DYNAMIC PLANNING OF CARGO TRANSPORTATION FOR UNCERTAIN ENVIRONMENTS

Yukihisa Fujita [(a)]

[(a)] Hitachi, Ltd. Japan

[(a)] yukihisa.fujita.hg@hitachi.com

ABSTRACT
Logistics is one of the mission-critical issues for the oil & gas business. Support logistics, which deals with supplying cargoes from ports to destinations, has a major economic impact on business, and transportation plays an important role in it. One of the important issues in support logistics in the oil & gas area is to mitigate the impact of uncertain factors such as bad weather. In this study, we describe a method of planning transportation for dynamic cargo to mitigate the impact. Our method uses an agent-based trade mechanism and changes the current plan whenever the situation changes. The evaluation results show that our method can make plans faster and more efficiently than weight-based simple planning. Additionally, we clarified that it can be applied to large-scale environments.

Keywords: cargo transportation, agent-based trade, uncertainty, planning method

1. INTRODUCTION

Logistics is one of the mission-critical issues for various businesses including oil & gas. In the oil & gas business, support logistics, which deals with supplying goods for operations, has become the main problem due to the expansion in oil & gas development. For example, recent explorations have found large oil fields located deeper than conventional ones. 'Deepwater' fields have been found in Brazil, West Africa, etc. (Guzman, Carvajal, and Thuriaux-Alemán 2013). Although such fields provide an attractive quantity of crude oil, the expansion makes the support logistics more difficult. Deepwater fields are located hundreds of kilometres from shore, meaning that much more time and money are required for support logistics than with conventional fields. Therefore, logistics performance has to be improved to expand oil & gas development.

Logistics performance can be improved in several ways. For example, Leonard and Voß (2014) proposed a cloud-based IT system to improve port operation efficiency. The key factor of the system is real-time visibility of port operations using Internet of things (IoT) technology. Parreño, Alvarez-Valdés, Oliveira, and Tamarit (2010) created a solving algorithm for container loading problems, one which optimises container location to minimise empty space. They clarified that their algorithm can solve the problem with a large data set. From the view of monetary cost, Tseng, Yue, and Taylor (2005) mentioned transportation plays an important role because around one third to two thirds of enterprises' logistics costs are spent on transportation. As for support logistics in the oil & gas business, transportation is also important because they need frequent supplies to ensure stable operations. Therefore, we focused on transportation optimisation in this study.

Transportation is typically optimised by using operations research techniques. The transportation activities are modelled as mathematical formulations: an objective function and constraint equations. The objective function represents maximising the performance, including cost minimisation, stock-out avoidance, etc. The constraint equations represent conditions such as the number of transportation vehicles and the vehicles' speed. This kind of logistics is typically regarded as a pick-up and delivery problem (PDP) or a pick-up and delivery problem with a time window (PDPTW). Hence, one approach for improving logistics performance is to solve such problems by considering the appropriate cost factors. For example, Romero, Sheremetov, and Soriano (2007) proposed a heuristic approach for the problem, focusing on offshore transportation by helicopters. Korsvik and Fagerholt (2010) proposed a vessle scheduling and routing method for the actual oil & gas support logistics. Their method is based on integration of machine learning and a PDPTW solver. We can use these techniques to make optimised transportation plans. However, transportation is disrupted by uncertain factors like weather changes, and transportation plans should be changed to minimise the effect of these uncertain factors.

Changing plans is crucial for transportation in support logistics because the influence of uncertainty is larger than in other area, like inter-continental transportation. In this study, we designed a dynamic transportation planning method for uncertain environments. The method changes the current plan when the situations are changed to mitigate the impact of the uncertainty. It uses an agent-based trading mechanism. In this paper, we show the preliminary performance evaluation results and discuss the scalability to apply the method to large-scale cases. The remainder of this paper consists of the following. Section 2 defines the problem in detail.

Section 3 explains related work from the viewpoints of operational research and an agent-based trading mechanism. Section 4 describes our approach: transportation planning based on the agent-based trading mechanism. Section 5 shows the evaluation results. Lastly, Section 6 concludes the paper.

2. PROBLEM DESCRIPTION

2.1. Overview

Figure 1 shows an overview of the problem setting used in this paper. As described in Figure 1, we focus on offshore transportation with three elements: offshore platforms, vessels, and ports. Offshore platforms demand supplies required for their operations within a certain period. To satisfy their demands, the supplies stored in cargoes are transported from onshore ports to offshore platforms using vehicles called platform supply vessels (PSVs) (Aas, Halskau, and Wallace 2009). The number of vessels is fixed, and each vessel has the fuel and capacity for storing cargoes. Transportation takes hours depending on the vessels' speed. The problem requires us to make a consistent transportation plan including cargo assignments and vessel routing.

In this paper, we only focus on transportation efficiency, i.e., the number of transported cargoes within a certain period. This means we do not consider cost factors such as fuel fees, purchase costs, and loss of production caused by delays. Additionally, we do not consider time window which is consider by PDPTW. These assumptions make the problem simple and enable us to evaluate the effects of our method easily.

2.2. Supply Cargoes

In this problem, cargoes are composed of supplies for transportation and are prepared at specific ports when they are in demand. Each cargo has a source, destination, and weight. The source is a specific port which prepares the cargo, and the destination is the specific platform which has demand for it. The weight is used for calculating the loading/unloading time and limiting the number of cargoes loaded on vessels.

2.3. Ports and Platforms

In this problem, ports and platforms have the same features. The only difference is whether they are the source or destination of the transportation. They have the number of berths and loading/unloading performance as features. The number of berths limits the number of vessels loading or unloading cargoes at the location simultaneously. The loading/unloading performance defines the time required for loading/unloading cargoes to/from vessels.

2.4. Goal

The goal of this problem is to make an efficient transportation plan, which consists of cargo assignments and vessel routing. Efficiency is typically measured using several criteria. From the viewpoint of monetary cost, the criteria are fuel consumption and the number of vessels used for transportation. In this paper, as noted earlier, we adopted the number of transported cargoes as the evaluation criteria.

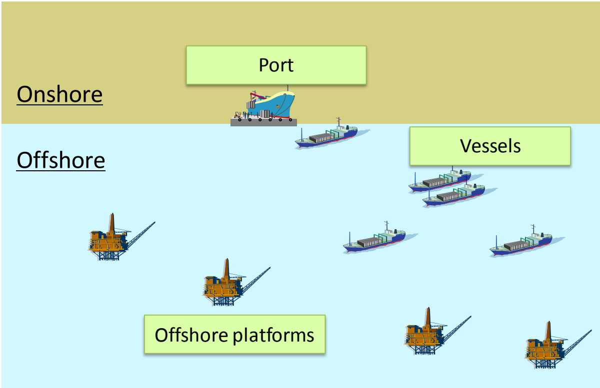

Figure 1 Problem setting overview.

3. RELATED WORK

The most similar problem of transportation planning for uncertain environments is called Dynamic PDPTW. Mitrović-Minić, Krishnamurti, and Laporte (2004) adopted a heuristic approach to make routing and scheduling plans dynamically. They clarified their approach can make near optimal plans in comparison with iterating a static PDPTW solver. Pureza and Laporte (2008) proposed a waiting and buffering strategy for future situation changes. They clarified the strategy can improve the quality of the solution.

However, these approaches typically cannot change constraints. For example, when some routes are closed by bad weather, we have to introduce new constraints representing road closures. To introduce the constraints, we can adopt another technique like constraint programming (Gavanelli and Rossi 2010). Constraint programming can change constraints anytime and output results. Elkhyari, Guéret, and Jussien (2004) applied a constraint programming technique to dynamic train scheduling and clarified it can improve computation time and stability of output schedules. While these approaches can make constraints flexibility, they sometimes require much computation time when solution spaces are large.

Our approach focuses on realising high-speed and flexible semi-optimisation under uncertain environments. We adopted an agent-based trading mechanism for making efficient plans. Using the agent-based approach, Koźlak, Créput, Hilaire, and Koukam (2006) applied a multi-agent mechanism to the PDPTW solver. Their approach is bottom-up, meaning each vehicle makes its plan while considering the estimated future demand. Our approach is another bottom-up approach which uses a trade mechanism to change the assignments of cargoes. Our approach can change plans without estimating future demand.

4. APPROACH

4.1. Overview

Our approach is based on agent-based trading which involves exchanging goods between agents. We regard vessels as trading agents. Each agent is assigned cargoes which should be transported from the source port to the destination platforms, and it has its own plan and cost for cargo transportation. The cost is determined using cargo attributes (e.g., weight and loading location) and transporting routes (e.g., the number of transport locations and the total distance). To make an efficient transportation plan, the agents trade their assigned cargoes with the other agents to minimise their cost. Additionally, our method can manage constraint changes by re-calculating whenever constraints change. Figure 2 shows the system containing our method. It consists of two modules: a planner and simulator. The planner does its work using the agent-based trade mechanism. The simulator modifies the plans to solve the constraints and to evaluate the plans by several criteria such as fuel consumption, monetary cost, and lead time. Additionally, the simulator has an event controller which generates weather changes—among other things—and new requests of cargo transportation. Once the events occur, the simulator provides the current state to the planner, and it modifies the plan. We can make and modify transportation plans which have no inconsistency under uncertain environments by using the system.

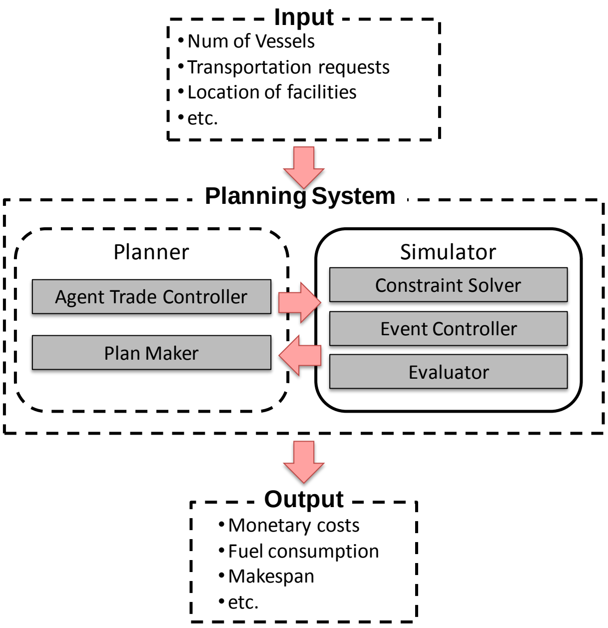

Figure 2 Overview of planning system containing our method.

4.2. Trade Algorithm

Our trade algorithm consists of the following steps.

1. Calculate the current cost.
2. Find cargoes which reduce the cost the most between all the agents.
3. Select the agent which has cargoes found in step 2 as the trade source agent.
4. Find a trade target agent which can reduce the total cost of both agents.
5. Execute the trade.

These steps are iterated until no cargoes are found in step 3, i.e., there are no tradable cargoes.

```
randomAssign()
repeat
    src ← findTradeSourceAgent()
    dest ← null
    max ← 0
    for all the agents except for src do
        profit ← calculateTradeProfit(src, agent)
        if (profit > max) then
            dest ← agent
            max ← profit
        end if
    end for

    if dest ≠ null then
        trade(src, dest)
    end if
until dest = null
```

Figure 3 Pseudo code of our trade algorithm.

Figure 3 shows the pseudo code of our trade algorithm. Function *randomAssign* assigns unassigned cargoes to agents randomly. The sum of the cargo's weight must be less than the max capacity of vessels. Then, the function *findTradeSourceAgent* finds one agent as a trade source using the following steps.

1. Select one agent.
2. Group cargoes assigned to a selected agent by source and destination.
3. Estimate cost reduction when one of the groups is removed from an assignment.
4. Go to step 1 if there are unselected agents.
5. Select one agent which has the max reduction cost.

The trading cargoes are also decided by these steps. The group of cargoes which has the max cost reduction in step 3 are the trading cargoes. Next, the function *calculateTradeProfit* finds the target trade agent by calculating the trading result for each agent. The function calculates cost reductions if two agents exchange a group of cargoes or pass the group from one to the other. Lastly, the function *trade* executes the trade. This process is repeated until it cannot find tradable cargoes, meaning it cannot reduce costs further for any trades.

4.3. Cost Function

The cost function is as essential a factor for our approach as it is for other operational research techniques. As we noted earlier, we use the number of transported cargoes in a certain period as the evaluation criteria. Hence, the higher the number of cargo vessels transporting in one voyage, the lower the costs should be. To represent such relationships, we define our cost function using the following viewpoints:

1. Maximise the weight of cargoes transported in one voyage.
2. Minimise the number of loading/unloading locations.
3. Minimise travel distance.

From these viewpoints, the cost function C of agent v_i is defined by the following equation.

$$C(v_i) = W(v_i) + L(v_i) + D(v_i) \tag{1}$$

$W(v_i)$, $L(v_i)$, *and* $D(v_i)$ are the costs for the weight, location, and distance, respectively.
Weight cost $W(v_i)$ is defined as follows.

$$W(v_i) = 1 + \frac{W_A}{1 + \exp(W_B \times AS(v_i) - W_C \times MC(v_i))} \tag{2}$$

W_A, W_B, and W_C are coefficient values, and function $AS(v_i)$ and $MC(v_i)$ are the available loading space for the current assignment and the max loading capacity of agent v_i, respectively. The weight cost formed as a sigmoid function facilitates loading a certain quantity of cargoes.
The location cost $L(v_i)$ is defined as follows:

$$L(v_i) = ((numPort(v_i)) - 1) \times L_A + 1) \times numPF(v_i) \tag{3}$$

L_A is the penalty coefficient, and $numPort(v_i)$ and $numPF(v_i)$ are the number of ports and platforms stopped in the current voyage. The penalty coefficient limits the number of destination ports because cargoes typically should be transported from one port to a few platforms. However, the number of destination platforms is not limited but increases the cost for aggregating cargoes of the same destination.
Distance cost $D(v_i)$ is defined as follows.

$$D(v_i) = D_A \times TD(v_i) \tag{4}$$

D_A is the coefficient, and $TD(v_i)$ is the total travel distance. This function represents a short voyage as being better than a long one.

4.4. Plan Updates

When a plan should be changed for reasons such as new demand, the current plan is updated. The update does the trading process again by considering the following constraints:

1. Loaded cargoes have to be unloaded only at destination platforms.
2. Assigned cargoes which have not been loaded are tradable.

To re-execute the trading process, all the costs are re-calculated. By calculating cost every time the situation changes, we can consider various events such as new demand and constraint changes.

5. EXPERIMENTS

5.1. Evaluation in Certain Environments

5.1.1. Settings

First, we evaluated our method in certain environments. In these environments, all the constraints and transportation demands were given during the initial planning. For the evaluation, we used the challenge problem proposed at ICKEPS2012 (Igreja, Silva, and Tonidandel 2012), which is the transportation planning problem for offshore oil production. The problem requires satisfying constraints (e.g., fuel, the cargo capacity of transporting vessels, and the number of berths of ports) and efficient plans for given demands.

Table 1 Value settings.

Name	Value
Number of vessels	10
Number of ports	2
Number of platforms	10
Number of cargoes	15
W_A	40
W_B	0.1
W_V	0.55
L_A	500
D_A	0.005

Table 1 shows the major value settings, and the other settings were the same as those in the challenge problem. We evaluated fuel consumption, time required to transport all the given cargoes, and the computation time (hereinafter referred to as the CPU time). As a comparison, we prepared the weight-based simple assigning method which assigns all the cargoes in ascending order of weight. For this experiment, we used a server which has an Intel® Xeon® X5675 with a 3.06-GHz CPU and 24 GBs of memory. All the programs were written in Java and run on OpenJDK 1.7.0u25.

5.1.2. Results

Table 2 shows the results for a static environment. Column 'answer' is one of the optimised results based on the result for the challenge problem (Toropila, Dvořák, Trunda, Hanes, and Barták 2012). The CPU time of 'answer' is the reference value estimated from their paper. The fuel consumption and required time of our method is intermediate between 'simple' and 'answer'. This means that our method makes a plan which is not optimal but nonetheless efficient. Additionally, our method made plans in less than a second while 'answer' required hundreds of seconds. From this result, we can say that our method can quickly make plans which are not optimised but efficient.

Table 2 Results for static environment. Answer is the optimised results. Fuel is the fuel consumption, time is the time required for transportation, and CPU is the computation time. CPU time of 'answer' is the estimated value from the paper.

Method	Fuel [L]	Time [hr]	CPU [s]
Ours	929	218.7	0.93
Simple	1188	222.6	0.11
Answer	887	203.5	600

5.2. Evaluation in Uncertain Environments

5.2.1. Settings

Next, we evaluated our method under uncertain environments which have unknown future demands. Almost all of the settings were the same as those in the previous evaluation. In this experiment, new cargo demands were generated in a certain period. The source and destination were decided randomly upon generation. Hence, we can regard the unpredicted demand as uncertainty.

Table 3 shows the settings of this experiment. The simulation length was 90 days. During the simulation, new demands appeared every generation period. In this experiment, we used the number of transportations for the evaluation. We averaged 50 results for each case.

Table 3 Settings for dynamic planning.

Name	Value
Simulation length	90 days
Generation period	1.5, 3.0, 4.5 days
Number of new demands	3, 5

5.2.2. Results

Figure 4 and Figure 5 show the number of transported cargoes for each generation period when the number of new demands was 3 and 5, respectively. From the results, we clarified that our method can make transportation plans more efficiently than simple plans in every case.

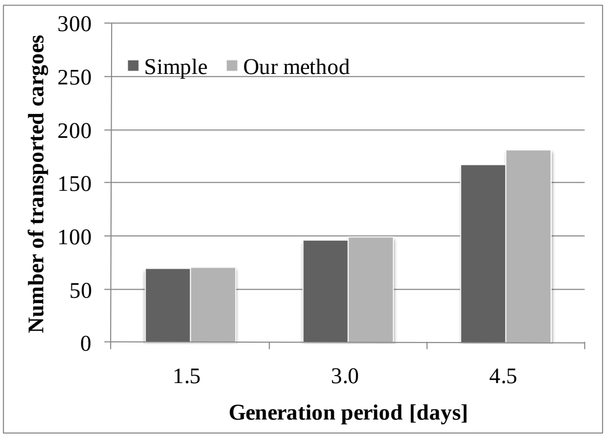

Figure 4 Number of transported cargoes for each generation period (number of new demands is 3).

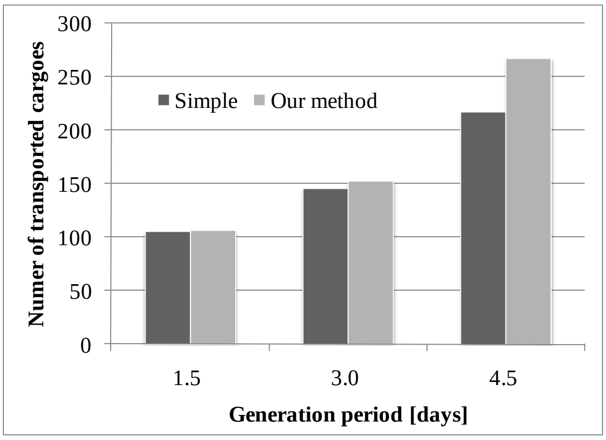

Figure 5 Number of transported cargoes for each generation period (number of new demands is 5).

Additionally, Figure 6 shows the CPU time for each case. From Figure 6, we found that the CPU time was less than 20 seconds in all cases. It is fast enough for periods when emergency response requires rescheduling within minutes.

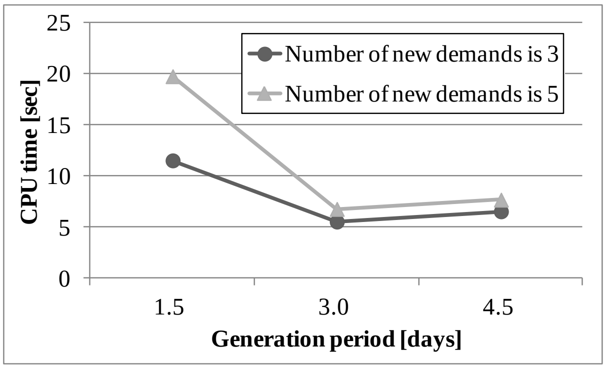

Figure 6 CPU time of our method for each generation period and number of new demands.

5.3. Performance Analysis

5.3.1. Setting

The settings of our previous experiment were for small cases, i.e., only 10 platforms. To apply our method to large cases, we evaluated the CPU time for them. In this evaluation, the generation period was 3, and the number of new demands was half the number of platforms. We averaged 20 results for each case.

5.3.2. Results

Figure 7 shows the CPU time for each case. From Figure 7, we clarified that all the cases were less than 5 minutes. We also found the CPU time increased drastically when the number of vessels was 10. This means there are bottlenecks in specific cases.

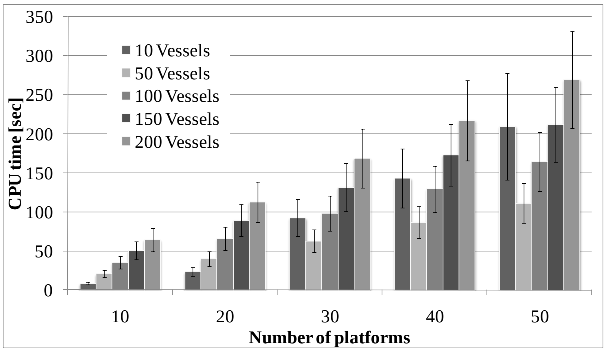

Figure 7 CPU time for large scale.

Figure 8 shows the relationship between the number of new demands and the CPU time when the number of vessels and platforms was 50. We found that the CPU time increased slightly when the number of new demands was less than the number of vessels. When the number of new demands exceeded the number of vessels, the CPU time increased drastically.

One of the causes leading to such drastic increases was calculating the cost function. In this implementation, we calculated the shortest path length when calculating the vessels' routes. The calculation was executed for every cost calculation. However, we could not store the calculation results; storage requires large memory space because the number of combinations is enormous. When we store only the path for 5 destinations under 50 platforms, the number of routes is ${}_{50}P_5 \simeq 2.0 \times 10^8$. In other words, we need 1.6 GBytes if we store each route with 8 bytes. Therefore, to avoid such a drastic increase, we need either machines with large memory or memory saving implementations such as storing frequently used routes.

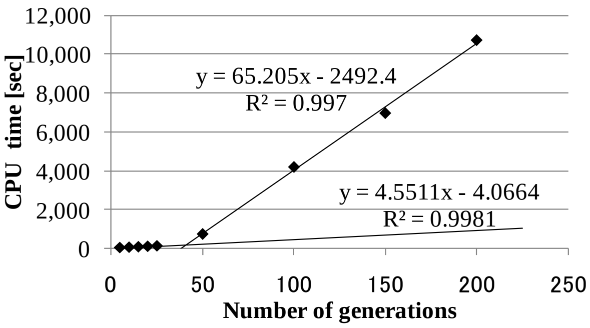

Figure 8 Relationship between number of generations and CPU time when the number of vessels and platforms are 50.

6. CONCLUDING REMARKS

We designed a high-speed transportation planning algorithm to enable dynamic planning of cargo transportation for uncertain environments. Our method is based on an agent-based trading mechanism. Experimental results show that our method can quickly make efficient plans under both certain and uncertain environments. However, our bottleneck analysis shows that we need implementation with efficient memory usage.

Our future work involves two tasks: improving memory usage and verifying constraint changes. The former, as noted, will enable handling large scale problems. The latter will enable handling various emergency situations. In this paper, we tested only the cases of changes in demand. However, many cases involve constraint changes such as platform shutdowns and bad weather conditions. Hence, we need to evaluate our algorithm under such environments.

REFERENCES

Aas B., Halskau O., and Wallace S.W., 2009. The Role of Supply Vessels in Offshore Logistics. Maritime Economics & Logistics.

Elkhyari A., Guéret C., and Jussien N., 2004. Constraint Programming for Dynamic Scheduling Problems. Proceedings of International Symposium on Scheduling, 84–89. May 24-26, Hyogo (Japan)

Gavanelli M. and Rossi F., 2010. Constraint logic programming. A 25-year perspective on logic programming, 64–86.

Guzman R., Carvajal P., and Thuriaux-Alemán B., 2013. Opportunities and challenges for global deepwater players—An industry on the search for new growth areas. PRISM, 66–75.

Igreja H., Silva J.R., and Tonidandel F., 2012. ICKEPS 2012 Challenge Domain: Planning Ship Operations on Petroleum Platforms and Ports. Available from: http://icaps12.icaps-conference.org/ickeps/ICKEPS-Petrobrasdomain-v1.pdf [Accessed 16 March 2015].

Korsvik J.E. and Fagerholt K., 2010. A tabu search heuristic for ship routing and scheduling with flexible cargo quantities. Journal of Heuristics, 16 (2), 117–137.

Koźlak J., Créput J.C., Hilaire V., and Koukam A., 2006. Multi-agent approach to dynamic pick-up and delivery problem with uncertain knowledge about future transport demands. Fundamenta Informaticae, 71 (1), 27–36.

Leonard H. and Voß S., 2014. A Cloud-Based SOA for Enhancing Information Exchange and Decision Support in ITT Operations. Proceedings of International Conference on Computational Logistics, 112–131. September 24-26, Valparaiso (Chile).

Mitrović-Minić S., Krishnamurti R., and Laporte G., 2004. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. Transportation Research Part B: Methodological, 38 (8), 669–685.

Parreño F., Alvarez-Valdés R., Oliveira J.F., and Tamarit J.M., 2010. Neighborhood structures for the container loading problem: a VNS implementation. Journal of Heuristics, 16 (1), 1–22.

Pureza V. and Laporte G., 2008. Waiting and buffering strategies for the Dynamic Pickup and Delivery Problem with Time Windows. Information Systems and Operational Research, 46 (3), 165–176.

Romero M., Sheremetov L., and Soriano A., 2007. A genetic algorithm for the pickup and delivery problem: An application to the helicopter offshore transportation. Theoretical advances and applications of fuzzy logic and soft computing, 435–444.

Toropila D., Dvořák F., Trunda O., Hanes M., and Barták R., 2012. Three Approaches to Solve the Petrobras Challenge: Exploiting Planning Techniques for Solving Real-Life Logistics Problems. Proc. of the 24th International Conference on Tools with Artificial Intelligence, 1, 191–198. November 7-9, Athens (Greece).

Tseng Y., Yue W.L., and Taylor M.A.P., 2005. The role of transportation in logistics chain. Proceedings of the Eastern Asia Society for Transportation Studies, 5, 1657–1672.

AUTHORS BIOGRAPHY

Yukihisa Fujita He received his Ph.D. from Nagoya University in Japan in 2010. He is now working as a researcher at the R&D division of Hitachi, Ltd. in Japan. His current research interests are agent-based simulations and optimisation by simulations. He is a member of the IEICE.