KNOWLEDGE OF REAL TIME POSITION OF VEHICLES AND ITS IMPACT ON THE IMPROVEMENT OF INTERMODAL DRAYAGE OPERATIONS

Alejandro Escudero Santana. University of Seville, Spain. aescudero@esi.us.es
Jesús Muñuzuri Sanz. University of Seville, Spain. munuzuri@esi.us.es
Ron Van Duin. Delft University of Technology, Netherland.
Luis Onieva Giménez. University of Seville, Spain.
José Guadix Martín. University of Seville, Spain.

ABSTRACT

The intermodal transport chain can become more efficient by means of a good organization of the drayage movements. Drayage in intermodal container terminals involves the pick up or delivery of containers at customer locations. There are some works on centralised drayage management, but most of them consider the problem only from a static and deterministic perspective. The main objective is normally the assignment of transportation tasks to the different vehicles, often with the presence of time windows. The work we present here considers the knowledge of the vehicles' real-time position, which permanently enables the planner to reassign tasks in case the problem conditions change. This exact knowledge of position of the vehicles is possible thanks to a geographic positioning system by satellite (GPS, Galileo, Glonass). This additional data are used to dynamically improve the solution.

Keywords: Intermodal transport, Drayage, Real time assignment, Stochastic Transit Time

INTRODUCTION

Road transport has been and continues to be prevalent for the on land movement of freight. However, increasing road congestion and the necessity to find more sustainable means of transport have determined different governments to promote inter-modality as an alternative. For inter-modality to become viable for trips that are shorter than 700 km, a cost reduction is necessary. Final road trips or drayage represent 40% of the intermodal transport costs. There is potential to overcome this disadvantage and make intermodal transport more competitive through proper planning of the drayage operation.

Following the path opened by De Meulemeester et al (1997) and Bodin et al (2000), the number of references on centralised drayage management has increased significantly over the last years, but most of them consider the problem only from a static and deterministic perspective. The main objective is normally the assignment of transportation tasks to the different vehicles, often with the presence of time windows (Wang and Regan, 2002). The first part of the work by Cheung and Hang (2003) develops a deterministic model with time windows, which is then solved by means of the discretization of the times, and by incorporating the concept of fictitious tasks for the beginning and the end of the day for the vehicle. Ileri et al (2006) cover a large number of task types, both simple and combined, and of costs involved in drayage operations, and solve the problem with a column generation method. Smilovik (2006) and Francis et al (2007) incorporate flexible tasks where either only either the origin or the destination is precisely known. Caris and Janssens (2009) propose a two phases insertion heuristic to construct an initial solution that is improve with a local search heuristic.

The work considers the knowledge of the vehicles' real-time position, which permanently enables the planner to reassign tasks in case the problem conditions change. This exact knowledge of position of the vehicles is possible thanks to a geographic positioning system by satellite (GPS, Galileo, Glonass). This additional data are used to improve the solution dynamically.

The paper is organized as follows. The section 2 is the problem definition. A description of the real-time optimization is given in section 3. The section 4 presents an insertion heuristic to solve a snapshot of the problem. Numerical experiments are shown in the section 5.

PROBLEM DEFINITION

This paper discusses the Drayage Problem with Time Windows (DPTW) and Stochastic Transit Time. This problem involves the pick up or delivery tasks of containers between terminal and customers. The intermodal terminal is open during a specified time window. All tasks are known in advance, however the stochastic transit times make it impossible to know the exact time of the trips.

The DPTW can be formulated in terms of a VRPTW with full container load. Assuming a homogeneous container type and size, the problem is to find the assignment of tasks to vehicles, in order to minimize the total cost of serving all tasks. All vehicles v have to return to the terminal before the end of their depot windows. A fixed vehicle cost is included to reduce the used fleet size. Each used truck incurs in a fixed cost. Travelling costs are proportional to the distance travelled. Also, as the transit time is stochastic there is possibility to arrive out of the time window. That is why penalty costs are applied. The total cost is the sum of fixed, travelling and penalty costs.

The static DPTW and Stochastic Transit Time is defined on a graph $G^S = (T_0, A^S)$, where T_0 represents the tasks set. Let T^D be the set of delivery tasks, T^P be the set of pick up task and $\{0\}$ the depot. The set A^S are the arcs that connect depot and customer locations. They are characterized by a distance d_{ij} and a transit time τ_{ij}.

$$T_0 = T^D \cup T^P \cup \{0\}$$
$$T^D \cap T^P = \varnothing$$

Then, there are two kinds of tasks, delivery tasks and pick up tasks. Both delivery and pickup tasks must satisfy a time window, as presented in Fig. 1. The delivery tasks involve catching a container in the terminal and delivering it in a customer location. The origin is the terminal and the destination is a customer. The distance between origin and destination is d_i, and its transit time is τ_i. There exist a service time in the origin and other in the destination, s_i^O and s_i^D. The time window is associated to the origin $[E_i^O, L_i^O]$, so the destination time window is open $[E_i^D, L_i^D]$. This time window is hard in its earliest time, but soft in its latest time. It can never start before the arrival of the train or vessel, while if the drayage driver is late the task can still be completed, but a given amount will have to be paid for the time the container spends waiting at the terminal, c_w per minute. The pick up tasks entail to pick up a container in a customer location and carry it to the terminal. The origin is the customer location and the destination is the terminal. The distance between origin and destination and the transit time are d_j and τ_j, respectively. The service time in the origin and destination are s_j^O and s_j^D. The time window is associated to the destination $[E_j^D, L_j^D]$. This time window is hard in its latest time, but can be soft in its earliest time. If the task is completed before the allocated time, the container will be subjected to a waiting cost; however, if the task is completed later than the time windows, the train or vessel is lost, and a penalized cost will be charged.

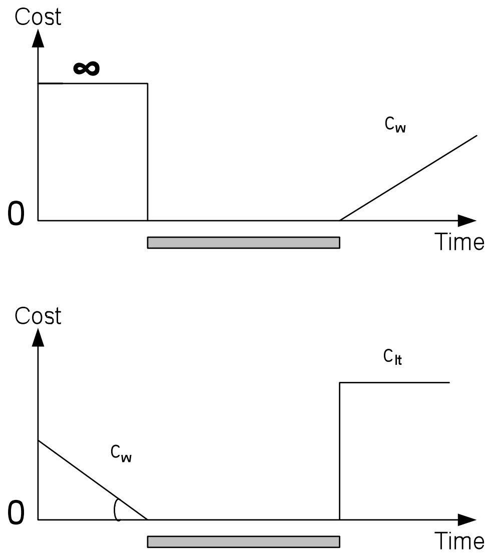

Figure 1 – Time Windows

Since the approach to solve the drayage problem is based on re-optimization, it is necessary to consider that the fleet of vehicle could be anywhere. Fictitious tasks are created to contain every vehicle state, T_f^{ini}. Fictitious tasks are also included to determine the depot window of every truck, T_f^{end}. Then a new graph is defined $G^D = (T, A)$. The set A is defined as the arcs between the destination tasks and the origin of the other tasks.

$$T = T^D \cup T^P \cup T_f^{ini} \cup T_f^{end}$$

REAL-TIME VEHICLE ROUTING

A dynamic methodology will be used to improve the solution at any moment. An optimization algorithm is run at the beginning of the day, and a solution is obtained supposing an expected transit time. As the transit time is a stochastic date, it is impossible to know the exact time to do a task. So, every time that an event happens, the algorithm is run again considering the updated data only for the remaining, pending tasks. There are some options in the re-optimization search:

1. Every fixed time.
2. When a task is accomplished.
3. When a vehicle is delayed of its expected position.

In this work, the re-optimization is done when a task is accomplished. When a task is accomplished, every vehicle state is verified. So, the system will have information about the real position of vehicles. This information lets the system know if a vehicle is FREE, BUSY or ASSIGNED. A truck is busy if it is performing a task in that moment; it is assigned if it is going to the origin of a task; and it is free if it is in the depot or it is going to there. The system has memory about the used and non-used truck until the moment, USEDVEH or NONUSEDVEH. Once vehicles state is verified, the tasks state is updated. Three possible values are possible: PENDING, IN PROCESS and FINISHED.

Then a new optimization is run. As the transit time is stochastic, it is possible that after some time the situation is not the expected one. So, the re-optimization lets the system correct potential delays in the task execution. The newfound solution could change tasks between trucks; however, the busy vehicles must finish the task in process to begin with a new task.

The assignment algorithm is an adaptation of the insertion heuristic of Caris and Janssens (2009) to the real-time drayage problem. Although the transit time is a stochastic variable, the algorithm will use its expected value (t_i, t_j and t_{ij}), so a fast convergence can be got.

A simplified scheme of all the process is presented in the Fig. 2.

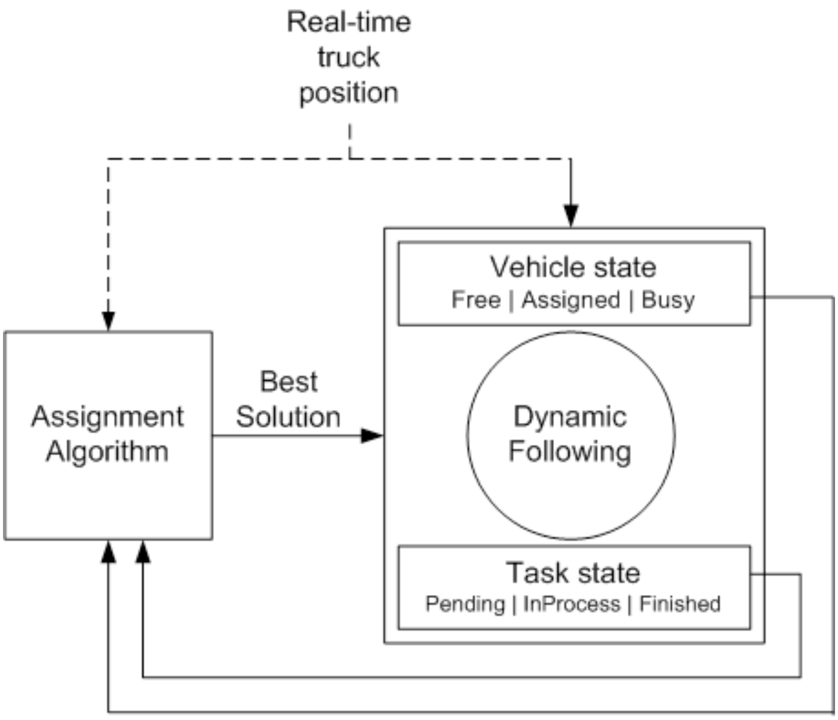

Figure 2 – Dynamism

INSERTION HEURISTIC

In this work, we have adapted the insertion heuristic of Caris and Janssens (2009) to the real-time drayage problem. Although the foundations are similar, there are important differences with our heuristic. These will be commented while the real-time insertion heuristic is explained.

The heuristic procedure is based on the saving of merging single pickup and delivery tasks. If the single tasks are realized individually, we are wasting the 50% of journey in empty trips, as shown in Figure 3a. However, the combination of delivery and pickup tasks may lead to reduction in travel distance, as presented in Figure 3b. As in the real-time drayage problem the assignment of tasks to vehicles is dynamic; the vehicles could be anywhere. As we said before, the position of vehicles would be regarded as fictitious tasks that could be combined with real tasks looking for saving cost. It will mean a saving only if the fictitious tasks are merged with a pickup task (See Figure 3c)

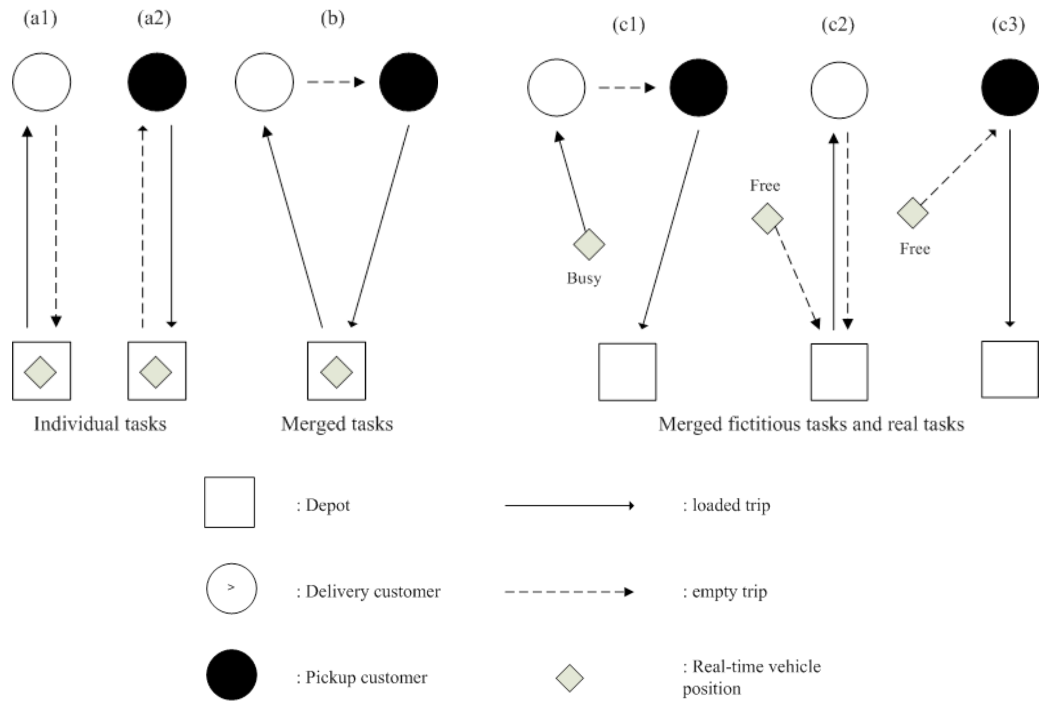

Figure 3 – Merging trips

The insertion heuristic is divided into two phases. In the first one, all the possible unions between tasks (including fictitious tasks) are analyzed. The best merged-tasks are selected. In the second phase, the combined tasks are inserted into routes. The heuristic procedure finds an initial solution, which is later improved by three local searches.

Pairing single tasks

As shown in Fig.1, the combination of two single tasks in merged tasks could mean a saving cost. These pairs will be composed for a first fictitious or delivery task and a second pick up task. Any other combination of tasks entails no saving cost.

It is necessary consider that not all pairs of tasks can be combined into feasible merged tasks. On the one hand, due to the existence of time windows, not all pairs are possible. The second task of the pair must be reached after the realization of the first task. On the other hand, the waiting time to merge two single tasks is limited to a maximum amount $MAXWAIT$. A pair is refused if its minimum waiting time $MINWAIT_{ij}$ is higher than $MAXWAIT$. This limit is founded on inefficient of large waiting time. If a vehicle must wait too much time between the realization of the first task of a pair and the second task of the same pair, much time will be taken up from the working day of this vehicle and it will be necessary to use a lot of vehicles.

These two conditions limit the feasible pairs. Its equations are shown below, the first one is the time windows condition, and the second one is the maximum waiting time condition.

$$\max(time, E_i^O) + s_i^O + t_i + s_i^D + t_{ij} + s_j^O + t_j \le L_j^D$$
$$E_j^D - \left(\max(time, L_i^O) + s_i^O + t_i + s_i^D + t_{ij} + s_j^O + t_j\right) \le MAXWAIT$$
$$MINWAIT_{ij} = \max\left(0, E_j^D - \left(\max(time, L_i^O) + s_i^O + t_i + s_i^D + t_{ij} + s_j^O + t_j\right)\right)$$

Once the feasible pairs are found, these are evaluated and ranked according to savings in travel distance, which is presented in the next expression

$$d_i + d_j - d_{ij}$$

The pair of feasible tasks with the highest value is selected first. All pairs containing some of the tasks of the selected pair will be deleted of the list. The process of pairing tasks up is repeated until no feasible combination exists in the list. The remaining tasks are served in individual trips and form an imaginary pair with a dummy customer.

Route Construction

A route for each vehicle is created. Every vehicle must carry out a route. These routes are composed of pairs of tasks. The first pair of every route is the pair that contains the initial fictitious task related to the vehicle of this route.

The other pairs are assigned sequentially. Vehicles with lower cost are used firstly. Pairs of tasks are selected to be inserted into routes in increasing order of their latest start time L_{ij}, where

$$L_{ij} = \min\left(L_i^O, L_j^D - t_j - s_j^O - t_{ij} - s_i^D - t_i - s_i^O\right)$$

A pair can be inserted into a route v if two conditions are complied. The first condition is that the vehicle v must be able to begin to serve the pair before L_{ij}. The second condition imposes that the vehicle v must return to the terminal within its depot windows.

$$\max(time, RS_v) \le L_{ij}$$
$$\max(time, RS_v, E_i^O) + RS_{ij} \le T_v$$

The route service time RS_v is initially set to $RS_v^{fictitious}$. If a pair can be inserted into a route of a USEDVEH, this is update and its route service time is increased. A new vehicle will be used otherwise. It is possible that no vehicle can serve the pair on time, even the vehicle remaining in the depot. Then, the pair will be assigned to the vehicle that can arrive to the pair of tasks with less delay.

$$RS_v \leftarrow \max(time, RS_v, E_{ij}) + RS_{ij}$$
$$RS_{ij} = s_i^O + t_i + s_i^D + t_{ij} + s_j^O + t_j + s_j^{D}` + MINWAIT_{ij}$$

The earliest starting time E_{ij} is defined as the earliest time a vehicle can start to serve a pair (i, j) without unnecessary waiting between tasks i and j:

$$E_{ij} = \begin{cases} L_i^O & \text{if } E_j^D - t_j - s_j^O - t_{ij} - s_i^D - t_i - s_i^O > L_i^O \\ E_j^D - t_j - s_j^O - t_{ij} - s_i^D - t_i - s_i^O & \text{if } E_i^O \le E_j^D - t_j - s_j^O - t_{ij} - s_i^D - t_i - s_i^O \le L_i^O \\ E_i^O & \text{if } E_j^D - t_j - s_j^O - t_{ij} - s_i^D - t_i - s_i^O \le E_i^O \end{cases}$$

If a route is only composed by a fictitious task after the construction procedure and $RS_v^{fictitious}$ is 0, this means that the route is really unused. The vehicle will be in the depot all time. If the route is only composed by a fictitious task but $RS_v^{fictitious}$ is not 0, then the vehicle is performing a task, has not any other assigned, and will go to the depot.

Local Searches Algorithm

Three local searches are proposed to improve the initial solution obtained by the above heuristic. CROSS, COMBINE and INSERT operator will be described in this section. The sequence in which these operators are implemented is CROSS → COMBINE → INSERT. First, the CROSS operator is applied to find the best combinations of pairs. Then, COMBINE and INSERT operator try to reduce the number of new vehicles to be used.

CROSS operator

Two pairs are selected, (g, h) and (i, j). These pairs are crossed, so two new pairs are obtained, (g, j) and (i, h), and their feasibility is checked. If both are feasible, it is checked if the new pairs can be reinserted into the routes. Then two possible combinations are analyzed: (g, j) inserted into the first route and (i, h) into the second route; or (i, h) inserted into the first route and (g, j) into the second route. If a combination is viable, this is added to the list of possible CROSS movements. The improvement of a CROSS movement is:

$$I_{ghij} = RS_{gh} + RS_{ij} + RS_{gj} + RS_{ih} + VR_{ghij}$$

where VR_{ghij} is the improvement because of the reduction of the number of necessary trucks. It could happen if a resulting route only contains dummy tasks, and its vehicle has never been used before, NONUSEDVEH.

The CROSS local search stops after a number of iterations without reduction in total expected cost.

COMBINE operator

It is checked if the pairs served by two different vehicles can be combined into a single route. The operator tries to combine the routes of the trucks that have not begun to work, NOUSEDVEH, with other routes. So, it is able to reduce the number of new trucks to be used.

Two routes can be combined if the last pair of the route to combine can be served before the latest starting time of the route which it is combined with.

INSERT operator

This operator removes pairs from their routes and reinserts them into other routes. Like COMBINE operator, INSERT operator tries to reduce the number of new vehicles. The routes to reinsert are the routes of the vehicles which have not been used yet.

TESTING AND RESULTS

To check the impact of knowledge the real time position of vehicle on the drayage management, 30 different problems have been tested. Each test is composed of 30 tasks to realize, 15 delivery tasks and 15 pickup tasks. Time windows, origin and destination are generated randomly. Customer and depot are distributed in a 50x50 area.

To simulate the stochastic transit time, this area is divided into 100 squares, and each square has a speed assigned. The average speed of every square is known but not the real speed. As the transit time is stochastic, 100 different frameworks of speeds are studied in every test problem.

The cost per kilometre is 1, the fixed cost per vehicle is 10, the waiting cost is 10 per hours and the cost of losing a task is 100.

Under these conditions the static insertion heuristic of Caris and Janssens (2009) and, the same heuristic with knowledge of real time position of vehicles and the re-optimization approach have been tested. The results obtained are shown in Table 1.

Table 1 – Results

Test	Average Improvement	Maximum improvement
1	9.9	26.7
2	18.4	28.4
3	1.6	14
4	6.7	14.4
5	5.8	27
6	19.5	25.5
7	-8.3	17.4
8	6.8	23.7
9	6.1	18.4

10	10.6	23.7
11	2.3	18.4
12	6.8	21.8
13	10.4	23.9
14	-0.1	7.4
15	22.4	27.5
16	12.7	21
17	4.6	24.8
18	-0.9	8.3
19	6.4	16.4
20	27.1	35.4
21	7.3	17.7
22	15.1	25.9
23	35.3	43.4
24	21.0	32.6
25	18.0	30.7
26	26.5	32.2
27	4.6	15.4
28	8.3	21
29	1.4	9.8
30	12.5	21.5
TOTAL	10.6	43.4

CONCLUSION

We have shown in this paper the importance of the exact knowledge of real-time locations of vehicle in a drayage fleet. This knowledge, together with a fast optimization algorithm reduces operation costs in a 10%. It is necessary to know that due to the stochastic transit time, although the average improvement is a 10%, the solution could get worse (as is shown in test 7 and 18).

REFERENCES

Bodin, L., Mingozzi, A., Baldacci, R., and Ball, M. (2000). The Rollon–Rolloff Vehicle Routing Problem. Transportation Science, vol. 34, no. 3, pp. 271-288.

Caris, A. and Janssens, G.K. (2009). A local search heuristics for the pre- and end- haulage of intermodal container terminals. Computers & Operations Research, vol. 36, pp. 2763-2772.

Cheung, R. K. and Hang, D. D. (2003). A time-window sliding procedure for driver-task assignment with random service times. IIE Transactions, vol. 35, no. 5, pp. 433-444.

De Meulemeester, L., Laporte, G., Louveaux, F. V., and Semet, F. (1997). Optimal sequencing of skip collections and deliveries. Journal of the Operational Research Society, vol. 48, no. 1, pp. 57-64.

Francis, P., Zhang, G., and Smilowitz, K. (2007). Improved modeling and solution methods for the multi-resource routing problem. European Journal of Operational Research, vol. 180, no. 3, pp. 1045-1059.

Ileri, Y., Bazaraa, M., Gifford, T., Nemhauser, G., Sokol, J., and Wikum, E. (2006). An optimization approach for planning daily drayage operations. Central European Journal of Operations Research, vol. 14, no. 2, pp. 141-156.

Smilowitz, K. (2006). Multi-resource routing with flexible tasks: an application in drayage operations. IIE Transactions, vol. 38, no. 7, pp. 577-590.

Wang, X. and Regan, A. C. (2002). Local truckload pickup and delivery with hard time window constraints. Transportation Research Part B, vol. 36, no. 2, pp. 97-112.