11th IMHRC Proceedings (Milwaukee, Wisconsin. USA – 2010)

Progress in Material Handling Research

2010

Dispatching Vehicles Considering Uncertain Handling Times at Port Container Terminals

Vu Duc Nguyen
Pusan National University

Kap Hwan Kim
Pusan National University

Recommended Citation

Nguyen, Vu Duc and Kim, Kap Hwan, "Dispatching Vehicles Considering Uncertain Handling Times at Port Container Terminals" (2010). *11th IMHRC Proceedings (Milwaukee, Wisconsin. USA – 2010)*. 14.
https://digitalcommons.georgiasouthern.edu/pmhr_2010/14

DISPATCHING VEHICLES CONSIDERING UNCERTAIN HANDLING TIMES AT PORT CONTAINER TERMINALS

Vu Duc Nguyen
Department of Industrial Engineering, Pusan National University,
Busan 609-735, South Korea
duc.ise@pusan.ac.kr

Kap Hwan Kim[†]
The corresponding author
Department of Industrial Engineering, Pusan National University,
Busan 609-735, South Korea
kapkim@pusan.ac.kr

Abstract

This paper considers the problem of vehicle dispatching at port container terminals in a dynamic environment. The problem deals with the assignment of delivery orders of containers to vehicles while taking into consideration the uncertainty in the travel times of the vehicles. Thus, a real-time vehicle dispatching algorithm is proposed for adaptation to the dynamic changes in the states of the container terminals. To evaluate the performance of the proposed algorithm, a simulation study was conducted by considering various values of decision parameters under the uncertainty in travel times. Further, the performance of the proposed algorithm was compared with those of heuristic algorithms from previous studies.

1 Introduction

Because of globalization, many cargoes today are transported from one area of the world to another. Over the last decade, cargo transportation by containerships has rapidly grown in popularity because of its cost efficiency. In container terminals, containers are transferred between containerships and the storage yard via discharging and loading operations. During discharging operations, containers in a containership are unloaded from the containership and stacked in the storage yard, and vice versa during loading operations. In this paper, we consider a port container terminal in which three main types of handling equipment, quay cranes (QCs), vehicles, and yard cranes (YCs), are used for ship operations. Figure 1 illustrates the layout of a seaport container terminal that consists of areas for the QCs, vehicle driving, and YCs. Container terminals have complicated handling systems, and thus, there are many sources of uncertainties during their operation. In particular, the travel times of vehicles may not be considered as being constant any more. This paper attempts to schedule the delivery operations of vehicles while taking into account the uncertainty in the travel times of the vehicles. Figure 2 shows the discharging and loading processes in port container terminals.

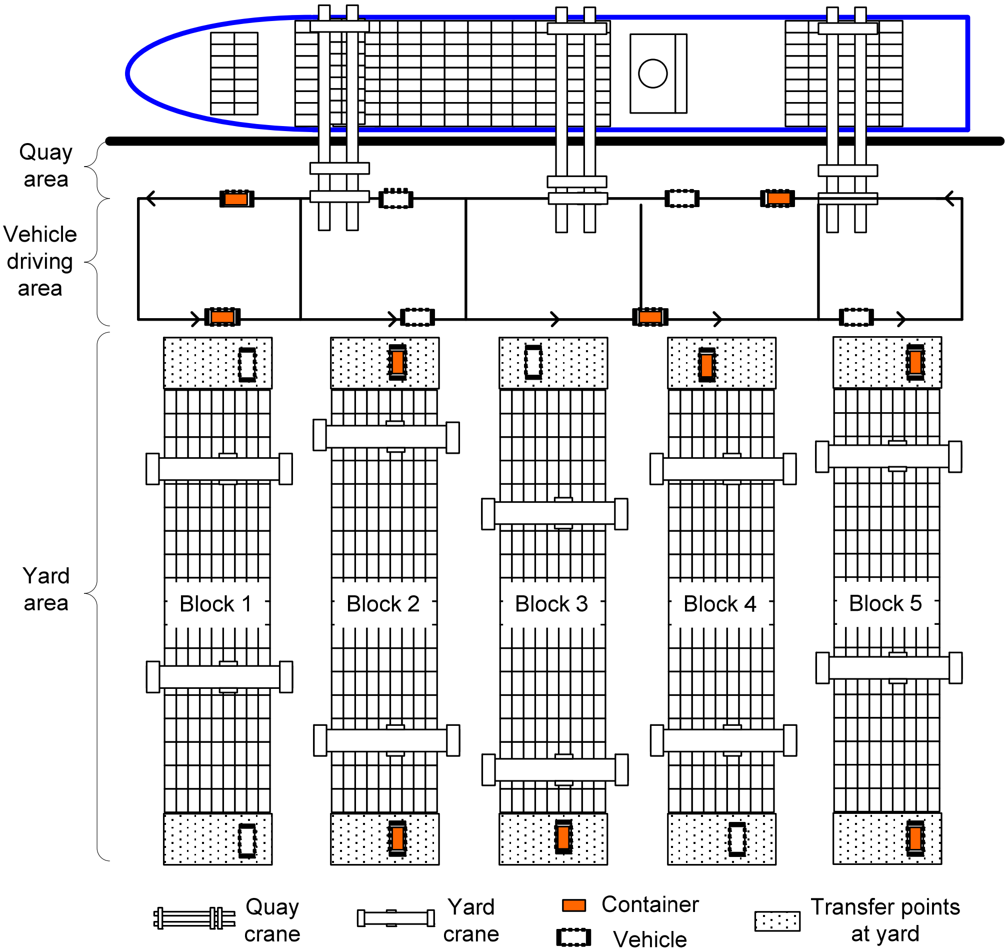

Figure 1: Layout of Seaport Container Terminal.

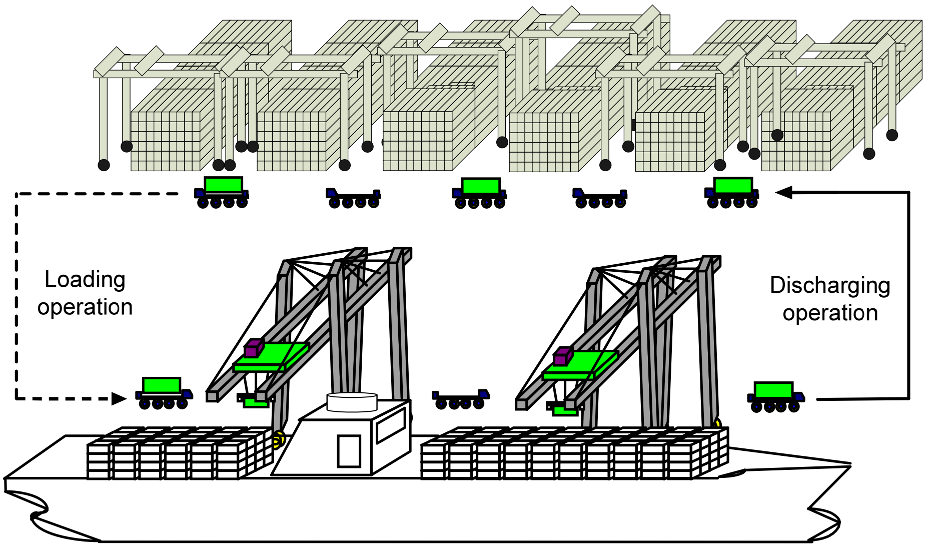

Figure 2: Discharging and Loading Processes.

Vehicle dispatching problems have been addressed by many researchers. Egbelu and Tanchoco [1] presented a dispatching method for single-load automated guided vehicles (AGVs) that incorporated a variety of priority rules. Egbelu [3] suggested a demand-driven rule in which AGVs are first assigned delivery tasks that are allocated to machines with the smallest number of tasks already present in their input buffers. Bilge and Ulusoy [5] presented a method for simultaneously scheduling the operation of machines and the transfer of materials by AGVs. Kim et al. [6] suggested an AGV dispatching method in which the primary criterion for selecting the next delivery task is to balance the workload across different workstations. Van der Meer [9] undertook a simulation study to evaluate various dispatching rules, including rules using pre-arrival information, for automated lifting vehicles (ALVs) in container terminals. Lim et al. [10] introduced an AGV dispatching method using a bidding concept in which the dispatching decisions are made through communication between related vehicles and machines. Kim and Bae [12] presented a mixed-integer programming model for assigning optimal delivery tasks to AGVs and suggested a heuristic algorithm for solving the mathematical model. Using a simulation study, the heuristic algorithm was performed and compared with other dispatching rules. Bish et al. [15] proposed a vehicle dispatching technique to minimize the total time taken to serve a ship. They developed easily implementable heuristic algorithms and identified both the absolute and the asymptotic worst-case performance ratios of these heuristics. Briskorn et al. [16] presented an alternative formulation of the AGV assignment problem that does not include due times and that is based on a rough analogy to inventory management; they proposed an exact algorithm for solving the formulation. Grunow et al. [17] described a simulation study of AGV dispatching strategies in a seaport container terminal, where AGVs can be used in either single- or dual-carrier mode. They compared a typical, on-line dispatching strategy adopted from flexible manufacturing systems with a pattern-based, off-line heuristic algorithm. Nguyen and Kim [19] developed a mathematical formulation of the dispatching problem for ALVs. They suggested a heuristic algorithm and compared its solutions to optimal solutions. Angeloudis and Bell [20] presented a flexible AGV dispatching algorithm capable of operating under uncertain conditions within a detailed container terminal model. Several performance indicators were described, focusing on generic features of vehicle operations as well as the assessment of uncertainty levels inside the terminal. From the results of the simulations, it was found that their technique outperforms well-known heuristics and alternative algorithms.

However, in container terminals there are many uncertain factors. Simulations have been used as a powerful tool for analyzing the performance of port container terminals in complex dynamic environments. Various levels of detail and the uncertainties in container terminals can be expressed in simulation models. Much research on the development of simulation models of container terminals has been published (Cho [2]; Yun and Choi [7]; Tahar and Hussain [8]). Hartmann [11] introduced an approach for generating scenarios that can be used as input data for simulation models of port container terminals. Through a simulation study, Vis and Harika [13] and Yang et al. [14] compared the performance of two types of automated vehicle, namely AGVs and ALVs. Lee et al. [18] undertook a simulation study comparing various handling systems consisting of different types of transport vehicle (prime movers and shuttle carriers) and different storage-yard layouts (with and without a chassis lane beside blocks).

This paper is organized as follows. Section 2 introduces the ship operation and the method of operational control for vehicles in container terminals. A heuristic algorithm for solving the vehicle dispatching problem bearing in mind the uncertainties is proposed in Section 3. Section 4 presents the results of a simulation experiment for comparing the proposed heuristic algorithm with other algorithms and analyzing the performance of the proposed heuristic algorithm. Finally, some conclusions and issues for further research are set out in Section 5.

2 Ship Operation and Operation Control Method for Vehicles

Before a ship arrives at a port container terminal, all information regarding the inbound and outbound containers is sent to the terminal by a shipping agent. Based on this information, a list of the sequence of discharging and loading operations for individual containers is then made. When the ship actually arrives, ship operations are usually performed on the basis of the discharging and loading sequence list.

For a discharging operation, after receiving a container from a QC, the vehicle delivers it to the designated storage yard. When the vehicle arrives at the yard, it waits at the transfer point (TP) in the yard for the container to be picked up by YC. A YC picks up the container and stacks it in an empty slot in a bay. Loading operations are performed in exactly reverse order. During the discharging operation, a vehicle and a QC must converge when the QC releases an inbound container onto the vehicle, and the vehicle and a YC must converge when the YC picks up the inbound container from the vehicle. Similar convergences occur during the loading operation. This necessity for synchronization frequently causes delays to transport operations in container terminals.

In container terminals, a vehicle can be considered a resource that has to be efficiently dispatched. To adapt to a changing environment, a dispatching decision must be made whenever an important event occurs. Ship operation planners develop the sequence of discharging and loading operations for each QC. The sequence is initially put into LIST *A* in Figure 3. Among the tasks (discharging and loading operations) in LIST *A*, a pre-specified number of the most immediate tasks for each QC are moved to LIST *B*. The tasks in LIST *B* are candidates for assignation to vehicles. Whenever a vehicle commences travel to pick up a container for a task, the task is removed from LIST *B*, and the next urgent task from the corresponding LIST *A* is moved to LIST *B*. Note that for each QC, the same number of immediate tasks must be maintained in LIST *B*, unless LIST *A* becomes mainly empty.

The dispatching algorithm is triggered when a vehicle becomes idle. When a vehicle completes a delivery task, the vehicle reports the completion of the task to the control system (CS). The CS will then trigger the dispatching algorithm for assigning delivery tasks to vehicles. Following the outcome of dispatching, if the vehicle is assigned a delivery task, it will commence traveling to the pickup position of the assigned task. Otherwise, the vehicle will move to the parking area to await the next assignment. When a QC completes a delivery task, completion of the task is reported to the CS, and the next task is added to the QC's task list. The CS will then trigger the dispatching algorithm for reassigning delivery tasks to vehicles. When an idle vehicle is assigned a task, the vehicle will commence travel.

When the vehicle arrives at the designated QC, its arrival will be reported to the CS. At the quay, the vehicle checks the status of the QC. For loading tasks, if the QC is not available, the vehicle has to wait until it becomes so. If it is available, the QC picks up the outbound container from the vehicle. Similarly, for discharging tasks, the vehicle has to wait until the QC becomes available, and the QC releases the inbound container onto the vehicle. The change in status of the vehicle is then reported to the CS. When the vehicle departs from the QC with an unloaded container, it will go to a designated block to deliver it. When the vehicle completes the delivery of a loading container, the dispatching procedure is triggered for assigning another task to the vehicle. When no task is assigned, the vehicle becomes idle and moves to the parking area. When a task is assigned, it moves to the pickup position of the next assigned task. All changes in the system status are reported to the CS.

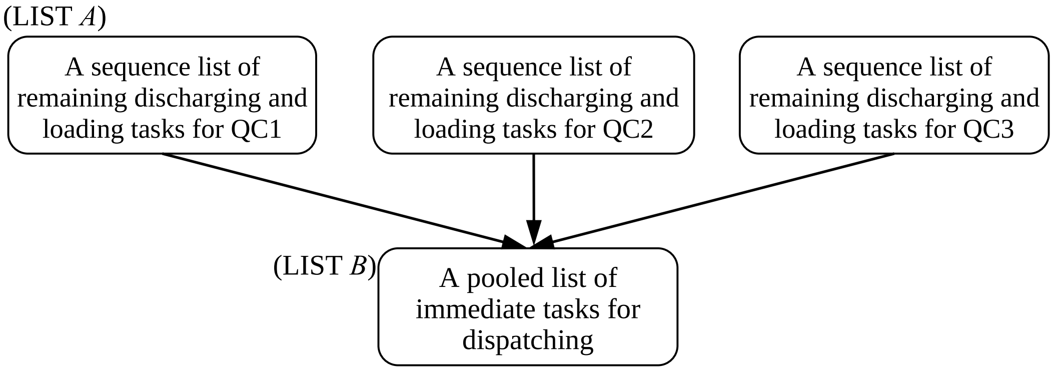
(LIST A)
A sequence list of remaining discharging and loading tasks for QC1
A sequence list of remaining discharging and loading tasks for QC2
A sequence list of remaining discharging and loading tasks for QC3
(LIST B)
A pooled list of immediate tasks for dispatching

Figure 3: Various Lists of Tasks for Dispatching.

3 Heuristic Algorithm Considering Uncertainties

The vehicle dispatching problem was formulated as a mixed-integer programming (MIP) model, and a detailed description of this formulation can be found in Kim and Bae [12]. Their suggested algorithm assumed deterministic handling and travel times for the equipment. The present paper extends their dispatching heuristic algorithm by relaxing this assumption.

The following first introduces a formulation of the dispatching problem and the heuristic algorithm by Kim and Bae [12]. A loading operation cycle by a QC begins with the pickup of a container from a vehicle, while a discharging operation cycle ends with the release of a container onto a vehicle. For a QC operation to be performed without delay, a vehicle must be ready at a specified location beneath the associated QC before the transfer of a container commences.

Let e_i^k be an event representing the moment at which a vehicle transfers the ith container of QC k (the ith operation of QC k). When the ith operation of QC k is a loading operation, event e_i^k corresponds to the beginning of the pickup of a container from a vehicle. When the ith operation of QC k is a discharging operation, it corresponds to the beginning of the release of a container onto a vehicle. The time of event e_i^k is denoted Y_i^k. A delay to an operation occurs when the corresponding vehicle does not arrive at the requested moment, which is the time of the event with no delays to QC operation and is represented by s_i^k, i.e., the earliest possible event time.

Three types of events are undergone by vehicles during a ship operation: The initial event, which represents the current state of each vehicle; the event when a vehicle begins to receive a container from a QC or when a vehicle begins to transfer a container to a QC; and the stopping event, when a vehicle completes all of its assigned tasks.

The notations related to ship operations are summarized as follows:

V = The set of vehicles.

K = The set of QCs.

e_v^O = The initial event of vehicle *v*, $v \in V$.

e_v^F = The stopping event of a vehicle *v*, $v \in V$. Note that, although the number of stopping events of vehicles is the same as the number of vehicles, stopping events with different subscripts do not need to be distinguished from each other.

e_i^k = The event that corresponds to the beginning of a pickup (or release) of a container from (onto) a vehicle for the task related to the *i*th operation of QC *k*. Assume that there exist m_k tasks for QC *k*.

T = The set of e_i^k for $i = 1, 2, \cdots, m_k$ and $k \in K$.

$l(e_i^k)$ = The location at which event e_i^k occurs. $l(e_v^O)$ represents the initial location of vehicle *v*. Here, $l(e_i^k)$ represents the position at which the *i*th container of QC *k* will be transferred. The location at which a vehicle completes its final delivery task is denoted $l(e_v^F)$.

t_{ki}^{lj} = The pure travel time from $l(e_i^k)$ to $l(e_j^l)$.

C_{ki}^{lj} = The time required for a vehicle to be ready for e_j^l after undergoing e_i^k, which is a random variable. For example, if both e_i^k and e_j^l are related to loading operations, then the starting moment (event) for evaluating c_{ki}^{lj} is the pickup of the *i**th* container of QC *k* by QC *k*. Included in C_{ki}^{lj} are the travel time from the apron to the location of the next container (the *j*th container of QC *l*) in the marshalling yard, the release time of the container by a YC, and the travel time of the vehicle to QC *l*.

Let S and D be the sets of e_v^O and e_v^F, $v \in V$, respectively. A feasible dispatching decision is then a one-to-one assignment between all the events in $S \cup T$ and those in $D \cup T$. Let $K' = \{O\} \cup K$, $K'' = \{F\} \cup K$, and x_{ki}^{lj} be a decision variable that becomes 1 if e_i^k is assigned to e_j^l, for $k \in K'$ and $l \in K''$. For $k, l \in K$, the assignment of e_i^k to e_j^l implies that the vehicle that has just delivered the *i*th container of QC *k* is scheduled to deliver the *j*th container of QC *l*.

Let α be the travel cost per unit time of a vehicle, and β be the penalty cost per unit time for a delay in the completion time. It is assumed that $\alpha << \beta$. Further, let m_O and m_F equal $|V|$. The dispatching problem can then be formulated as follows:

Minimize
$$\alpha \sum_{k \in K'} \sum_{i=1}^{m_k} \sum_{l \in K''} \sum_{j=1}^{m_l} t_{ki}^{lj} \cdot x_{ki}^{lj} + \beta \sum_{k \in K} E[(Y_{m_k}^k - s_{m_k}^k)^+] \qquad (1)$$

Subject to

$\sum_{l \in K''} \sum_{j=1}^{m_l} x_{ki}^{lj} = 1$, for $\forall k \in K'$ and $i = 1, \cdots, m_k$ (2)

$\sum_{k \in K'} \sum_{i=1}^{m_k} x_{ki}^{lj} = 1$, for $\forall l \in K''$ and $j = 1, \cdots, m_l$ (3)

$Y_j^l - (Y_i^k + C_{ki}^{lj}) \ge M(x_{ki}^{lj} - 1)$, for $\forall k \in K', l \in K$, $i = 1, \cdots, m_k$, and $j = 1, \cdots, m_l$ (4)

$Y_v^O = 0$, for $\forall v = 1, \cdots, |V|$ (5)

$Y_{i+1}^k - Y_i^k \ge s_{i+1}^k - s_i^k$, for $\forall k \in K$ and $i = 1, \cdots, m_k - 1$ (6)

$y_i^k \ge s_i^k$, for $\forall k \in K'$ and $i = 1, \cdots, m_k$ (7)

$x_{ki}^{lj} = 0$ or 1, for $\forall k \in K'$, $l \in K''$, $i = 1, \cdots, m_k$, and $j = 1, \cdots, m_l$ (8)

Because $\alpha << \beta$, the sum of the delays to QC operations will be minimized first. For the same value of the sum of the delays, the total travel distance of the vehicles will be minimized. Constraints (2) and (3) force the one-to-one assignment between all the events in $S \cup T$ and those in $D \cup T$. Constraint (4) implies that two events that are served consecutively by the same vehicle must be set apart by at least the time required for the vehicle to travel and transfer a load between the two events. That is, x_{ki}^{lj} can be 1 only if $Y_j^l - Y_i^k \ge C_{ki}^{lj}$. Note that x_{ki}^{Fj}, $k \in K$, is not restricted by constraint (4). Constraint (6) implies that two events that are served by the same QC must be set apart by at least the time required for the QC to perform all the movements between the two events. Constraint (7) signifies that the actual event time is always more than or equal to the earliest possible event time. A feasible solution of (x_{ki}^{lj}) is a one-to-one assignment from a node in $S \cup T$ to a node in $D \cup T$.

Let us express the above formulation in a more general way as follows:

Minimize $f(X, Y)$ subject to $g(X, Y, C) = 0$ (9)

Kim and Bae [12] solved the problem by setting $C_{ki}^{lj} = c_{ki}^{lj}$ and increasing the values of Y by the smallest possible increments so that delay cost could be minimized. Once the values of Y ($Y_i^k = y_i^k$) and C ($C_{ki}^{lj} = c_{ki}^{lj}$) are given, (9) becomes an assignment problem, with some assignments forbidden because of constraint (4). That is, for a given set of y_i^k and y_j^l, if the inequality $y_j^l - (y_i^k + c_{ki}^{lj}) \ge 0$ holds, then x_{ki}^{lj} cannot equal 1. The remaining problem is how to increase the values of Y_i^k. Kim and Bae [12] fixed them as follows: Suppose that Y_i^k (which are equal to s_i^k in the initial stage), $k = 1, ..., |K|$, and $i = 1, \cdots, m_k$ are given and the events are sequenced in increasing order of Y_i^k. We denote "event (*j*)" as the *j*th event in the sequence, y_i as the time of event (*i*), c_{ij} as the time required for a vehicle to be ready for event (*j*) after it goes through event (*i*) (corresponding to the notation of c_{ki}^{lj}), t_{ij} as the pure travel time from the location of event (*i*) to the location of event (*j*), and x_{ij} as the decision variable for the assignment of event (*j*) from event (*i*). Let T_ξ be a subset of *T*, which includes only the first ξ events in the sequence. The constraint subset ξ of (2)–(4) can then be written as follows:

(*Constraint subset* ξ)

$$\sum_{i \in S \cup T_\xi} x_{ij} = 1, \qquad \text{for } \forall j \in D \cup T_\xi \tag{10}$$

$$\sum_{j \in D \cup T_\xi} x_{ij} = 1, \qquad \text{for } \forall i \in S \cup T_\xi \tag{11}$$

$$y_j - (y_i + c_{ij}) \geq M(x_{ij} - 1), \qquad \text{for } \forall i \in S \cup T_\xi \text{ and } j \in T_\xi \tag{12}$$

$$x_{ij} = 0 \text{ or } 1, \qquad \text{for } \forall i \in S \cup T_\xi \text{ and } j \in D \cup T_\xi \tag{13}$$

In the algorithm, for given values of y_i^k, the feasibility of each is checked one at a time from the constraint subset 1 to the constraint subset $|T|$. In the process, if an infeasible constraint subset is found, the infeasibility is resolved by increasing an event time so that one or more x_{ij} can be allowed to be 1 by constraint (12). During iterative procedures of the algorithm, attempts to minimize the delay to QC operation are made by increasing y_i^k by the least possible amount. However, after a feasible solution to constraint subset $|T|$, which is equivalent to constraints (2) and (3), is found, the total travel time of vehicles, which is the first term of objective function (1), will be minimized by applying the assignment problem technique.

A similar procedure will be followed in the algorithm described in this paper. In Kim and Bae [12], because c_{ij} is constant, for a given set of values of *Y*, it is clear whether the constraint $y_j - (y_i + c_{ij}) \geq 0$ is satisfied or not. However, in constraint (4), because C_{ij} is a random variable, it is not certain whether or not $y_j - (y_i + C_{ij}) \geq 0$ holds. Thus, we define a probability function $P_{ij} = P\{ y_j - (y_i + C_{ij}) \geq 0 \}$, which can be easily evaluated once $P(C_{ij})$ is given, and modify the constraint subset ξ as follows:

(*Revised constraint subset* ξ)

$$x_{ij} - 1 \leq P_{ij} - \theta, \qquad \text{for } \forall i \in S \cup T_\xi \text{ and } j \in D \cup T_\xi \tag{14}$$

$$y_j - \left\{ y_i + \left(E[C_{ij}] / P_{ij} \right) \right\} \geq M\left(x_{ij} - 1 \right), \qquad \text{for } \forall i \in S \cup T_\xi \text{ and } j \in T_\xi \tag{15}$$

and (10), (11), and (13).

Constraint (14) implies that e_i can be connected to e_j only if $P_{ij} \geq \theta$. A higher penalty is given to the assignment with a lower probability of timely delivery. That is, c_{ij} in (12) is replaced with $E[C_{ij}]/P_{ij}$.

A detailed heuristic algorithm can then be described as follows:

Step 0. *Initializing.* Set $y_i^k = s_i^k$ and $y_i^O = 0$, for all vehicles. Set $\xi = 0$.

Step 1. *Next Task.* $\xi = \xi + 1$. If $\xi > m$ (m = the total number of tasks in sequence T), then go to Step 4. Otherwise, sequence the events in increasing order of y_i^k and go to Step 2.

Step 2. *Feasibility Checking.* Check the existence of a feasible solution to revised constraint subset ξ. If there is a feasible solution, then go to Step 1. Otherwise, go to Step 3.

Step 3. *Delaying Event Time.*

Let $\pi_{i^*\xi} = \min_{i \in S \cup T_{\xi-1}} \left[\max\left\{ (E[C_{i\xi}]/P_{i\xi}) - (y_\xi - y_i), 0 \right\} \right]$, where $\left\{ (i, \xi) : P_{i\xi} \geq \theta \right\}$.

Denote y_λ^γ as the event time of event (ξ). Then update $y_j^\gamma = y_j^\gamma + \pi_{i^*\xi}$, for $j \geq \lambda$.

Go to Step 2.

Step 4. *Task Assignment.* After setting the assignment cost of event i to j so as to be equal to (t_{ij}/P_{ij}), solve the assignment problem with the objective of minimizing the total assignment cost. Stop.

Feasibility Check: In this step, for given values of y_i^k, the feasibility can be checked by solving a maximum cardinality matching problem in a bipartite graph (Evans and Minieka [4]). When the maximum cardinality is the same as $|S \cup T|$, revised constraint subset ξ has a feasible solution. When solving the matching problem in the bipartite graph, arcs from node i to j are linked in the graph only if $P_{ij} \geq \theta$.

Delaying Event Time: To satisfy the revised constraint subset, one or more additional x_{ij} must be allowed to become 1 by relaxing constraint (15). In other words, the time for event ξ is delayed so that at least one $x_{i\xi}$, for $i < \xi$, becomes 1, denoted $\pi_{i^*\xi}$. The process is repeated until the current constraint subset becomes feasible.

4 Simulation Experiments

A simulation model was developed using Plant-Simulation software. Detailed operation of the hypothetical container terminal (Figure 1) can be described as follows. When a ship arrives, it is assigned a berth if there is one available for the ship to enter. Otherwise, the ship must wait until one becomes available. When the ship enters a berth, a pre-specified number of QCs are assigned to the ship. A discharging and loading sequence for containers is then generated for each QC. Based on the specified sequence, QCs start to discharge and load containers.

The wharf of the model terminal in Figure 1 has one berth and three QCs. The yard consists of six storage blocks, and two YCs of the same size are deployed at each block. The total number of vehicles is six. The vehicles are shared between all the QCs, that is, a pooling strategy is used for dispatching vehicles. From LIST *A* for each QC, the eight most immediate tasks are moved to LIST *B* for dispatching. That is, the number of looking-ahead tasks is 24. The total number of containers transferred by QCs during one simulation run is about 1000. The detailed movements of QCs and YCs (gantry, trolley, and hoisting movements) are modeled in the simulation.

The travel time of vehicles is assumed to follow a uniform distribution: $U(\mathrm{E}[C_{ij}] \pm \Delta\cdot\mathrm{E}[C_{ij}])$. $\mathrm{E}[C_{ij}]$ is calculated using the travel distance from the position of event i to that of event j divided by the speed of vehicles, and Δ is a constant referred to here as the "uncertainty factor". The uncertainty factor is set to be 0.2 in the experiments. The threshold of the connecting probability, θ, has a value of 0.5 to 1.0. The performance measures compared in the simulation experiments are the total delay time of QCs, the total travel time of vehicles, the total travel time of empty vehicles, and the vehicle throughput, which is the number of delivery tasks performed per hour.

The performance of the proposed heuristic algorithm supporting the uncertainties of travel times (LADP-*un*) was compared with that of the Greedy algorithm and that of the heuristic algorithm suggested by Kim and Bae [12] for the deterministic case (LADP-*de*). For the Greedy algorithm, whenever a vehicle becomes idle, it is assigned the delivery task that incurs the minimum assignment cost (t_{ij}/P_{ij}) of all the tasks in *LIST B* that can be performed by vehicles without violating constraints.

Table 1 lists the total delay time of QCs, the total travel time of vehicles, the total travel time of empty vehicles, the vehicle throughput, and the computational time for each algorithm. As can be seen in Table 1, LADP-*un* showed the best performance, and both LADP-*un* and LADP-*de* significantly outperformed the Greedy algorithm in terms of the total delay time of QCs and the vehicle throughput. However, the Greedy algorithm was the best in terms of the total travel time of vehicles and the total travel time of empty vehicles, because both the LADP algorithms first attempted to minimize the total delay time of QCs before then attempting to minimize the total travel time of vehicles as a secondary objective. In addition, the Greedy algorithm spent the least computational time solving instance problems, and LADP-*un* took relatively longer than LADP-*de* in terms of computational time per instance. Figure 4 shows the improvement in the performance of LADP-*un* over that of other algorithms (Greedy and LADP-*de*). Its performance was a 55.11% improvement on that of the Greedy algorithm and 18.45% on that of LADP-*de* in terms of the total delay time of QCs. The vehicle throughput of LADP-*un* was 25.83% larger than that of the Greedy algorithm and 4.15% greater than that of LADP-*de*.

Table 1: Comparison of LADP-*un*, Greedy, and LADP-*de* Algorithms.

Algorithm	Total delay time of QCs (s)	Total travel time of vehicles (s)	Total travel time of empty vehicles (s)	Vehicle throughput (moves/hour)	Computational time per instance (ms)
Greedy	306574	**87557**	**41378**	728	**17**
LADP-*de*	168748	89443	43078	879	1186
LADP-*un*	**137616**	89319	42452	**916**	1525

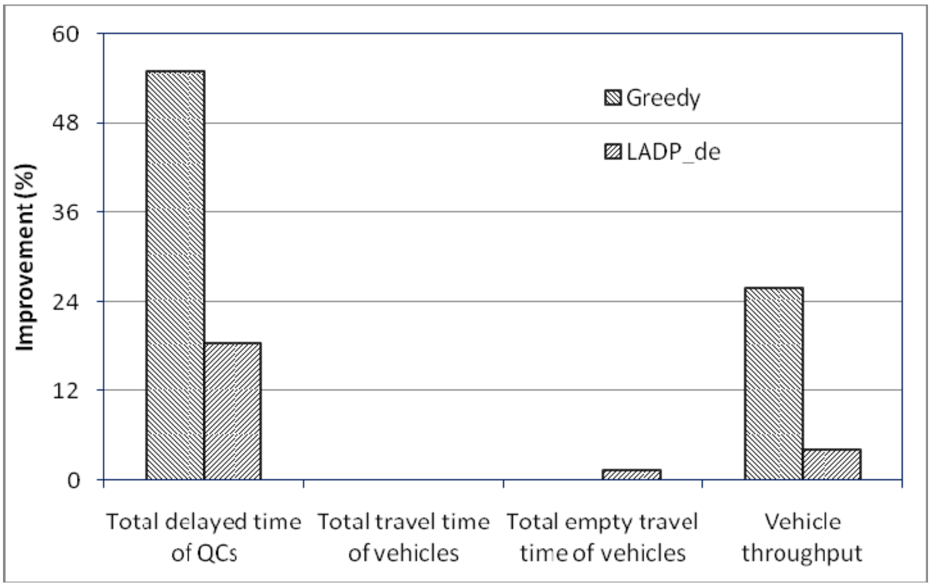

Figure 4: Improvement in Performance of LADP-*un* Over That of Greedy and LADP-*de* Algorithms.

Figures 5–8 show the changes in the total delay time of QCs and computational time, the total travel time of vehicles, the total empty travel time of vehicles, and the vehicle throughput, respectively, for various thresholds of the connecting probability. Figure 5 shows that the total delay time of QCs decreases rapidly as the threshold of the connecting probability decreases. As the threshold of the connecting probability decreases, the number of arcs connecting nodes in the bipartite graph increases. As a result, the average computational time per instance increases. However, because the feasible solution of the problem becomes larger, the solution quality improves, as can be observed in Figure 5. When the threshold of the connecting probability falls below a certain value (0.6, as indicated in Figure 5), the change in the reduction in the total delay time becomes smaller.

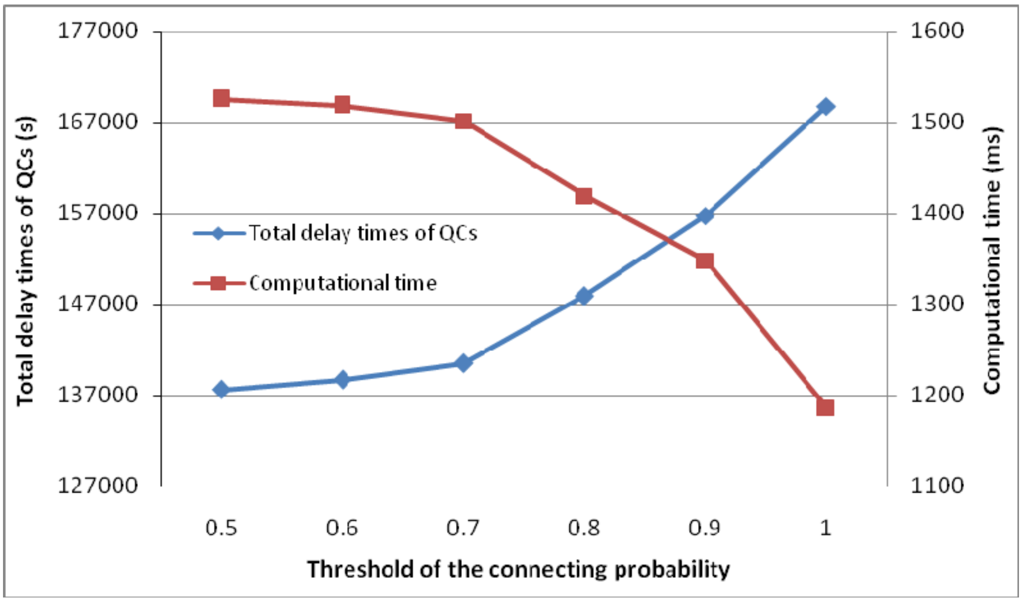

Figure 5: Effects of Threshold of Connecting Probability on Total Delay Time of QCs and Average Computational Time per Instance.

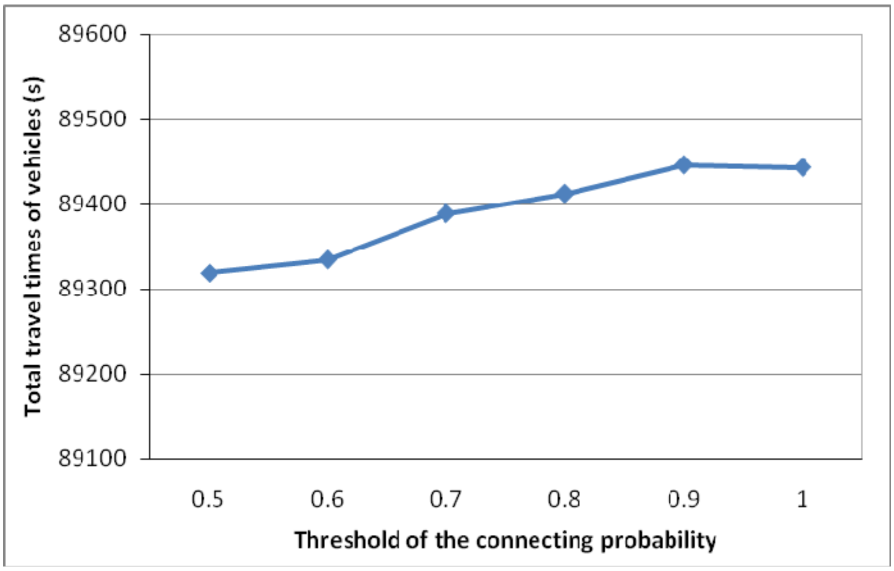

Figure 6: Effects of Threshold of Connecting Probability on Total Travel Times of Vehicles.

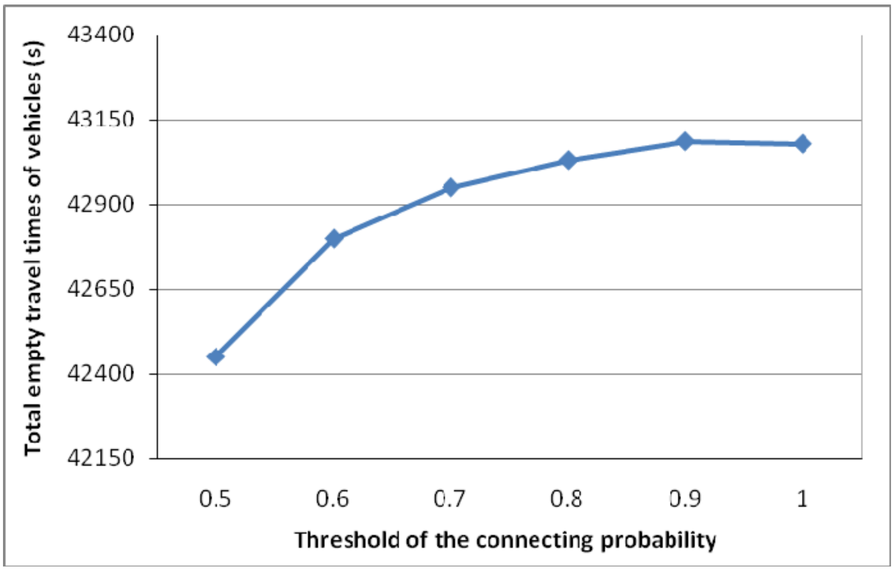

Figure 7: Effects of Threshold of Connecting Probability on Total Empty Travel Times of Vehicles.

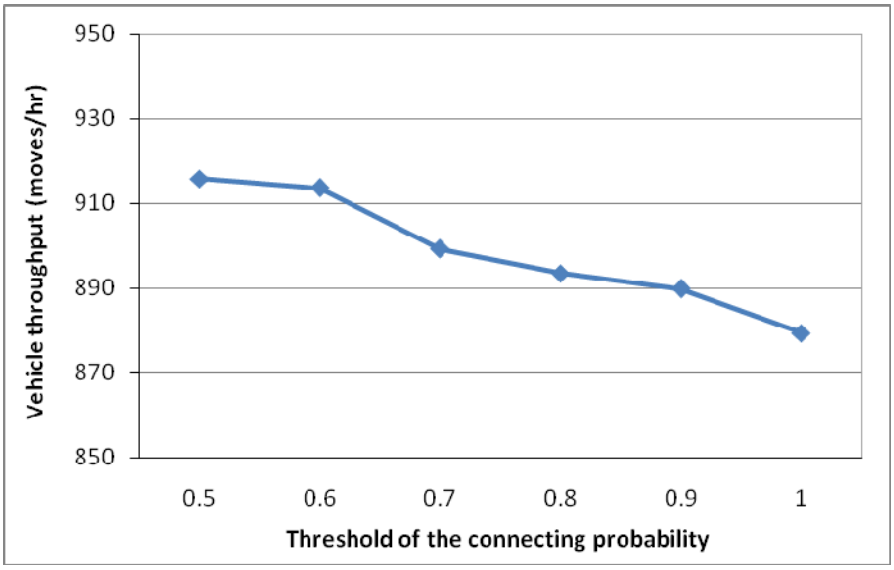

Figure 8: Effects of Threshold of Connecting Probability on Vehicle Throughput.

Similarly, the changes in the total travel time of vehicles and total empty travel time of vehicles are shown in Figures 6 and 7. The total travel time of vehicles and the total empty travel of vehicles increases quickly as the threshold of the connecting probability increases and reaches 0.9. However, the vehicle throughput decreases as the threshold of the connecting probability increases, as shown in Figure 8. Note that the results in Figures 5–8 compare LADP-*de* and LADP-*un* because the case in which the threshold equals 1 corresponds to LADP-*de*. The results show that LADP-*un* outperforms LAPD-*de* in its objective values at the expense of greater computational time. For example, Figure 5 shows that the percentage difference between the two algorithms in the total delay time of QCs increased from 7.14% to 18.45% as the threshold of the connecting probability decreased from 0.9 to 0.5.

5 Conclusions

This paper has discussed the vehicle dispatching problem in port container terminals while taking into account the uncertainty in the travel times of vehicles. A heuristic algorithm (LADP-*un*) was proposed for solving the problem. Simulation models were developed to evaluate the performance of the proposed heuristic algorithm under various conditions. The performance of LADP-*un* was compared with a greedy heuristic rule (Greedy) and a heuristic algorithm for the case with deterministic travel times (LADP-*de*). From the experimental results, it was found that LADP-*un* outperformed the other algorithms in terms of the total delay time of QCs and the vehicle throughput.

It was also found that the total delay time of QCs, the total travel time of vehicles, and the empty travel time of vehicles decreased rapidly when the threshold of the connecting probability decreased. Moreover, the vehicle throughput increased as the threshold was reduced.

This study mainly introduced the scheduling problem for vehicles. As part of future studies, the combined scheduling problem for YCs and QCs as well as for vehicles may be addressed.

Acknowledgements

This study was supported by the Korean Ministry of Education & Human Resources Development through the Research Center of Logistics Information Technologies (LIT).

References

[1] Egbelu, P. J. and Tanchoco, J. M. A., “Characterization of Automatic Guided Vehicle Dispatching Rules,” *International Journal of Production Research*, 22, 359–374 (1984).

[2] Cho, D. W., “A Computer Simulation Model for Container Terminal Systems,” *Journal of the Korean Institute of Industrial Engineers*, 11, 173–187 (1985).

[3] Egbelu, P. J., “Pull Versus Push Strategy for Automated Guided Vehicle Load Movement in a Batch Manufacturing System,” *Journal of Manufacturing Systems*, 6, 209–221 (1987).

[4] Evans, J. R. and Minieka, E., *Optimization Algorithms for Networks and Graphs*, Marcel Dekker, New York (1992).

[5] Bilge, U. and Ulusoy, G., "A Time Window Approach to Simultaneous Scheduling of Machines and Material Handling System in an FMS," *Operations Research*, 43, 1058–1070 (1995).

[6] Kim, C. W., Tanchoco, J. M. A., and Koo, P. H., "AGV Dispatching Based on Workload Balancing," *International Journal of Production Research*, 37, 4053–4066 (1999).

[7] Yun, W. Y. and Choi, Y. S., "A Simulation Model for Container-Terminal Operation Analysis Using an Object-oriented Approach," *International Journal of Production Economics*, 59, 221–230 (1999).

[8] Tahar, R. M. and Hussain, K., "Simulation and Analysis for the Kelang Container Terminal Operations," *Logistics Information Management*, 13, 14–20 (2000).

[9] Van der Meer, R., *Operational Control of Internal Transport*, Ph.D. dissertation, Erasmus Research Institute of Management, Rotterdam, The Netherlands (2000).

[10] Lim, J. K., Kim, K. H., Yoshimoto, K., Lee, J. H., and Takahashi T., "A Dispatching Method for Automated Guided Vehicles by Using a Bidding Concept," *OR Spectrum*, 25, 25–44 (2003).

[11] Hartmann, S., "Generating Scenarios for Simulation and Optimization of Container Logistics," *OR Spectrum*, 26, 171–192 (2004).

[12] Kim, K. H. and Bae, J. W., "A Look-ahead Dispatching Method for Automated Guided Vehicles in Automated Port Container Terminals," *Transportation Science*, 38, 224–234 (2004).

[13] Vis, I. F. A. and Harika, I., "Comparison of Vehicle Types at an Automated Container Terminal," *OR Spectrum*, 26, 117–143 (2004).

[14] Yang, C. H., Choi, Y. S., and Ha, T. Y., "Simulation-based Performance Evaluation of Transport Vehicles at Automated Container Terminals," *OR Spectrum*, 26, 149–170 (2004).

[15] Bish, E. K., Chen F. Y., Leong, Y. T., Nelson, B. L., Ng, C. J. W., and David, S. L., "Dispatching Vehicles in a Mega Container Terminal," *OR Spectrum*, 27, 491–506 (2005).

[16] Briskorn, D., Drexl, A., and Hartmann, S., "Inventory-based Dispatching of Automated Guided Vehicles on Container Terminals," *OR Spectrum*, 28, 611–630 (2006).

[17] Grunow, M., Günther, H. O., and Lehmann, M., "Strategies for Dispatching AGVs at Automated Seaport Container Terminals," *OR Spectrum*, 28, 587–610 (2006).

[18] Lee, L. H., Chew, E. P., Tan, K. C., Huang, H. C., Lin, W., Han, Y., and Chan, T. H., "A Simulation Study in the Uses of Shuttle Carriers in the Container Hub," *Proceedings of the Winter Simulation Conference*, Washington D.C., 1994–2002 (2007).

[19] Nguyen, V. D., and Kim, K. H., "A Dispatching Method for Automated Lifting Vehicles in Automated Port Container Terminals," *Computers & Industrial Engineering*, 56, 1002–1020 (2009).

[20] Angeloudis, P. and Bell, M. G. H., “An Uncertainty-aware AGV Assignment Algorithm for Automated Container Terminals,” *Transportation Research Part E*, 46, 3, 354–366 (2010).