complete graph. A dynamic version of the problem means that a pool of requests is not known at the beginning of computations. Subsequent requests appear during the course of the simulation. It requires from a system a continuous adjustment to changes. It is not necessary that a graph of a transport network is complete, but it may be, to preserve a more likelihood to the classic problem. The next step is to take into consideration different elements and to treat transport units as complex structures, built with elements such as trucks, trailers, drivers, etc. Representatives of component classes may differ among themselves regarding properties and attributes: e.g. power and velocity of trucks, load of trailers, types of loads to carry, drivers' qualifications, their knowledge of routs, languages, experience or a number of hours worked without any break.

2 State-of-the-Art

Holonic multi-agent systems. Combining the approach based on holons with the multi-agent systems has been a subject of a significant amount of research as the concept is promising however at the same time raises many important questions. They concern the structure and rules of how the holon organisation functions, in particularly, the relationship of the agent on the one hand and its limitations caused by being a part of larger structure - holons, the motivation of an agent to join holons and the protocols of actions needed for leaving and joining a holon.

To enable and facilitate the organisation of agents in holons, it is useful to describe the know-how of agents in a way such as to enable them to identify the agents possessing the capacity and features needed for them. In [13] a concept of capacity and its application in holonic multi-agent systems is presented. In [7] an adaptive architecture for multi-agent holonic systems which uses immunological algorithms is presented. This approach was applied to the management of a group of mobile robots playing football in an FIRA league.

The holonic agent approach may be applied to many domains and particularly to modelling the functioning of companies and supply-chains, in Distributed Control Systems [4] and management of a shipping company [3].

Transportation problem solving. Different approaches are used for solving transportation problems: evolutionary algorithms [8], tabu search ([10]), Squeaky Wheel Optimisation [11] or multi-agent approach. The multi-agent solutions are based on Contract Net Protocol [14] and for optimisation simulated trading [6] or DCSP (Distributed Constraint Satisfaction Problem) [5,12] are used.

The first multi-agent transportation system based on holonic approach was Teletruck [3]. As opposed to models used for solving classical transportation problems, it took a higher flexibility and diversity of elements into consideration – different kinds of trucks, trailers, containers, drivers as well as their features and characteristics. The main ideas of an algorithm for allocating requests was similar to the solution used in MARS system [6].

3 Concept of Multi-agent System

The purpose of the work was to built a system which is functionally close to TeleTruck and to concentrate on the analysis of its possibilities in comparison with existing classic algorithms. Next, we want to carry out a comparison of results gained by such a system with classic solutions of transportation problems. For analysis we chose PDPTW and extensions introduced by us, which are related to holonic approach. The system enables a solution of static and dynamic problems; it may operate in an environment based on Euclidean distance and a graph representing road connections between locations.

3.1 Agents' Description

In the constructed system the following main types of agents exist: Dispatcher Agent (DA), Driver Agent (DRA), Truck Agent (TKA), Trailer Agent (TRA), and Transportation Unit Agent (TUA).

Dispatcher Agent. DA plays a role of a manager in the system. It controls the work and life cycle of Transportation Unit Agents (TUA), as well as distributes transport requests between them in the best possible way. In the system only one Dispatcher Agent exists.

$$DA = (G_D, S_D, K_D, A_D) \tag{1}$$

where:

G_D - goal which is to minimise the function:

$$G_D = \Sigma_{i=1}^{n} Cv^i + \Sigma_{i=1}^{n} Cd^i TD^i \tag{2}$$

where n – number of $TUAs$, Cv^i – cost of exploitation (mainly cost of amortisation and driver's salary) of TUA^i, TD^i – distance travelled by TUA^i, Cd^i – average cost of $TUA\ i$ for travelling a distance unit .

S_D – state — $S_D = \{LocAgent_j\}, j = 1 \dots n$ estimated location of all agents j,

K_D – knowledge, $K_D = \{Reqs, StatReqs, Env\}$, where $Reqs$ – information about requests, $StatReqs$ – information about status of requests, which assigns, to each of them, a value from the set {received, rejected, allocated, picked–up, delivered}, Env - information about transport network,

A_D - actions to be realised, $A_D = \{AllocReq, GetAgPos, SendAgPos\}$, $AllocReq$ – auction of a request and its allocation to a Transportation Unit, $GetAgPos$ – get info about agent positions, $SendAgPos$ – send info about agent positions.

The more detailed descriptions of actions, taking into consideration elements of the agent having a direct influence on them (left parts of expressions, before arrows) and being modified as a result of the action (right parts of expressions, after arrows) are as follows:

$AllocReq$: $Reqs \times StatReqs \rightarrow StatReqs$

$GetAgPos$: $LocAgent_j \rightarrow LocAgent_j$

$SendAgPos$: $LocAgent_j \rightarrow another\_agent$ (communication action, sending information to the other agent)

Truck Agent. Truck Agent i (TKA^i) represents a main element (vehicle) of TUA which is able to travel on its own together with the driver but without a load capacity:

$$TKA^i = (S^i_{TKA}, K^i_{TKA}, A^i_{TKA}) \tag{3}$$

where:

S^i_{TKA} – state: $S^i_{TKA} = \{TKAf^i, Loc^i, Bel^i_{TUA}, Res^i_{TUA}$, where $TKAf^i$ – TKA features (among others power and acceptable types of connections with a trailer), Loc^i –location, Bel^i_{TUA} – belonging to the TUA, Res^i_{TUA} – reservation for TUA and the due time.
K^i_{TKA} – knowledge $K^i_{TKA} = \{Info^i_{DRA}, Info^i_{TRA}\}$: information about neighbouring (in the same depot) drivers ($Info^i_{DRA}$) and trailers ($Info^i_{TRA}$),
A^i_{TKA} – actions $A^i_{TKA} = \{Join^i_{TUA}, Disj^i_{TUA}\}$: $Join^i_{TUA}$ – join to the TUA, $Disj^i_{TUA}$ – disjoin from the TUA.

Trailer Agent. Trailer Agent i (TRA^i) represents everything which is used by the truck to carry the load:

$$TRA^i = (S^i_{TRA}, K^i_{TRA}, A^i_{TRA}) \tag{4}$$

where:

S^i_{TRA} –state, $S^i_{TRA} = \{TRAf^i, Loc^i, Bel^i_{TUA}, Res^i_{TUA}\}$, where $TRAf^i$ – TRA features (among others kind of load, maximal and current load, acceptable types of connections with a truck), Loc^i –location, Bel^i_{TUA} – belonging to the TUA, Res^i_{TUA} – reservation for TUA and the due time.
K^i_{TRA} – knowledge, $K^i_{TRA} = \{Info^i_{DRA}, Info^i_{TKA}\}$, information neighbouring (in the same depot) drivers ($Info^i_{DRA}$) and trucks ($Info^i_{TKA}$).
A^i_{TRA} actions: $A^i_{TRA} = \{Join^i_{TUA}, Disj^i_{TUA}\}$, where $Join^i_{TUA}$ – join to the TUA, $Disj^i_{TUA}$ – disjoin from the TUA.

Driver Agent. Driver Agent i (DRA^i) represents a driver of a truck:

$$DRA^i = (S^i_{DRA}, K^i_{DRA}, A^i_{DRA}) \tag{5}$$

where:

– S^i_{DRA} –state, $S^i_{DRA} = \{DRAf^i, Loc^i, Bel^i_{TUA}, Res^i_{TUA}\}$ where $DRAf^i$ – DRA features (among others hitherto performed working hours, licences for driving given kinds of trucks and shipment, route knowledge, foreign languages knowledge, preferences concerning intermediate points), Loc^i – location, Bel^i_{TUA} – belonging to the TUA, Res^i_{TUA} – reservation for TUA and the due time.
– K^i_{DRA} – knowledge, $K^i_{DRA} = \{Info^i_{TKA}, Info^i_{TRA}\}$: information about neighbouring (in the same depot) trucks ($Info^i_{TKA}$) and trailers ($Info^i_{TRA}$),
– A^i_{DRA} actions: $A^i_{DRA} = \{Join^i_{TUA}, Disj^i_{TUA}\}$, where $Join^i_{TUA}$ – join to the TUA, $Disj^i_{TUA}$ – disjoin from the TUA.

Transportation Unit Agent. Transportation Unit Agent i (TUA^i) manages the transport unit which realises transport requests. Together with other units it participates in auctions of transport requests, and then it realises the assigned requests.

$$TUA^i = (G^i_{TUA}, P^i_{TUA}, Loc^i_{TUA}, S^i_{TUA}, K^i_{TUA}, A^i_{TUA}) \tag{6}$$

where:

G^i_{TUA} agent's goal to obtain the maximal possible income:

$$G^i_{TUA} = Cv^i + Cd^i TD^i \tag{7}$$

where n – number of TUs, Cv^i – cost of exploitation (mainly cost of amortisation and driver salary) of TUA^i, TD^i – distance travelled by $TUA\ i$, Cd^i – average cost of TUA^i for travelling a distance unit .

P^i_{TUA} - planned travel route, contains list of locations to be visited, performed pick up and delivery operations and holon reorganisation,

Loc^i_{TUA} current position of TUA,

S^i_{TUA} - agent's state, $S^i_{TUA} = \{AlReqs^i, HolStruc^i, Res^i_{DRA}, Res^i_{TRA}, Res^i_{TKA}\}$, where $AlReqs^i$ – allocated requests, $HolStruc^i$ – information about holon structure and component agents, reserved drivers (Res^i_{DRA}), trucks (Res^i_{TRA}) and trailers (Res^i_{TKA}),

K^i_{TUA} - agent's knowledge $K^i_{TUA} = \{Reqs^i, Env^i\}$ comprising $Reqs$ - set of requests, Env - information about environment (transport network),

A^i_{TUA} - actions of the TUA:

$$A_{TUA} = \{ReqPart^i, Move^i, PickUp^i, Delv^i, HolCrt^i, HolDis^i, HolRorg^i\}, \tag{8}$$

where $ReqPart$ - participation in auctions of transport requests, $Move$ – transfer between locations, $PickUp$ - loading, $Delv$ - unloading, $HolCrt$ – holon creation, $HolDis$ – holon dissolution, $HolRorg$ – holon reorganisation.

The more detailed description of actions is as follows:

$ReqPart$ - ($AlReqs^i$, Env^i) $\rightarrow$ ($AlReqs^i$, P^i_{TUA}),
$Move$ – $Loc^i_{TUA} \rightarrow Loc^i_{TUA}$
$PickUp$ - ($AlReqs^i$, Loc^i_{TUA}) $\rightarrow AlReqs^i$
$Delv$ - ($AlReqs^i$, Loc^i_{TUA}) $\rightarrow AlReqs^i$
$HolCrt$ – ($HolStruc^i$, Res^i_{DRA}, Res^i_{TRA}, $Res^i_{TKA}\}$) $\rightarrow$ ($HolStruc^i$, Res^i_{DRA}, Res^i_{TRA}, $Res^i_{TKA}\}$)
$HolDis$ – ($HolStruc^i$, Res^i_{DRA}, Res^i_{TRA}, $Res^i_{TKA}\}$) $\rightarrow$ ($HolStruc^i$, Res^i_{DRA}, Res^i_{TRA}, $Res^i_{TKA}\}$)
$HolRorg$ – ($HolStruc^i$, Res^i_{DRA}, Res^i_{TRA}, $Res^i_{TKA}\}$) $\rightarrow$ ($HolStruc^i$, Res^i_{DRA}, Res^i_{TRA}, $Res^i_{TKA}\}$)

3.2 Optimisation Algorithm

Allocation of requests between agents is performed on the basis of the Contract Net Protocol.

The transportation unit agents (completed or not) announce cost estimations of request realisations. These estimations may be based on capabilities of transportation units (parameters of the unit, its location and already accepted request) or the estimated possibilities of holon creation by them, which perform auctions of offers of given components which could make a holon (section 3.3 and section 4).

Requests are assigned to these agents which may realise them with the smallest increase of costs. Next, optimisation of solution is carried out – every given interval of time each TUA tries get rid of offers which are less profitable for it from the point of view of costs and it sends them to auctions. In this way the requests may be transferred to other TUAs (section 4).

The holon can also reorganise itself (section 3.3) to obtain a better configuration of components to fulfil requests allocated to it at lower costs.

3.3 Holon Organisation

Cooperation between agents consists in completion of transport units needed to execute requests and their common realization, as well as minimization of costs of a given set of requests. The applied model assumes that agents resign from a part of their independence only. The whole holon is represented by Tranport Unit Agent - TUA, which communicates with the whole society. It has the right to assign resources, plan, and negotiate with other agents on the basis of plans and aims of component agents. At the same time a task of TUA is to replace elements of the holon or to join and separate them.

Holons are created when a new request is received, if it enables a cheaper realisation. Moreover, one can assume that some holons already exist at the beginning of simulation. The created TUA does not have any elements of holon at the beginning, so it has to be created from the basis. A task is sent to all types of elements, and then a unit waits for their replies. After it receives them, the best offers are chosen, and a cost of task execution is computed and then an offer is sent to Dispatcher. In cases where a positive answer is received, elements are informed and a holon is created. Other elements receive a negative reply. Such an answer is sent to all elements of a holon, when the holon is not chosen for the task. Then, such a unit stops to exist. If a holon is created, its elements loose a part of their autonomy. They will not be able to receive information about new requests from units other than the unit which is the manager of their holon and they will not be able to send answers to such requests.

Reorganisation of a holon may be done in cases of new requests which cannot be realized by a given unit.

Dissolving of a holon takes place when a transportation unit performs all entrusted tasks. Resources are freed and they may join other holons, whereas TUA waits for subsequent requests. If they appear, it tries to collect the next transportation unit and starts to realise them.

4 Realisation

The system is divided into some modules. They are, viz. computational module, layer controlling a simulator including user's interface, means of visualisation and tests generator. In particular, the computational module comprises agents of the system which are responsible for finding a solution. Tests generator is able to change static tests into dynamic ones, to increase the number of available transport bases as well as modify a transport network. Agents use JADE [2] platform (Java Agent Development Framework, Version 3.3) and communicate among themselves undertaking actions to solve a given problem.

Realisation of agents' tasks in the system is related to communication between them. Main tasks of agents are following:

– assignment of a new request, when holon (TUA) with requests already exists,
– assignment of a new request, when holon (TUA) already exists, but without any requests,
– assignment of a new request, when holon (TUA) does not exist and it is necessary to create it (creation of TUA),
– lack of requests for holon (TUA) after finishing all tasks, dissolution of the holon.

The assignment of a new request, when holon exists, but requests have not yet been assigned to it, is presented below. This situation is connected with the reorganisation of TUA (fig. 1). TUA receives a request and verifies its capability to realise the request. If such a possibility exists, Dispatcher Agent accepts TUA. Otherwise, if TUA is not appropriate to realise the request, it tries to reorganise itself. Let us assume, that an element which is not suitable to realise the given request is a trailer. Then trailers, which have not joined any TUA yet, are informed and send their offers. After choosing the best offer, the offer for

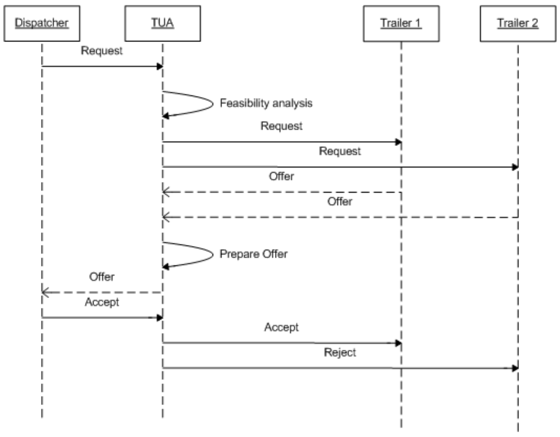


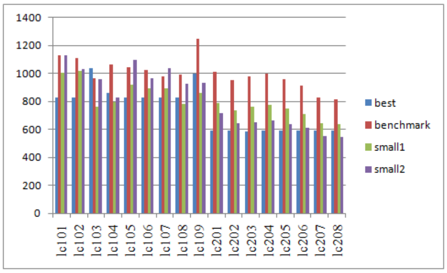
Fig. 1. Sequence of actions: a new request and TUA reorganisation

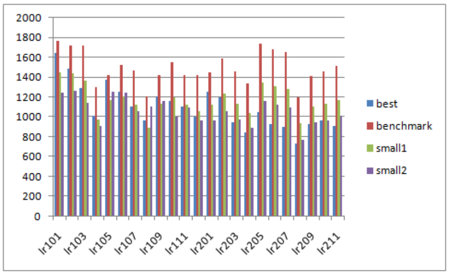
realising the given request is sent to Dispatcher. The chosen TUA transfers the information to the trailer which replaces the one present until now in the holon.

5 Experiments and Results

To verify the quality of solutions obtained, sets [1] of tests prepared by Li and Lim [10] have been used. Tests have been modified to include holons. Standard tests comprise of PDPTW with a different number of points of pickup and delivery. Results obtained for problems with 100 pickup and delivery points are presented below. Problems may be categorised into classes where points of requests are located in clusters (LC), distributed uniformly (LR) or comprise points grouped both in clusters as well as distributed uniformly (LRC). Moreover lxx are problems with narrow time windows, whereas 2xx - with wide ones. Since we examine the activity of a transport company which possesses a pool of vehicles, the key parameter concerning criteria of optimisation is distance, which limits fuel consumption of available vehicles, and makes easier to be included in subsequent requests. Therefore the number of used vehicles does not play a primary role.

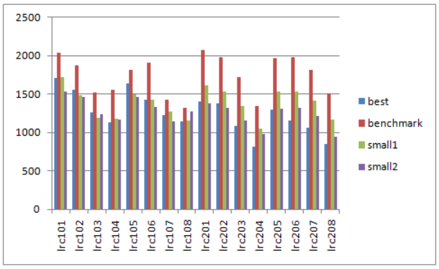
In subsequent charts (fig. 2a, fig. 2b and fig. 3a) results (costs) obtained for the above-mentioned groups of problems are shown. In each figure there are results for narrow and wide time windows. A modification introduced to the tests was the possibility of using trailers of different load capacities. Subsequent bars, viz. *best*, *benchmark*, *small1*, *small2* present costs of realisation of requests for the best known solutions of problems, our solutions for vehicles of load in agreement with those used in a benchmark, whereas for *small1*, and *small2* we have at our disposal trailers of different capacities - those of *small1* are smaller then those of *small2*. In particular, if for a given transport problem, the capacity of vehicles in a test set was x, then for *small1* problems we have at our disposal 10 vehicles for each capacity: x, 3x/4 and x/2, whereas for *small2*, 10 vehicles for each capacity: x/2, 3x/8 and x/4. Costs are related to travelled distance and capacity of the used trailer; they are calculated as: $CD = (CD_{DR} + CD_{TK} + CD_{TR})/3$, where CD_{DR}, CD_{TK}, CD_{TR} – average travel costs for a distance unit of Driver, Truck



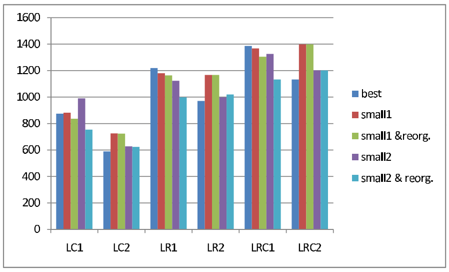


a) b)

Fig. 2. Results: Results for LC (a) and LR (b)problems with narrow and wide time windows



a)



b)

Fig. 3. Results: (a) LRC problems with narrow and wide time windows (b) Families of results for 100 request points, without and with holon reorganisation groups

and Trailer and each of these travel element costs CD_x are calculated as follows: where: $CD_x = (1/3 + (2 * Cp_{max})/(3 * Cp_d))$, Cp_{max} – maximal capacity of the given element, Cp_d – capacity of vehicles used in a standard solution.

The presented results show that costs of travel for benchmarks versions (that is, capacities in consent with those used to obtain best known solutions of benchmarks) usually only slightly differ from the best known solutions. However, using smaller vehicles in *small1* and *small2* problems, it was possible to decrease costs substantially, which is a proof that the whole capacities of vehicles were not usually used in solving test problems. It also shows advantages of flexible planning, when we have trailers of different capacities at our disposal, because it may limit costs significantly.

Aggregated results for main groups of benchmarks are presented in fig. 3b. There are LC1 and LC2, – clusters with narrow and wide time windows, LR1 and LR2 – uniformly distributed with narrow and wide time windows, LRC1 and LRC2 – clusters and uniformly distributed with narrow and wide time windows. For each case the results for different sets of vehicle capacities as well as for the algorithm without and with holon reorganisation are taken into consideration.

We can notice that the holon reorganisation significantly decreased costs of solutions for cases with narrow time windows. For the cases with wide time windows the improvement did not occur, it is related to the fact that capacities of vehicles exceeded considerably the needs.

6 Conclusions

In this paper, the results obtained using our holonic system for modelling and optimising transportation requests were presented. We described the pilot version of the application. Subsequent work will focus on the introduction of new algorithms for constructing routes and increasing the flexibility of algorithms for constructing holons. The next domain of work will take into consideration different kinds of critical situations and in particular, traffic jam formations, route closing as well as breakdown of vehicle or other holon elements. These aspects have een analysed in our various works in this area [9], and ind the future be integrated with the holonic approach.

References

1. Benchmarks - Vehicle Routing and Travelling Salesperson Problems, `http://www.sintef.no/static/am/opti/projects/top/`
2. Java Agent DEvelopment Framework, `http://jade.tilab.com/`
3. Burckert, H.-J., Fischer, K., Vierke, G.: Transportation scheduling with holonic MAS - the TELETRUCK approach. In: Third International Conference on Practical Applications of Intelligent Agents and Multiagents (PAAM 1998) (1998)
4. Choinski, D., Nocon, W., Metzger, M.: Application of the holonic approach in distributed control systems designing. In: Mařík, V., Vyatkin, V., Colombo, A.W. (eds.) HoloMAS 2007. LNCS (LNAI), vol. 4659, pp. 257–268. Springer, Heidelberg (2007)
5. Dorer, K., Calisti, M.: An Adaptive Solution to Dynamic Transport Optimization. In: Proceedings of the AAMAS 2005 industry track, Utrecht, The Netherlands (2005)
6. Fischer, K., Muller, J., Pischel, M.: Cooperative Transportation Scheduling: an Application Domain for DAI. Applied Artificial Intelligence, pp. 1–33 (1996)
7. Hilaire, V., Koukam, A., Rodriguez, S.: An adaptative agent architecture for holonic multi-agent systems. ACM Trans. Auton. Adapt. Syst. 3(1), 1–24 (2008)
8. Homberger, J., Gehring, H.: Two evolutionary meta-heuristics for the vehicle routing problem with time windows. INFORMS Journal in Computing 37(3), 297–318 (1999)
9. Konieczny, M., Koźlak, J., Żabińska, M.: Multi-agent crisis management in transport domain. In: Proceedings of ICCS 2009. LNCS. Springer, Heidelberg (accepted for publication, 2009)
10. 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 2001), Dallas, USA (2001)
11. Lim, H., Lim, A., Rodrigues, B.: Solving the Pick up and Delivery Problem using "Squeaky Wheel" Optimization with Local Search. In: Proceedings of American Conference on Information Systems, AMCIS 2002, USA (2002)
12. Neagu, N., Dorer, K., Calisti, M.: Solving Distributed Delivery Problems with Agent-Based Technologies and Constraint Satisfaction Techniques. In: Dist. Plan and Schedule Management, AAAI Spring Symp., USA. The AAAI Press, Menlo Park (2006)
13. Rodriguez, S., Gaud, N., Hilaire, V., Galland, S., Koukam, A.: An analysis and design concept for self-organization in holonic multi-agent systems. In: Brueckner, S.A., Hassas, S., Jelasity, M., Yamins, D. (eds.) ESOA 2006. LNCS, vol. 4335, pp. 15–27. Springer, Heidelberg (2007)
14. 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)