

Continuous Delivery Scheduling and Execution with Multiagents

Jun Sawamoto*, Member

We propose a cooperative multiagent system for solving the delivery scheduling problem where the environment changes dynamically. In the truck delivery problem, the environmental conditions such as the traffic condition of roads changes and additional delivery orders arrive dynamically during the execution of the delivery, then the reactive and adaptive measures are required to cope with such situations. Multiple agents for the delivery center and delivery trucks cooperate to cope with the re-scheduling problem of additionally arriving orders after once the delivery has started according to the initial delivery plan. We show that the proposed method provides a flexible solution for the dynamic re-scheduling problem.

Keywords : Delivery scheduling, Re-scheduling system, Multiagent system

1. Introduction

Problems of truck delivery scheduling in logistic systems involve the generation of plans under a variety of constraints, which change continuously depending on numerous factors, and at present, solutions of such problems rely on the efforts of human experts. There have been studies combining numerical methods and AI (heuristic) techniques to construct a system for vehicle routing problem with time window (VRPTW) ⁽¹⁾, researches utilizing digital road network information ⁽²⁾⁽³⁾, and researches employing domain models ⁽⁴⁾.

Algorithms for pickup and delivery problem with time windows (PDPTW) are studied as a generalization of vehicle routing problem ⁽⁵⁾. The MARS system ⁽⁶⁾ models PDPTW within a society of shipping companies with multiple trucks as a cooperative multiagent system.

However, several themes of research on delivery scheduling problems require further work, among them, (1) realization of adaptive systems conforming to actual problems and accommodation of multiple evaluation parameters, (2) creation of problem-solving models using knowledge of multiple levels, and (3) establishment of a method for deriving, within a practically useful length of time, approximate solutions within a permissible range which satisfy the imposed constraints. An effective framework for solving such problems is sought.

Methods for cooperative problem solving are now being developed as a basic technology for use in scheduling, but at present there is insufficient application of such methods to delivery scheduling problems of the type addressed in this paper.

We have been conducting studies on the application of cooperative problem solving models to truck delivery scheduling ⁽⁷⁾. However, this re-scheduling model has a shortcoming of not well adapting to changes, which are not incorporated as the default knowledge at the time of initial delivery scheduling.

We propose a new method employing a negotiation process by cooperative agents to achieve more adaptive real-time re-scheduling procedure. We discuss the multiagent functions of a decision-making support system in solving delivery problems,

aiming at the realization of a highly responsive system. Delivery scheduling problems contain the separate problems of generating an initial static delivery plan and of dynamic re-scheduling to correspond to the real world changing at the time of actual plan execution (delivery execution). In particular, the latter coping with dynamic changes in the real world incorporates the notion of so-called continuous planning ⁽⁸⁾.

In this paper, we propose a cooperative multiagent system for solving the delivery scheduling problem with time window where the environment changes dynamically. In section 2 of this paper we describe delivery-scheduling problem. In section 3 we present the method proposed for solution of truck delivery re-scheduling problem with application to benchmark problems, and in section 4 we evaluate the result and discuss remaining problems.

2. Truck Delivery Scheduling Problem

2.1 Overview of the Truck Delivery Scheduling The problem we attempt to address is that of planning operations involving multiple trucks dispatched from a delivery center to deliver goods to numerous locations; the schedule includes personnel allocation, vehicle allocation, delivery route selection and other parameters, with consideration paid to delivery costs. Specifically, a delivery scheduling system has the following goals;

- (1) Reduction of costs through efficient utilization of personnel and vehicles,
- (2) Improvement of service by meeting deadlines for delivery to locations,
- (3) Rapid generation of a delivery schedule without relying on human experts,
- (4) Generation of large-scale delivery schedules within a practical time frame.

Fig. 1 shows a representative model of a truck delivery scheduling problem. Orders collected at one delivery center are divided among multiple trucks and are delivered to locations dispersed throughout a delivery region. The delivery region is partitioned into a number of delivery sub-regions. The delivery schedule implemented at the delivery center determines delivery routes, allocates personnel and trucks, and decides times for departure and arrival at the center and delivery locations. Examples of such vehicle routing applications include;

* Space Solution Systems Development Division, Mitsubishi Electric Corporation
2-2-3 Marunouchi, Chiyoda-ku, Tokyo 100-8310

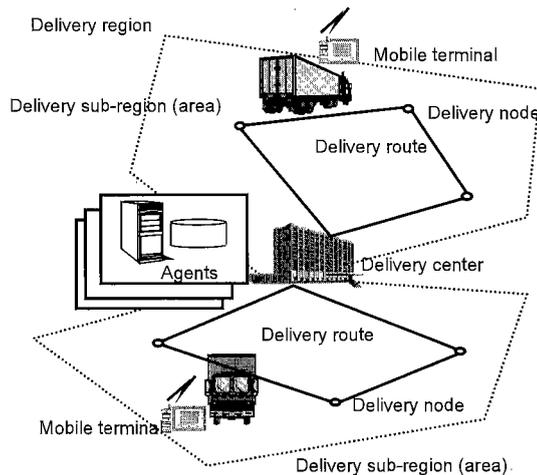


Fig. 1. Typical model of a truck delivery scheduling system.

- Services for delivery of merchandise to customers from a department store delivery center,
- Post office parcel delivery operations,
- Beverage delivery to bars and restaurants,
- Currency delivery and scheduling at ATM machines,
- Internet-based home grocery delivery,
- Wholesale distribution from warehouses to retailers.

In creating a delivery schedule, consideration must be paid simultaneously to numerous constraints:

- The upper limit to the number of available personnel,
- The experience and skill of personnel (vehicles they are able to operate, time required for inspection of goods, etc),
- The upper limit to the number of trucks available for use,
- Constraints on truck loads (volume, weight),
- Time window specified for delivery to a given delivery node,
- Special conditions in effect along the roads on the routes.

For actual real-scale problems, this becomes a problem of optimizing a large-scale combination, making reduction to a formula difficult and a rigorous solution impractical.

2.2 Continuous Re-Scheduling and Execution Problem

Continuous re-scheduling is performed corresponding to dynamic changes in the real world during the plan execution (when deliveries are being executed). We have reported some of the results of the reactive scheduling method for the conditional changes during the execution of the delivery⁽⁷⁾. However, the case of dynamic arrival of additional orders still remains as an issue to be fully addressed in the course of efficient execution of a delivery operation. We propose a new method employing a negotiation process between the delivery center and trucks on the road to achieve more adaptive real-time re-scheduling procedure considering the current status of delivery operation.

The domain of truck delivery scheduling could be basically modeled as the vehicle routing problem with time window (VRPTW)⁽⁹⁾.

- (1) Input data
 - Delivery center : The single delivery depot (denoted as c)
 - Delivery nodes : Geographical location of customers ($node-i$)
 - Routes : Geographical connection of delivery nodes (and the center) ($node-i, node-j$)

- Travel distance (time) : Distance between delivery nodes ($distance (node-i, node-j)$)
 - Delivery order : Order for a node with time window ($order (node-i, volume, earliest time for delivery, deadline for delivery)$)
 - Delivery truck : A truck has a limited capacity ($truck (maximum load volume, cost per time)$)
 - Sub-region : Division of the delivery region into sub-regions ($sub-region A (node-i, node-j, \dots, node-n)$)
- (2) Output data
 - Delivery route : A path that starts/ends at the center and visits delivery nodes ($(c, node-i, node-j, \dots, c)$)
 - Trip timetable for each delivery route: Including the vehicle/personnel used, table of departure and arrival time at each node, and table of items for delivery orders
 - Loading table : Listing the quantities and order of items for loading
 - (3) Evaluation Criteria
 - Evaluation value (delivery cost) for a route is simply assumed the route mileage of the vehicle
 - (4) Re-scheduling for additional orders
 - Using the initial delivery schedule, trucks loaded with the initial orders are on the road when a new order $o-i = order (node-i, vol-i, t-1, t-2)$ arrives at time $t-i$ at the center. We assume that additional orders are small in number (less than 10% of the total) and the initially assigned trucks could deliver the incoming orders. The problem is that the re-scheduling should find the new plan with the minimum cost increase within the reasonable timeframe, which allows the real-time delivery operation goes on.

3. Solving Truck Delivery Scheduling Problem With Multiagents

3.1 Structure of Multiagents in Problem Solving

Because practical-scale problems involve optimization of large-scale combinations, it is extremely difficult to formulate such problems or to find an exact solution. We consider cooperative problem solving with multiagents (Fig. 2).

We propose a framework for cooperation between agents at the delivery center and agents on the trucks' mobile terminals (truck agents). Some of the main agents composing the delivery scheduling system are as follows;

- (1) Problem-division agent: Divides the problem into sub-problems. Specifically, divides tasks into delivery sub-regions according to the delivery order,
- (2) Integration and evaluation agent: Coordinates sub-problem solutions, and generates an overall schedule free of inconsistencies,
- (3) Sub-problem solving agents: Generate delivery schedules for sub-problems, and calculate evaluation parameters. One such agent is generated for each delivery sub-region,
- (4) Resource management agent: Manages overall resources (personnel, vehicles), performs resource allocation,
- (5) Re-scheduling agent: Manages interfaces with dynamic changes of the environmental conditions in the real world during delivery execution,
- (6) Truck agents: Communicate with the delivery center and control information to/from the delivery personnel and know the current status of the delivery operation.

We describe the required functions of the components for each

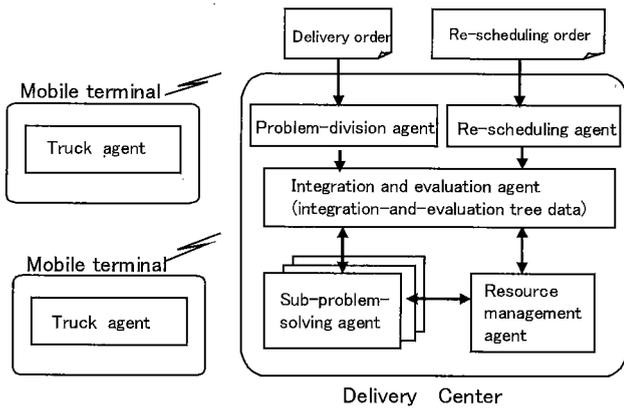


Fig. 2. Structure of multiagents in problem solving.

of these.

(1) Center function

At the center, cooperative problem solving is conducted by making adjustments among the requests from multiple terminals. In a delivery problem, the status of execution and other information from terminals is included for tracking vehicle activity, adjustments are made by the center, and new execution instructions are dispatched. Functions (multicast/broadcast functions, etc.) are also necessary for sharing information, which changes with time (road conditions, for instance) with all terminals as necessary.

(2) Terminal function

The terminal function includes the following items;

- allows the driver to report the current status and activity, e.g. on-site, in-service, out-of-service with time stamping,
- provides GPS (Global Positioning System) function, and the geographical co-ordinates of the vehicle can be transmitted to the center,
- receives geographical information (route map, road condition information, etc.) with updated delivery instructions from the center,
- provides auxiliary peripherals, e.g. bar code scanner, external keyboard or a mobile printer as needed.

3.2 Basic Approach to Problem-Solving Using Distributed Cooperation

The basic approach we have proposed for solving truck delivery scheduling problem is as follows. Readers may refer (7) for more detailed discussion.

(1) Division into sub-problems and its problem solving

The system is based on delivery region division at the global level using existing routes called "a priori route" and route improvement at the sub-problem level. The division of the delivery region is regarded as division of the problem into sub-problems, and delivery sub-regions are partitioned so as to overlap. By this means, interactions and adjustments between agents are introduced, and the result is a more flexible problem-solving framework. By dividing the delivery region into sub-regions and integrating partial solutions in the sub-regions into the solution of the whole delivery region, the quality of the solution reduces because the generation of the route combination is restricted.

(2) Coordination and evaluation by the integration and evaluation agent

When destination overlap is allowed in dividing the delivery region into sub-regions, the route combinations within each

delivery sub-region by a sub-problem solving agent interfere with each other. The integration and evaluation agent takes this into consideration in generating a data set, called an integration-and-evaluation tree, for use in managing the route combinations without inconsistencies. The evaluation values associated with this integration-and-evaluation tree can be used to choose a route combination.

(3) Local optimal vs. global optimal

The definition of "local optimal solution" in our method is defined as follows,

"The local optimal solution is the solution initially derived by integrating partial solutions of local optimal route combinations by sub-regions and re-scheduled by inserting additional orders to the current local optimal routes inside relevant sub-regions".

And "global optimal solution" is also defined as follows,

"The global optimal solution is the optimal solution initially derived and re-scheduled considering the delivery region as a whole without regional division".

Route combinations are computed within each delivery sub-regions, which are partitioned so as to overlap, the total solution is the integration of mutually affecting local optimal route combinations. Also, dynamically arriving orders are tried to insert to the current local optimal routes inside relevant sub-regions.

In the experiment, the deviation of "local optimal solution" from "global optimal solution" is evaluated.

(4) Re-scheduling

Re-scheduling is performed corresponding to the changes in the real world during the delivery execution. Communication functions between the center and the mobile terminals on the delivering trucks are crucial for the timely exchange of various information and the execution instructions from the center to achieve the successful plan execution.

3.3 Dynamic Re-Scheduling for Continuous Delivery Execution

An important feature of the system is that trucks actually start executing the delivery schedule when time is up. Additional orders, which arrive after the start of the delivery, have to be picked up at the center and delivered to the customers (delivery nodes) within the time windows. Trucks are important resources and limited in number, we assume that the dispatch of a new truck for the additional orders is not realistic and the trucks (truck agents) in operation try to re-schedule the initial plan to cover additional orders with the minimum cost increase in cooperation with the re-scheduling agent at the center.

3.3.1 Contract Net Protocol (CNP)

We introduce Contract Net protocol (CNP) for the negotiation process by truck agents and the re-scheduling agent to identify the cost (truck travel mileage) minimum assignment of the additional order. Sub-regions of the delivery region are considered here also as discussed in our previous article (7).

CNP is shown in Fig. 3 and the process is described as follows.

- A new order $o = \text{order}(\text{node-}i, \text{vol-}i, t-1, t-2)$ arrives at time $t-0$ at the delivery center. The new order is to deliver goods of volume $\text{vol-}i$ to destination $\text{node-}i$ from the delivery center within the time window (*earliest time for delivery* $t-1$, *deadline for delivery* $t-2$). The new order is immediately announced to truck agents.
- Each truck agent that has received the announcement, checks the possibility to accept the new order considering the current

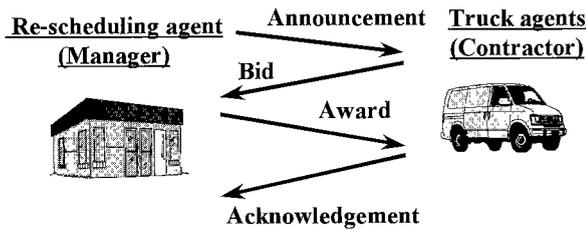


Fig. 3. Contract net protocol.

status of its delivery operation and constraints such as time windows at the delivery nodes, capacity limit of the trucks. Then, if acceptable, *TruckAgent-k* ($1 \leq k \leq N$ where N is the number of trucks) computes a bid (*TruckAgent-k*, o , c) where c is the additional cost for inserting the new order to the current delivery route.

- (c) *TruckAgent-k* sends its bid (*TruckAgent-k*, o , c) to the re-scheduling agent at the delivery center.
- (d) Re-scheduling agent selects the optimal bid and sends the award message to a truck agent as the successful contractor and the reject message to other truck agents.
- (e) The truck agent (successful contractor) sends an acknowledgement of the award.

If the contract is not successful with any truck agent, the delivery center may report to the customer and try to negotiate about possible solutions such as to shift the order to the next day's delivery or to dispatch an emergency truck with higher cost.

3.3.2 Extension of Contract Net Protocol

(1) Hierarchical organization using sub-problem division

The CNP could be organized hierarchically as shown in Fig.4. This organization has the advantage of simplifying the communication protocol and reducing the communication cost for the protocol. Also Sub-problem solving agents could have a chance to consider other factors such as the skill level of drivers, the idle level of trucks etc. in addition to the delivery cost to select a contractor for a job.

Fig.5 shows message flows for the contract process in the hierarchical organization. Re-scheduling agent (a) announces the new order to sub-problem solving agent(s), which cover the sub-region(s) where the new order customer belongs. Here, acceptance-guaranteed protocol⁽¹⁰⁾ could be used for the efficient negotiation. The sub-problem agent(s) (b) announce the new order to trucks in its sub-region(s). Truck agents compute their bids and (c) send them back to the sub-problem agent. Bids from trucks are with commitment duration because of the real-time nature of the negotiation process. (d) Bid(s) are returned to re-scheduling agent again with commitment duration. (e) Award and (h) Acknowledge process could be skipped if the acceptance guaranteed protocol is used between re-scheduling agent and sub-problem solving agent(s).

(2) Time-bound protocol⁽¹⁰⁾

The negotiation process has to be executed during the actual delivery operation by trucks. Because of the real-time nature of the problem domain, the duration allowed for the negotiation process is very restricted. The extension of the CNP to consider the commitment duration of messages (announcement, bid, and award) is crucial.

Firstly, the announcement message is sent with a certain time duration saying that bids have to be returned from truck agents

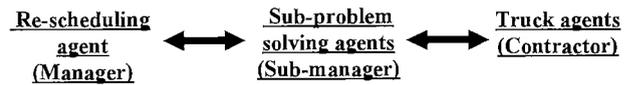


Fig. 4. Hierarchical organization of CNP.

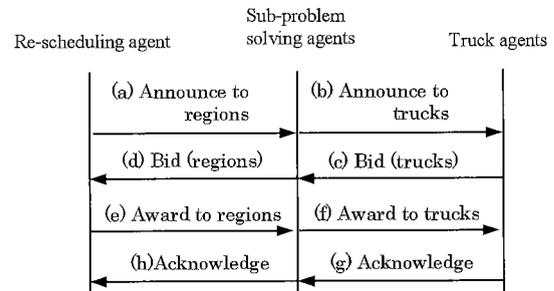


Fig. 5. Message flows of hierarchical organization.

within the time duration.

Secondly, when a truck agent returns a bid, it can only guarantee its bid until the insertion point of the announced new order. In that case, the bid message has to be attached with commitment duration to show that the bid is time-bound.

For the hierarchical organization, re-scheduling agent and sub-problem solving agents are in cooperative coordination, and the acceptance guaranteed protocol simplifies the negotiation process.

3.3.3 Experimental Results

In order to evaluate the procedure proposed in this paper for re-scheduling, we consider Solomon test set⁽¹¹⁾ of the vehicle routing problem with time window. This test-set is a problem set for initial vehicle routing. We assume that at certain time during delivery operation a new delivery order arrives at random node. The delivery cost is the mileage needed by the trucks for the route. Trucks have the equivalent load capacity in a problem. It is assumed that there is a direct path connecting two nodes and travel time between two nodes = distance between two node x 1(unit amount).

(1) Communication load

For the experimental setting, we consider the problem of the following scale from the test set;

- Re-scheduling agent (Manager) 1,
- Sub-problem solving agents (Sub-manager) 5,
- Truck agents (Contractor) max. 25,
- Delivery nodes 100.

To see the improvement of the communication load, we compare our hierarchical organization with non-hierarchical organization of CNP.

(a) Non-hierarchical organization

Re-scheduling agent announces the new order to all the truck agents and capable truck agents send back computed bids. The number of communication messages T is estimated as follows.

$$T = TA \times (1 + \alpha) + 2$$

where TA is the number of truck agents

α is the rate of bidding agents $0 \leq \alpha \leq 1$

In the test set, TA is 15 to 20 and the estimated value of α is 0.35. Then the number of communication messages T is in the range from 22.25 to 29.0. And the bid rejection (by contractor) rate is as high as $1 - 0.35 = 0.65$.

(b) Hierarchical organization

In Fig. 5, the number of communication messages T is estimated as follows.

$$T = SA \times \beta \times (1 + \gamma) + STA \times SA \times \beta \times (1 + \delta) + 4$$

- where SA is the number of sub-problem solving agents
- STA is the number of truck agents in a sub-region
- β is the rate of relevant sub-regions $0 \leq \beta \leq 1$
- γ is the rate of bidding sub-problem solving agents $0 \leq \gamma \leq 1$
- δ is the rate of bidding truck agents $0 \leq \delta \leq 1$

In the test set, $SA=5$, STA is 3 to 5 and the estimated value of $\beta=0.4$, $\gamma=1.0$, $\delta=0.6$. Then the number of communication messages T is in the range from 17.6 to 24.0. And the bid rejection rate is about 0.3.

(2) Computational cost

The most time consuming process of the re-scheduling is the process of searching for the new optimal re-scheduled route with constraints such as time windows at the delivery nodes, capacity limit of the trucks.

Fig. 6 shows computation time for inserting a new order into the initial delivery route optimally where the number of initial orders varies 25, 50, and 100. We used a PC with Pentium III (500MHz) to run our computation. The computation time is roughly $O(N^2)$ where N is the number of initial orders on the route.

Set names R103, C101, and so on from the original Solomon benchmarks have the following meaning. Sets C have clustered orders whose time windows were generated based on a known solution. Problem sets R have orders location generated randomly over a square. Sets of type 1 have narrow time windows and small vehicle capacity. Sets of type 2 have large time windows and large vehicle capacity.

The solutions of type C and type 2 problems have more longer delivery routes and take more time for the computation. The result shows that real-time re-scheduling is feasible for the problems of the practical number of delivery nodes.

(3) Trade-off of computational cost and the optimal delivery solution by division of the region

Fig.7 shows an example of the initial solution of the delivery routes for problem R203 of 25 nodes from the Solomon test set and routes are *Truckagent-1* = (c, 6, 5, 8, 17, 16, 14, 13, c), *Truckagent-2* = (c, 2, 15, 23, 22, 21, 4, 25, 24, 3, 12, c), and *Truckagent-3* = (c, 18, 7, 19, 11, 20, 9, 10, 1, c). A new order at node No.26 is given as *order(No.26, 17, 0, 500)* at time 170.

Although Solomon test set does not have the concept of sub-region, we assume division into sub-regions, area-1, area-2, area-3 as shown in Fig.7.

Fig.8 shows a local optimal solution inside area-2 that is the new order is announced only to *Truckagent-2* in area-2 where the new customer No.26 is located.

The local optimal solution is *Truckagent-2* = (c, 2, 15, 23, 22, 21, c, 26, 4, 25, 24, 3, 12, c), and the cost increase is 33.3 in increase of the distance the truck has to travel to pick up the order at the center and deliver it to node 26 then return to node 4.

If we consider the case where the new order is announced to all the truck agents. Then the global optimal solution is *Truckagent-1* = (c, 26, 6, 5, 8, 17, 16, 14, 13, c), and the cost increase is 20.0. Deviation from the global optimal in this case is $(33.3/20.0) \times 100 = 166.5\%$.

This result indicates that the computational complexity

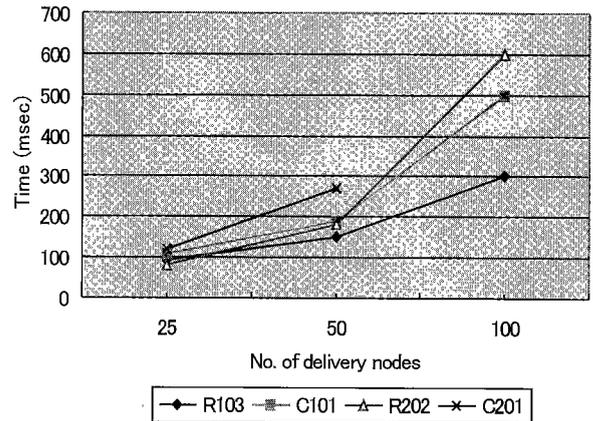


Fig. 6. Process time for inserting a new order.

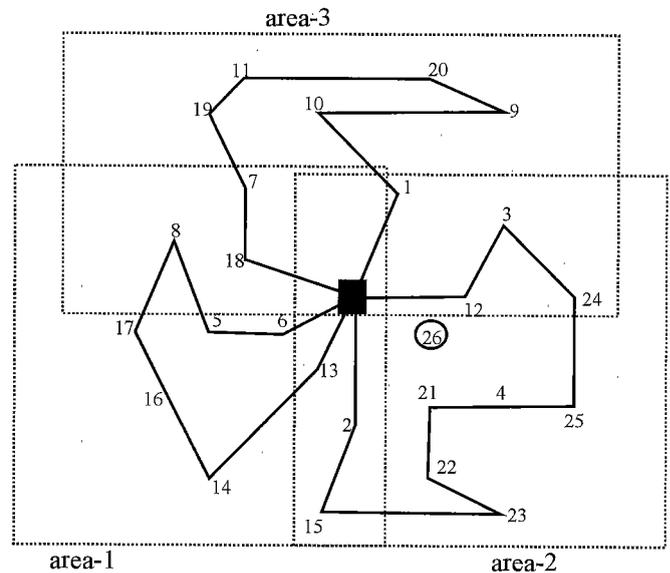


Fig. 7. Problem example of 25 nodes.

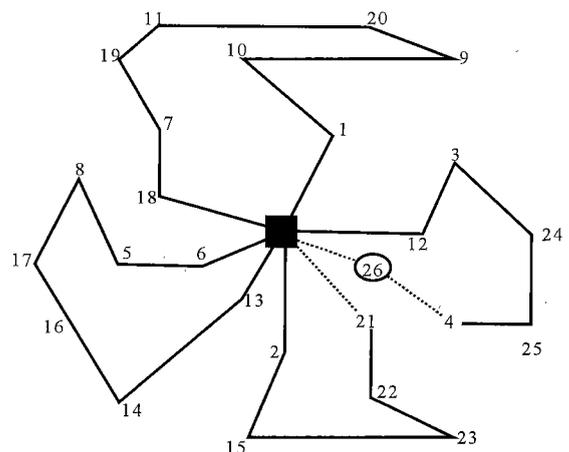


Fig. 8. The local optimal solution inside area-2 for new order at node 26.

increases in the CNP process by announcing all the truck agents; there is a good chance of getting better solution than restricting the announcement only inside the relevant sub-regions. Fig. 9 shows

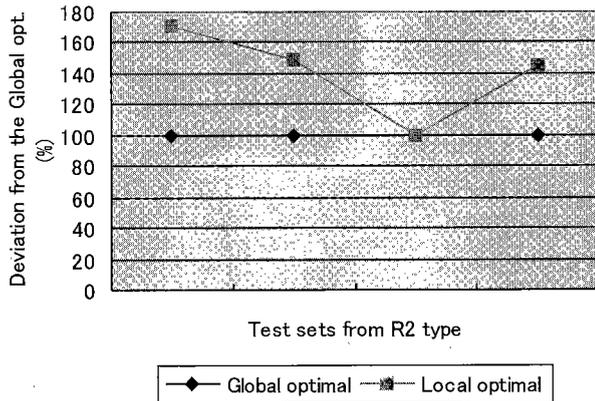


Fig. 9. Samples of the deviation from the global optimal.

some samples of the deviation from the global optimal under the simple overlapping division of the region into sub-regions.

We consider that by creating the sub-regions under certain criteria, there is a trade-off point between the computational cost and the delivery cost.

4. Related work

In this paper, we have applied our delivery scheduling and execution method to the Solomon test set of the vehicle routing problem with time window and discussed its computational cost and optimality of the delivery cost (trucks' travel distance).

Various heuristic algorithms for PDPTW, which are generated from VRPTW of Solomon test set, are reported in ⁽⁵⁾. These algorithms are applicable only where all the input data is known when the planning process is started and they are not able to deal with the problem of dynamic changes during the execution of the delivery. It is reported that resulting deviation from the optimal solution is within about 50% for problems of R2 type. Our work deals with dynamic cases and although direct reference doesn't hold, deviation within 70% (see Fig.9) could be viewed reasonable.

The MARS system ⁽⁶⁾ uses an auction mechanism of delivery orders among shipping companies and we assume that the MARS system has some difficulty due to computational time in case of increasing number of delivery orders. In our work, the method is based on the division of the delivery region into sub-regions to alleviate the same difficulty of computational time. A sub-problem agent is assigned to each sub-region and our hierarchical framework of "delivery center - delivery sub-regions - delivery trucks" can solve the real-time re-scheduling problem efficiently.

5. Discussion

We evaluate and analyze the delivery re-scheduling method proposed in this paper and consider problems remaining to be further investigated.

(1) Relation to previous work

In the previous work ⁽⁷⁾, we employed bottom-up (forward) inference mechanism in the problem-solving agent for each delivery sub-region and to deal with the changing environmental conditions at the time of actual delivery execution, we considered hypothetical reasoning framework. Conditions such as the traffic conditions of roads were incorporated as the defaults knowledge in the system and the initial delivery schedule was modified

according to the changes of the knowledge. However, this re-scheduling method had a shortcoming of not well adapting to changes, which are not incorporated as the default knowledge at the time of initial delivery scheduling.

In this paper, we propose a new method employing an extended CNP by cooperative re-scheduling agent at the delivery center and truck agents to achieve more adaptive real-time re-scheduling procedure and to deal with randomly arriving unpredictable orders. Our work shows that the proposed extended CNP framework provides a very flexible re-scheduling process such as hierarchically organized work process for a new delivery plan and time-bound process for negotiation.

We evaluate the proposed method with computer simulation applying to a standard benchmark test-set. It is shown that real-time re-scheduling is feasible in terms of computational time for the situation of the practical number of delivery nodes. For the optimality of the delivery cost, it is shown that the local optimal solution, in case of the division into sub-regions, is practical compared to the global optimal solution without any sub-region division.

(2) Efficient negotiation process

Here, time-bound negotiation process among multiagents is proposed for the negotiation. We expect that the process could be used for automated negotiation framework. We need to investigate the strategy and to experiment more to identify the best system architecture for this particular domain of the delivery re-scheduling.

(3) Division of the region into sub-regions

By dividing the region into delivery sub-regions, and allowing overlapping between neighboring sub-regions, we have discussed the trade-off of computational cost and the optimal delivery cost. Further study of the optimal degree of overlap between sub-regions will be necessary.

Computation time for inserting a new order into the initial delivery route is roughly $O(n^2)$. And this is the most time consuming process of the re-scheduling.

(4) Event driven approach in the re-scheduling

In the truck delivery scheduling, we consider the occurrence of event, which requires the execution of re-scheduling is moderate and the real time event-driven approach as proposed in this paper is feasible.

On the other hand, information gathering and re-scheduling with a certain time interval increases the cost of operation but could produce more optimal result of delivery by adjusting the delivery operation against gradual deviation from the original schedule.

6. Conclusion

In this paper we have proposed a cooperative multiagent system and applied to the Solomon test set of the vehicle routing problem with time window. Additional delivery orders, which arrive dynamically during the execution of the delivery according to the initial delivery plan, are considered. Multiagents for the delivery center and delivery trucks cooperate to cope with the re-scheduling problem. We show that the proposed extended CNP framework provides a flexible negotiation process for the dynamic re-scheduling problem.

(Manuscript received July 30, 2002, revised August 25, 2003)

References

- (1) L.M. Gambardella, E.Taillar, and G. Agazzi : "MACS-VRPTW:A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows", Technical Report IDSIA-06-99, IDSIA, Lugano, Switzerland (1999)
- (2) K. Uchimura and K. Kanki : "A Method for Finding Round Trip Routes on Road Network Data", *IEEJ Trans. EIS*, Vol.114, No.4, pp.456-461 (1994)
- (3) H. Niwa, Y. Yoshida, and T. Fukumura : "Path Finding Algorithms Based on the Hierarchical Representation of a Road Map and Its Application to a Map Information System", *Transact. of Information Proc. Society of Japan*, Vol.31, No.5, pp.659-665 (1990)
- (4) D.E Wilkins and M. desJardins : "A Call for Knowledge-based Planning", *AI Magazine*, Spring 2001, Vol.22, No.1, pp.99-115 (2001)
- (5) H.C. Lau and Z. Liang : "Pickup and Delivery Problem with Time Windows, Algorithms and Test Case Generation", Proc. of IEEE Conference on Tools with Artificial Intelligence (ICTAI) (2001)
- (6) K. Fischer, J.P. Müller, and M. Pischel : "A Model for Cooperative Transportation Scheduling", Proc. of the 1st International Conference on Multiagent Systems (ICMAS'95), San Francisco (1995)
- (7) J. Sawamoto, Y.Ohta, H. Tsuji, and H. Koizumi : "A Proposal of a Method of Reactive Scheduling for the Delivery Scheduling Problem", *IEEJ Trans. EIS*, Vol.122, No.5, pp.832-842 (2002)
- (8) K.L. Myers : "Towards a Framework for Continuous Planning and Execution", Proc. of the AAAI Fall Symposium on Distributed Continuous Planning (1998)
- (9) J. Larsen : "Parallelization of the vehicle routing problem with time window", Ph. D. Thesis, IMM, Technical University of Denmark (1999)
- (10) K.J. Lee, Y.S. Chang, and J.K. Lee : "Time-Bound Negotiation Framework for Electronic Commerce Agents", *Decision Support Systems*, Vol.28, pp.319-331 (2000)
- (11) <http://www.idsia.ch/~luca/macsvrptw/problems/welcome.htm>

Jun Sawamoto



(Member) received the B.E. and M.E. in mechanical engineering from Kyoto University in 1973 and 1975. He joined Mitsubishi Electric in 1975, He was assigned to the Institute for New Generation Computer Technology between 1985 and 1988. He worked on research and development of knowledge data processing systems, system engineering technologies, multimedia application systems etc. Currently, he is assigned to the Space Solution Systems Development Division. His research interests are agent-based problem solving, cooperative problem solving, Web services and ITS, He is a member of IEEJ, IPSJ.