Multi Agent System for Resource Allocation and Scheduling

Vladimir Gorodetski, Oleg Karsaev, Victor Konushy

St. Petersburg Institute for Informatics and Automation, Russia
`{gor, ok, kvg}@mail.iias.spb.su`
`http://space.iias.spb.su/ai/`

Abstract. The paper considers multi agent approach and respective multi-agent system for resource allocation and scheduling within shipping logistics problem. A well known benchmark that is Vehicle Routing Problem with Time Windows is considered as a case study. The paper proposes an approach to the shipping logistics problem solving that includes auction-based resource allocation and scheduling, distributed reallocation algorithm and distributed version of the "look ahead" algorithm. The proposed approach is implemented and studied experimentally. The experimental results are outlined.

1 Introduction

The planning and scheduling tasks are of great concern for decades. To date a lot of ideas and paradigms are proposed. To cope with the computational complexity, the most of approaches use a decomposition of applied problems into weekly correlated subtasks and a kind of coordination. Artificial intelligence noticeably contributed to this problem through exploitation of knowledge-based search. A significant progress in the planning and scheduling is being provided by use of multi-agent paradigm which also exploits knowledge-based search but is additionally focused on distributed search and approximate coordination. The latter is based on various versions of Contract Net Protocol ([4]) implemented in various forms of auctions. Auction-based approach realizes an approximate knowledge-based coordination of decisions computed by agents possessing limited information.

Experience summarized in many publications proved that the auction-based coordination model is very profitable ([7], [9]). However, the scope of planning and scheduling is very broad, different task statements can differ significantly and these differences are firstly caused by the types of constraints peculiar to applications. At that some of very important types of planning and scheduling tasks are paid undeservedly insufficient attention within multi-agent system (MAS) scope. Between them the tasks with real-time or temporal constraints must be referred to. Such type of planning and scheduling tasks is the subject of this paper. Particularly, it presents model and software implementation of the multi-agent approach to solving of a widespread problem that is *shipping logistics* ([1], [8]).

In the most of planning and scheduling tasks the hardest constraints are caused by restricted resources. This is why resource allocation is an important component of many real-life planning and scheduling tasks. From algorithmic point of view it can be considered either as a part of the planning or a part of the scheduling. A mixed approach is also used. In the most applications of the shipping logistics task in question resource allocation is solved on scheduling phase.

Approximately, shipping logistics task is formulated in the paper as follows: given a "portfolio of contracts", potential contract executors possessing certain resources and constraints that include real-time ones, to search a "contract" allocation that is "good" in some sense formulated formally. The idea of the approach used for solution of this task is auction-based parallel solution of the resource allocation and scheduling subtasks constituting the shipping logistics problem. At that, distinctive features of the approach (as compared, for example, with the most known work [Fisher] dealing with the same application) which also determine the paper contribution are as follows:

- Use of "contract" reallocation procedure performed according to a new auction-based protocol, which significantly decreases the information exchange overhead thus providing the approach with higher efficiency.
- Use of distributed version of the protocol implementing "look ahead" approach that was developed and used for centralized planning ([2]).
- Implementation of an interactive mode of the shipping logistics task solving that makes it simple to combine power of multi-agent technology and user experience amplified by cognitive graphics.

The shipping logistics task is very interesting because many tasks from different application areas have close formal statements. Thus, experience acquired through shipping logistics task formalization and MAS-based implementation is very useful with regard to other tasks. An example is mission planning and scheduling ([1]) that is one of the hottest tasks in business and military applications. Technological issues are also very important.

The rest of the paper is organized as follows. In *section 2* an outline of the shipping logistics problem and benchmark used is given. *Section 3* is devoted to the description of some architectural issues of the MAS implementation used. *Section 4* describes auction-based resource allocation algorithm and also how time window constraints are met. *Section 5* describes the developed protocols aiming at distributed "contract" reallocation and distributed "look ahead" algorithm. Conclusion briefly presents experimental results obtained on the basis of the well known benchmark, and also outlines future work.

2 Vehicle Routing Problem with Time Windows

Vehicle Routing Problem With Time Windows (VRPTW) is a well known benchmark developed by Solomon. Content of the problem is the following. A shipping company business is to deliver loads to customers according to a set of orders which specify the

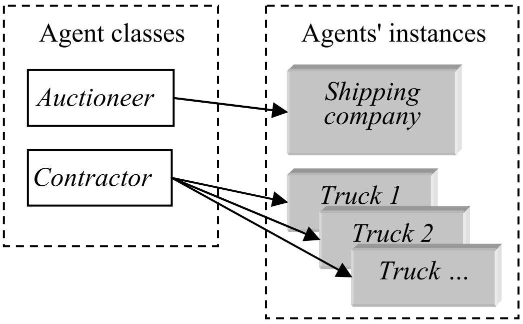


Fig.1. Agent classes and agents' instances

requested services and requirements and constraints to the quality of services to be provided. The components of each order specification are as follows:

- coordinates of customer location,
- weight of load,
- service time window indicating the admissible real time interval for delivery and unloading,
- duration of unloading ("service time").

Load delivery is carried out by vehicles (trucks) with constrained capacity that is upper threshold for total weight of loads that is admitted for it.

Each customer has to be serviced within the given service-time window. If a vehicle reaches a customer before the service-time determined by time window, a waiting time occurs. The distance between a pair of customers is considered as equal to the travel time between them. The total time spent by a vehicle for all allocated services provision is determined as the sum of the total travel time, total waiting time and total service time[1].This time is one of the components of the quality of service vector mapped to the solution produced by multi-agent planning and scheduling system. Informally speaking, the main goal of MAS in question is to search a schedule that provide all customers with the requested services while minimizing the total time spent and meeting vehicles' capacity constraints and customers' time-service windows. In an alternative variant of problem statement minimization of the number of vehicles used is considered as the main goal instead of total time spent.

The benchmark consists of data sets of different characteristics. Task statements differ in the time-service windows, scheduling horizon and customers' coordinates. Scheduling horizon determines the number of customers per route. Customers' locations in different data sets are randomly generated, clustered and mixed. It is worthy to note that time-service windows noticeably affect the admissible routes. More details can be found in ([8]).

3 Agent Classes, Roles and Software Agent Instances

Development and deployment of VRPTW MAS was carried out with use of a multi-agent platform called Multi-agent System Development Kit, MASDK ([6]). In this tool kit, functionalities and data structures that practically independent from application to application are implemented as a unified (reusable) components united within so-called *Generic Agen*t. The latter comprises a hierarchy of invariant software

---

[1] It is assumed in the benchmark that the time in measured in distance units ([8]).

classes and generic data structures. Generic agent is considered as a nucleus that is "bootstrapped" by the developer through specialization of the software classes and data structures and through replication of software agent classes in their instances composing MAS. Thus, design and implementation of the software agents of an applied MAS is reduced to the specification their application-oriented attributes and properties which are performed in three steps. At the first step, the shared application ontology and also agents' interaction protocols are specified by designers. The protocols are represented in terms of roles assigned to particular agents. At the second step, the functionalities of agents' classes and respective private components of the application ontology are specified. At the third step, the instances of agents' classes that are software agents are generated and also software agents' particular attributes and properties are specified. The above steps are supported by user-friendly editors provided with clear interfaces. In addition to Generic Agent, these editors together constitute the second component of MASDK tool kit.

The correlation between mentioned above notions "agent class" and "software agent's instance" is similar to the correlation between the notions "software class" and its instance in object-oriented programming. For example, specifications of application-oriented behavior scenarios defined for an agents' class are inherited by all software agent's instances of agents' class. Fig 1 graphically demonstrates the above property of the respective agents of the VRPTW MAS. In the application under consideration, two agents' classes are introduced, *Auctioneer* and *Contractor*. The former agents' class has single software agent instance, *Shipping company*, which performs the respective role and behaviors. The latter agents' class has many software agent's instances, that are *Truck 1, Truck 2, Truck ....*

The agents' classes determine the agents' roles and, respectively, their roles in interaction protocols. In VRPTW MAS several interaction protocols intended for solving different tasks are used. For example, these tasks are allocation and reallocation of resources and scheduling. Each protocol supposes that two classes of agents (roles) are involved in the interactions, *Auctioneer* and *Contractor,* at that the former has the single instance whereas the last one has multiple instances. In general case agents' classes can be assigned several roles within the same or different protocols. The functions performed by agents of different roles in VRPTW MAS are as follows:

Auctioneer

1. Meta-level management of the VRPTW task solving. This functionality manages the meta-level processes in MAS. In particular, it determines what of the processes must at the moment be executed, i.e. what of resource allocation and scheduling, preparation to execution of the process of resource reallocation or rescheduling, and what interaction protocol correspond to this or that chosen process.
2. Sequencing of orders in both allocation and reallocation auctions.
3. Analysis of quality of services offered by "contractors", winner selection.
4. Management of resource reallocation preparatory phase which aims at forming of a subset of orders considered as the subjects for reallocation.
5. Selection of the subset of potential contractors for the next auction.

Auctioneer also solves a number of auxiliary tasks.

All above functionalities can be executed in both automatic and interactive modes. The latter supposes that user can intervene into the solution procedure. In particular the approval of the final decision is here the user's responsibility. In interactive mode, user can agree with the allocation proposed by MAS (auctioneer) or disagree and select it manually.

Contractor

1. Any-time algorithm for search of the shortest route through-passing nodes corresponding to the customer locations, i.e. loads destinations.
2. Calculation of bid proposals. The "cost" of a bid ("cost" of the order execution) is calculated as increase of total distance along the new rout in comparison with the total distance along the previous one.
3. Participation in computing of the subset of orders for reallocation.

4 Resource Allocation

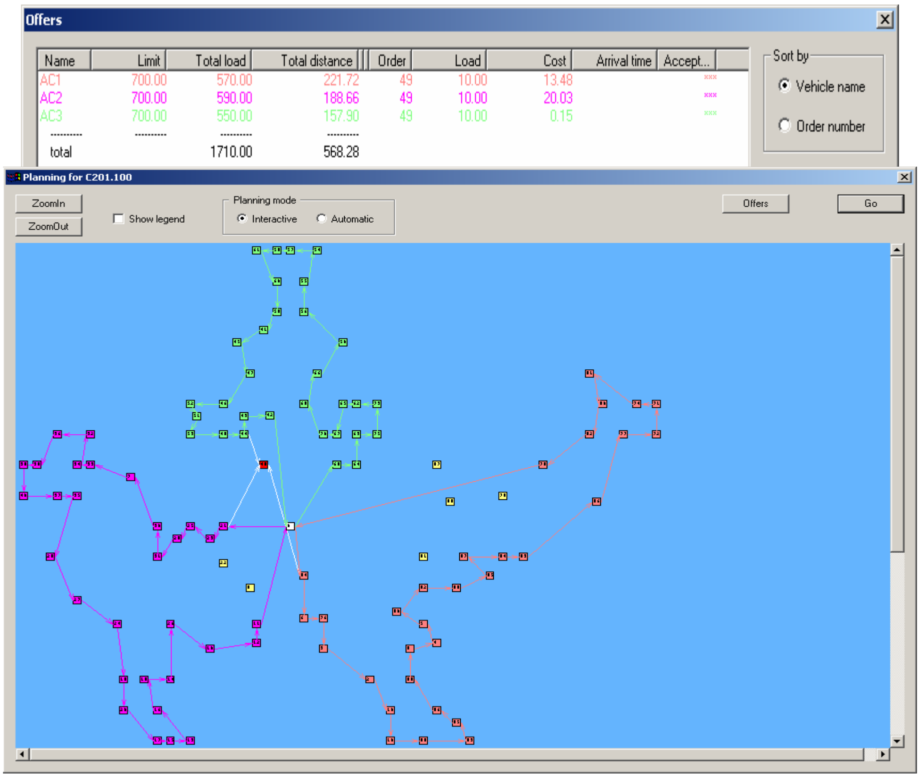
Resource allocation is realized as a sequence of auctions, at that each of them aims at allocation of a particular order to a contractor. Every auction consists of the following sequence of steps (tasks):

1. Selection of an order "to be sold" during forthcomong auction.
2. Selection of a multitude of the potential contractors to participate in the forthcoming auction.

Auctioneer solves both above tasks.

3. Auction realization. In it, contractors ftom the selected subset receive from the auctioneer specification of the order to be allocated in current auction, calculate certain offers and send them to auctioneer.
4. Analysis of offers and decision making. The decision can be one of the following:
    - Auctioneer refuses from the current order allocation due to lack of satisfactory offers if no contractor has proposed an admissible variant.
    - Auctioneer selects a winner and allocates to it the order to execute.
    - Auctioneer cannot select a predominant winner between offers although admissible variants exist and decides to postpone the allocation of the current order.
    - Auctioneer extends the multitude of contractors via adding a new one. This outcome takes place in case if the decision is not made in the current auction.
5. Auctioneer decides whether it is necessary to continue order allocation procedure. In general case resource allocation procedure is interrupted if all orders were proposed for bargaining at least one time independently of the results, i.e. even if some orders were not allocated due to lack of admissible proposals. Otherwise the allocation procedure continues.

As a rule, several iterations for order allocation are needed to find the final one. It is also possible to generate several such decisions between which the user is authorized to select the final one.



Name	Limit	Total load	Total distance	Order	Load	Cost	Arrival time	Accept...
AC1	700.00	570.00	221.72	49	10.00	13.48		***
AC2	700.00	590.00	188.66	49	10.00	20.03		***
AC3	700.00	550.00	157.90	49	10.00	0.15		***
----------	----------	----------	----------					
total		1710.00	568.28					

Sort by: ● Vehicle name ○ Order number

☐ Show legend

Planning mode: ● Interactive ○ Automatic

Fig.2. Example of interactive mode of auction management for VRPTW task

The most significant influence on the quality of the obtained decisions the tasks 1 and 4 affect. In the automatic mode, auctioneer makes decisions on the basis of knowledge base rules. As a matter of fact, the rules used in the VRPTW MAS are of heuristic kind and practically several sets of rules are used at that different sets of rules leads to different solutions obtained. For example, in procedure implementing the task 1 aiming at selection of order allocation sequencing, a number of rules are used. Between them, the simplest one is "first received first allocated". Other rule is sequencing of order allocation according to the increase of time service beginning. Both these rules are very simple and their truth values can be computed on the basis of the VRPTW database. The third rule is more sophisticated and requires additional computations. This rule supposes that firstly the most remote orders are allocated. Its usage leads to better results, i.e. to finding of shorter routes. There are also used several other rules of heuristic nature.

To select auction winner (Task 4), the proposed bids are compared and several rules are user to sequencing them according to "better-worth" order relation. The contractors' offers include attribute specifying "cost" of the order execution (increase of route distance), and also a number of other attributes that specify certain integral evaluations of spatial configuration of the location of the customer of current order and locations of customers of orders that have already been assigned to the contractors in the previous auctions. The simplest rule of winner selection is allocation the order to the contractor proposed the smallest "cost". The second variant is firstly to take into account the "cost" of offers, then, if there are several proposals with the close "costs", to take into account the second preferable attribute and so on.

Interactive mode of the VRPTW MAS performance permits to take several advantages. A characteristic feature of the VRPTW task as well as of several other planning and scheduling tasks is that experience accumulated during VRPTW MAS use can prompt to user new potentially very useful heuristics (rules) and this is an argument to use the interactive mode of the VRPTW MAS performance. Additional argument is that sometimes the user can select better decision than MAS. An example is if auctioneer receives offers with close qualities of services, the interactive mode of winner selection can be more profitable since user can take into account information concerning configuration of the customers' locations that cannot be processed automatically.

In particular, Fig.2 presents an example of user interface window which definitely can help to user to make or correct decisions at the current step of the VRPTW task solving procedure. The lower area of the window in Fig.2 depicts the truck routes corresponding to the order allocations that have already been done previously and proposals for current order received from contractors. The orders that have not yet been allocated are also depicted in this window. The upper window presents the table specifying the contractors' attributes of the states of contractors and their offers. The presented graphics and table provide user for the important information needed for winner selection. Additionally, in the window in Fig.2 the decision made by auctioneer on the basis of its heuristic knowledge base is also depicted. The user can either agree with the auctioneer or select its own decision. It is worthy to note that while selection a winner, the user additionally takes into account the visually presented information concerning the orders that were not yet allocated.

The same is true with regard to solving of other tasks, for example, with regard to tasks 1, 2 and 5. While solving the task 1, the user can select the following order to allocate at the forthcoming auction. While solving the task 2, the user can correct the list of participating contractors. While solving the task 5, the user can interrupt the process of resource allocation and initiate reallocation procedure.

It should be noted that interactive mode makes it possible to use heuristic rules that are more intuitive and cannot be specified formally in a simple way, for example, rules reflecting the topology of truck routes selected, locations of the order to be allocated and locations of unallocated orders.

5 Resource Reallocation

Resource reallocation procedure aims to improve the quality of the current decision (resource allocation) via reallocation of a subset of the orders that were allocated at the previous auctions. The major motivation of using resource reallocation procedure is that it can lead to changes of the routes of trucks leading to a less total cost. It is shown in ([5]) that in VRPTW application, such kind of allocation modifications can result in noticeable improvement of the resource allocation quality. Practice proved that this improvement takes place typically for each particular VRPTW task. Indeed, each auction takes into account the attributes of the current order and previously made decisions but cannot anticipate new constraints raised as a result of the currently made decision. Reallocation procedure allows to take into account the consequences of all the decisions made during allocation and thus to improve scheduling on the whole.

Let us consider the peculiarities of the reallocation procedure used in VRPTW MAS in question and firstly consider the reallocation protocol. The protocol of agent interaction proposed for the same task in ([5]) and called "*simulated trading protocol*" supposes to use both vertical agents' interactions (between auctioneer and contractors) and horizontal ones (between contractors). The protocol used in the developed and implemented *VRPTW* MAS differs in certain features from the simulated trading protocol. Particularly, the developed protocol doesn't use horizontal interactions between contractors because the latter leads to the significant increase of the agent communication traffic overhead and also requires significant increase of the computation cost.

The proposed protocol is executed in two stages. *First*, the subset of orders to be reallocated is chosen. This subset is chosen according to a criterion that helps to determine the orders which reallocation can potentially lead to increase of the allocation plan quality. *Second*, the reallocation procedure itself is performed. This procedure uses rules of heuristic kind. Let us specify the reallocation algorithm in more detail.

1. Each contractor analyzes its own route and refuses from executions of the orders that it would like to pass to someone else. A criterion determining such kind of orders uses analysis of each order "cost" and its location within the contractor's route.
2. Each contractor analyses orders from the batch of its offers during allocation phase but assigned to other contractors to select from them a subset of those that can be executed by it for the less total "cost".
3. Join of the sets of orders offered by contractors for reallocation is sent to auctioneer. The latter forwards these orders to the contractors that were awarded by the respective contracts in allocation auctions. These contractors compare the received costs with those own costs and refuse from the offers of less "cost" (step 1).
4. Auctioneer selects the contractors that are potential executors for the orders to be reallocated and sends to each such contractor the list of respective orders.
5. Each contractor analyzes the auctioneer's offers and for every offered order calculates whether it is capable to execute the latter without violation of the constraints.
6. The list of such orders is returned to the auctioneer.

The steps 1-6 together can be interpreted as a preparatory phase. The reallocation procedure itself is realized at the subsequent steps.

Reallocation procedure is close to the basic allocation one. It is modified according to the ideas of the "*look ahead*" approach that was proposed for the task "pure scheduling without resource allocation" ([2]). Particularly, auctioneer acts according to the following steps.

7. Auctioneer determines the sequencing of orders reallocation taking into account the information returned by contractors at the step 6. To determine the above se-

quencing, auctioneer for each order returned computes the total number of contractors that are "ready" to execute it. The first order to be presented to auction is the order assigned the least number.

8. The selected order (with the least number of pretenders for its execution) is allocated through auction in which the winner is determined in a more sophisticated way than in the auction used in the allocation algorithm (see section 4). The potential contractors send to auctioneer two attributes: 1) the "cost" of the contract execution and 2) the list of the orders that it would be not capable to execute if it was the winner. Auctioneer uses several heuristics (rules) to determine the winner. The aim of the aforementioned heuristics is to select the winner between those contractors that propose the best "close" costs and, at that preserve for each remaining (unallocated) order as many potential executors as possible. Formally this condition means that the second aim is to maximize the number of potential contractors for the order having the least number of pretenders. The subset of unallocated orders which the winner became incapable to execute after its win is deleted from the respective list formed on the step 6.
9. If the set of unallocated orders is to this step not empty then the algorithm goes to the step 7 in which auctioneer sequences the rest of orders to determine the first one to be allocated (see step 8). If the set of unallocated orders is empty, then the algorithm stops.

Reallocation procedure is repeated iteratively and stops in cases if either allocation is not improved or if time resource assigned for allocation procedure is exhausted.

6 Conclusion: Experimental Results and Future Work

Testing of the developed approach and implemented MAS was carried out on the basis of the VRPTW benchmark data sets, generated by Solomon ([8]). The results of simulation can be concluded as follows. *First*, reallocation procedure improves the results of allocation procedure approximately from 10 till 15%. For instance, allocation of orders corresponding to data set RC107 ([8]) results in the total distance equal to 1358 whereas reallocation procedure leads to the total distance equal to 1168. *Second*, one can compare results of the developed approach with the other ones implemented in not multi agent manner. For instance, one of such approaches, implemented by use of the branch-and-bound method, is described in ([3]). This algorithm allows to search about optimal solutions and time of solving depends on data set. Experiments used this algorithm were performed on 300MHz Pentium II computer and parallel experiments were performed on a cluster of 32 of such workstations. Particularly for data set RC111 the algorithm was capable to search the result with about optimal total distance equal to 1048 but consumes for this computation 41879 seconds. For the same data set the developed multi agent system generates result with total distance equal to 1244 but consumes for it only 1800 seconds. At that experiment was performed for the case in which all agents were deployed on one 1400 MHz Pentium IV computer.

The further research will concern the followings:

- User-based extraction of new heuristics.

- Development of the reusable components peculiar to shipping logistics problem aiming at development of the problem-oriented MAS software tool.
- Enrichment of the MAS functionalities aiming at getting ready to development and implementation of practically interesting MAS applications in the shipping logistic scope.

Acknowledgement

This research is supported by grant of European Office of Aerospace R&D (Project #1992P)

References

1. Becker, M. and S.F. Smith, "Mixed-Initiative Resource Management: The AMC Barrel Allocator", In *Proceedings 5th International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2000)*, Breckenridge, CO, April, 2000.
2. A.Cesta, A.Oddi, and S.Smith. A Constraint-Based Method for Project Scheduling with Time Windows. Tech. Report CMU-RI-TR-00-34, Robotics Institute, Carnegie Mellon University, February, 2000.
3. Cook,W and J.L.Rich. A Parallel Cutting-Plane Algorithm for the Vehicle Routing Problem with Time Windows. Technical Report TR99-04 Department of Computational & Applied Mathematics, Rice University, USA, http://www.isye.gatech.edu/~wcook/papers/vrptw.ps.
4. R.Davis and R.G.Smith. Negotiation as a metaphor for distributed problem solving, *Artif. Intell.,* vol.20, pp. 63-109, 1983.
5. K.Fisher, B.Chaib-draa. A Simulation Approach Based on Negotiation and Cooperation Between Agents: A Case Study. In IEEE Transactions On Systems, Man, And Cybernetics – Part C: Applications And Reviews, Vol. 29, No.4, November 1999.
6. V.Gorodetski, O.Karsaev, I.Kotenko, A.Khabalov. Software Development Kit for Multi-agent Systems Design and Implementation.In B.Dunin-Keplicz, E.Navareski (Eds.), From Theory to Practice in Multi-agent Systems. Lecture Notes in Artificial Intelligence, Vol. # 2296, pp.121-130, 2002.
7. Multiagent systems: a modern approach to distributed artificial intelligence / edited by Gerhard Weiss, The MIT Press, London, 1999.
8. Solomon. http://web.cba.neu.edu/~msolomon/problems.htm
9. W.Vickrey. Counter-speculation, auctions, and competitive sealed tenders, Journal of Finance, Vol.16, No.1, pp.9-37, March 1961.