International Journal of Management Science
Vol 6, No 2, November 2000

A POOLED DISPATCHING STRATEGY FOR AUTOMATED GUIDED VEHICLES IN PORT CONTAINER TERMINALS

Jong Wook Bae

Port Operation & Information System Research Division
Korea Maritime Institute, Korea

Kap Hwan Kim

Department of Industrial Engineering
Pusan National University, Korea

(Received October 1999 ; revision received July 2000)

ABSTRACT

It is discussed how to assign delivery tasks to automated guided vehicles (AGVs) for multiple container cranes in automated container terminals. The primary goal of dispatching AGVs is to complete all the loading and discharging operations as early as possible, and the secondary goal is to minimize the total travel distance of AGVs. It is assumed that AGVs are not dedicated to a specific container crane but shared among multiple cranes. A mathematical formulation is developed and a heuristic algorithm is suggested to obtain a near optimal solution within a reasonable amount of computational time. The single-cycle and the dual-cycle operations in both the seaside and the landside operations are analyzed. The effects of pooling AGVs for multiple container cranes on the performance of an entire AGV system are also analyzed through a numerical experiment.

1. INTRODUCTION

In port container terminals, in order to achieve a shorter turnaround time of a containership, it is important to reduce the time needed for ship operations. Ship operations consist of the discharging operation, during which containers in a container-ship are unloaded from a containership and stacked in a marshalling yard, and the loading operation, during which containers are handled in the reverse direction of the discharging operation.

In this study, it is assumed that three different types of equipment are used for ship operations: container cranes (CCs), prime movers, and yard cranes. A CC transfers an inbound container from a containership to a prime mover. Then, the prime mover delivers the discharged container to a yard crane, which picks it up and stacks it into a position in a marshalling yard. For the loading operation, the process is carried out in the opposite direction. The handling activities performed by CCs are called "seaside operations," while those performed by prime movers and yard cranes are called "landside operations."

It is assumed that the port container terminal is automated. The yard crane may be an automated rail-mounted gantry crane, an automated stacking crane (ASC), or an automated overhead bridge crane, and the prime mover is an automated guided vehicle (AGV). However, the mathematical model and the algorithm in this paper can also be utilized even in conventional container terminals. Figure 1 illustrates an automated port container terminal and the equipment used in the loading and discharging operations.

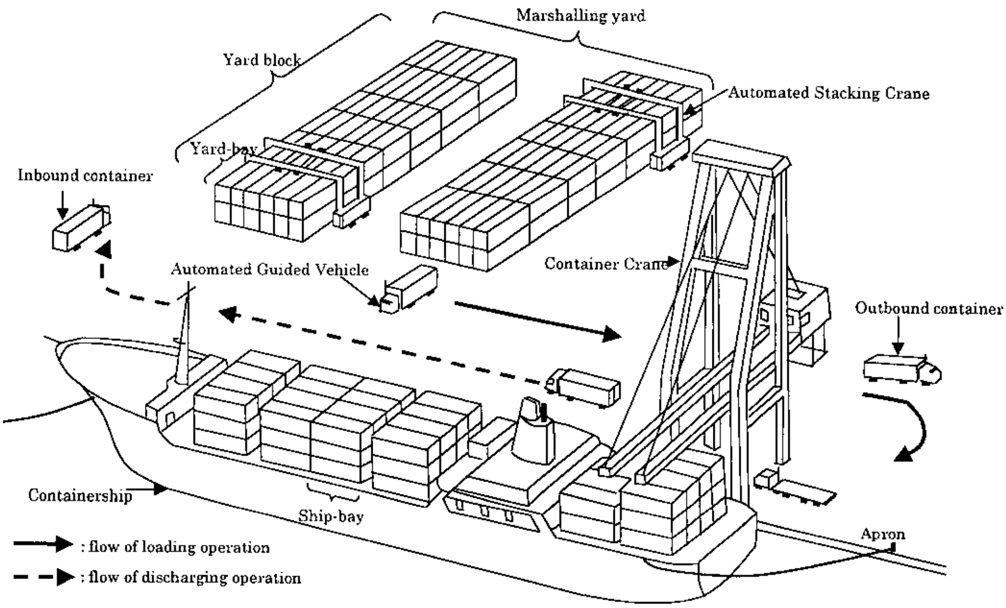


Figure 1. A bird's−eye view of the ship operation in an automated port container terminal

Either of two operating methods can be adopted for a seaside operation. The methods are a single-cycle operation and a dual-cycle operation. When a CC performs a seaside operation in a single-cycle, either loading operations or discharging operations are carried out consecutively. However, during a dual-cycle seaside operation, it happens that a loading operation immediately follows a discharging operation, or viceversa. By performing the loading and the discharging operations alternately, the empty traversal time of a CC trolley can be reduced.

The landside operation can also be performed in either a single-cycle or a dual-cycle. During the single-cycle landside operation, an AGV delivers a container from an apron (yard) to a yard (apron) and returns empty for the next inbound (outbound) container. When the dual-cycle landside operation is performed, an empty AGV, which delivered an outbound container to a CC, can receive another inbound container from a CC at the apron. Similarly, the AGV that delivered an inbound container to an ASC can receive another outbound container instead of traveling empty to the apron.

In this paper, two different dispatching strategies for AGVs, "dedicated dispatching" and "pooled dispatching", are considered. In dedicated dispatching, every AGV is assigned to a single CC. In most conventional container terminals, the dedicated dispatching strategy is usually used for dispatching yard trucks. In this case, the control logic becomes simple, although the productivity of delivery by AGVs may decrease because of the longer empty travels.

In pooled dispatching, an AGV performs delivery tasks for more than one CC. For example, an AGV that delivered a container from a yard (apron) to an apron (yard) for a CC can deliver another container from an apron (yard) to a yard (apron) for another CC. Though the control logic for the pooled strategy is a little more complicated than that for the dedicated strategy, higher productivity of ship operations is expected.

For port container terminals, Durrant-Whyte [2] described the design of an autonomous guided vehicle system to transport containers in a port environment. Evers and Koppers [6] proposed a distributed control architecture, which utilizes a signaling concept for traffic control, for an AGV system. Their research was aimed at a situation where a large number of vehicles move within the same infra-structural facility.

Most previous studies [3, 4, 9, 10, 11] on AGV dispatching methods have assumed that pickup calls are issued randomly and that the sequence of calls cannot be known in advance. Thus, the dispatching decision is made after a pickup call is issued, or when a vehicle becomes free from a previous delivery task. Anwar and Nagi [1] and Ihsan and Hommertzhein [7] treated the dispatching problem as a scheduling problem in which travels of AGVs as well as the operations of machines are scheduled.

In the case of the ship operation, delivery tasks are known in advance, and the sequence of the tasks is predetermined. This paper discusses how to preplan the dispatching by utilizing the information about the sequence of tasks. Recently, Kim and Bae [8] proposed the dispatching method for AGVs during ship operations in automated container terminals. However, they dealt with the dispatching problem for AGVs serving a single CC. In this study, AGVs are assumed to serve multiple container cranes simultaneously, which is called "pooled dispatching".

In the next section, we introduce a ship operation that utilizes AGVs in container terminals. Next, a mixed integer programming model is formulated for the dispatching problem. In section 3, a heuristic algorithm is suggested. Finally, in section 4, performances of various dispatching strategies are compared with each other through a numerical experiment.

2. THE SHIP OPERATION AND THE PROBLEM FORMULATION

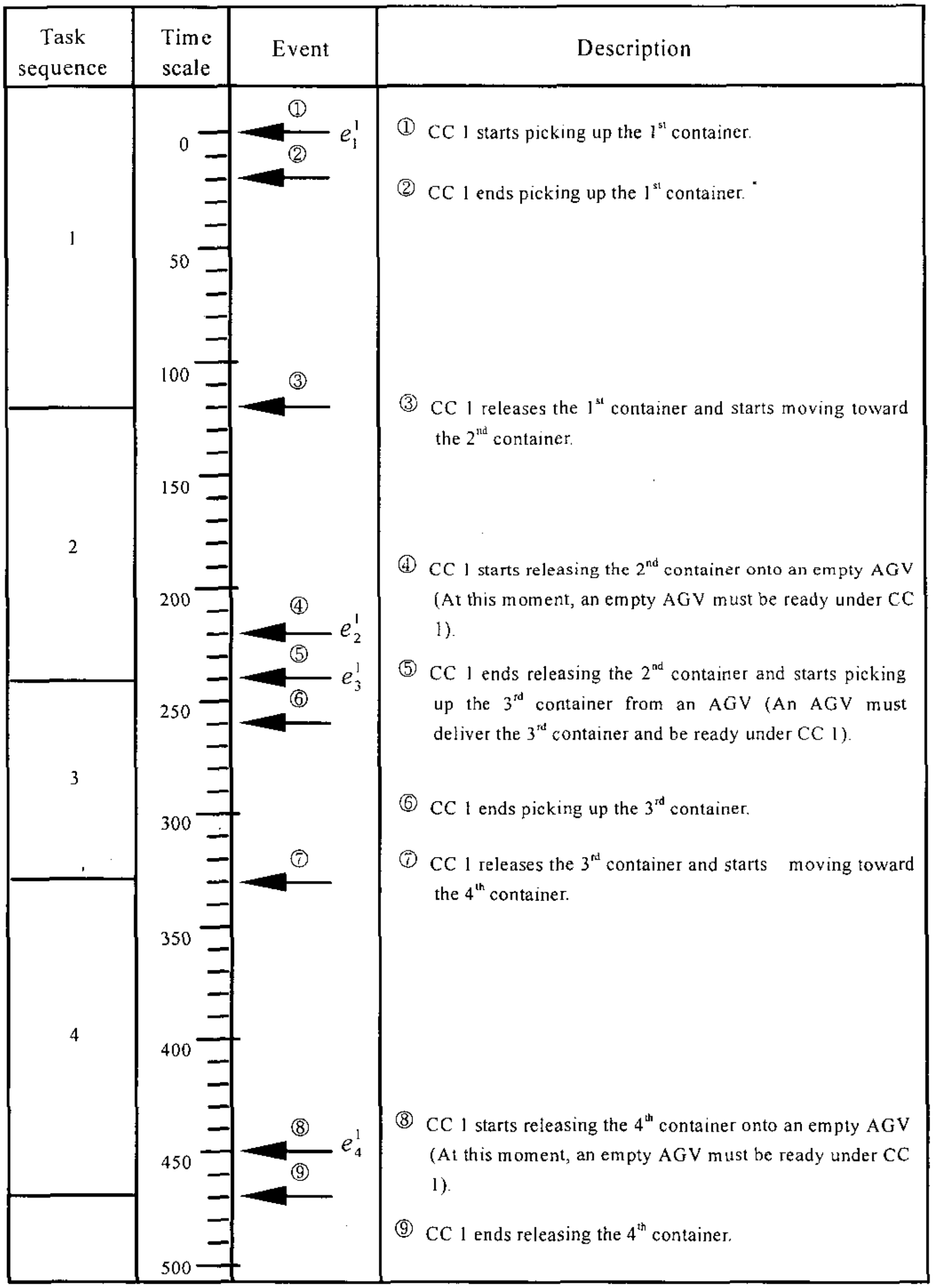
2.1 The Seaside Operation of a CC

A loading operation cycle by a CC begins with a pick-up of a container from an AGV, while a discharging operation cycle ends with the release of a container onto an AGV. Table 1 shows an example of sequence lists of the ship operations of two CCs (CC 1 and CC 2), which are performed in a dual-cycle. The progress of operations by CC 1, which performs operations in Table 1, can also be represented as shown in Figure 2. Note that Figure 2 assumes that the operations are not delayed due to late arrivals of AGVs. In order for an operation of a CC to be performed without a delay, an AGV must be ready before the transfer of a container begins at a specified location under the corresponding CC.

Table 1. An example of a working sequence list

CC 1						CC 2					
Task Seq.	Type*	Ship location†	Yard Location‡	Operation cycle time	s_i^1 ††	Task Seq.	Type*	Ship location†	Yard Location‡	Operation cycle time	s_i^2 ††
1	L	12/03/04	C/21/3/2	120	0	1	L	19/05/02	B/07/3/2	135	0
2	D	12/04/10	B/17/4/3	120	220	2	D	19/06/12	B/17/3/2	135	250
3	L	12/03/06	C/07/4/1	110	240	3	L	19/05/04	B/11/3/2	90	270
4	D	12/04/08	C/21/3/1	120	450	4	D	19/06/10	C/11/3/2	110	450
:	:	:	:	:	:	:	:	:	:	:	:

* L : loading, D : discharging † ship-bay no./ row no./ tier no (working position for ship-bay 12 and 19 is "A w/p" and "C w/p", respectively). ‡ yard block / yard-bay no. / row no./ tier no. †† earliest possible event time without delay



* We assume that the release time and the pickup time of a container by a CC is 20 seconds, respectively.

Figure 2. The progress of the ship operation of CC 1 and events

Let e_i^k be an event representing the moment that an AGV transfers the i^{th} container of CC k (the i^{th} operation of CC k). Event e_i^k corresponds to the beginning of the pickup of a container from an AGV when loading is the i^{th} operation, while it corresponds to the beginning of the release of a container onto an AGV when discharging is the i^{th} operation. The event time of e_i^k is denoted by y_i^k. The delay of an operation occurs when the corresponding AGV does not arrive at the requested moment. When there is no delay of operations in the event as shown in Figure 2, the value of y_i^k is denoted by s_i^k, which we call the earliest possible event time. The event time, y_i^k, is a decision variable in this model. Table 1 also illustrates the earliest possible event times (s_i^k).

The notations related to seaside operations are summarized as follows:

K : The set of CCs involved in a seaside operation.

m_k : The number of tasks for the ship operation assigned to CC $k\ (k \in K)$.

m : The total number of all tasks assigned to CCs ($m = \sum_{k\in K} m_k$).

s_i^k : The earliest possible event time of e_i^k.

y_i^k : The actual event time of e_i^k (a decision variable). When there is no delay in the operation of CC k, $y_i^k = s_i^k$.

2.2 Landside Operations of AGVs

When the dual-cycle strategy is applied to a landside operation, the total travel time of AGVs becomes different for different assignment of delivery tasks. When an AGV, which is delivering a container, is ordered to pick up another container at a location close to the delivery location, the empty travel time can be significantly reduced.

This dispatching problem has the following three special properties that must be considered in developing a solution procedure:

(1) Tasks must be carried out in the exact same order that is predetermined as in a sequence list. Planners in terminals make a discharging and loading sequence list before the ship operation begins. And the sequence list is confirmed by the corresponding shipping company. Then, the ship operation is performed in the exact same order as specified in the sequence list.

(2) The objective of minimizing delays of operations by CCs has a higher priority

than the objective of minimizing the total travel time of AGVs. This means that a higher priority is given to the operation of CCs. Because a CC is much more expensive than an AGV or a yard crane, a CC is usually a bottleneck resource in port container terminals. Thus, the main objective of the operation of AGVs and yard cranes is to support CCs so that CCs could be fully utilized. This is why minimizing delays of operations by CCs was assumed to have a higher priority than the objective of minimizing the total travel time of AGVs.

(3) A delay in a seaside operation of a CC results in delays, by the same amount of time, of all succeeding seaside operations assigned to the same CC.

Figure 3 illustrates the layout of a port container terminal, which was used in the numerical experiment of this study. Because ASCs are used as yard cranes, pickup and drop-off (P/D) points are located in front of each block. It is also assumed that the location of a CC does not change until all tasks under consideration for a ship-bay are completed. The travel time between every combination of P/D points is provided in Table 2. In the example, there are five working positions of CCs in the apron (A – E w/p) and five yard blocks in the marshalling yard (A – E block).

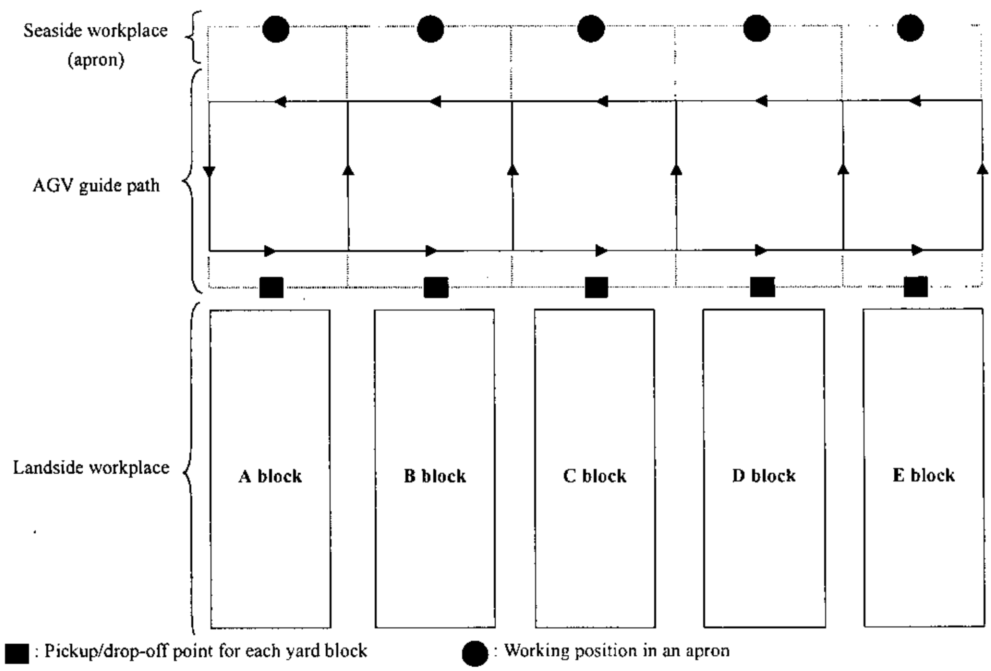


Figure 3. The layout of a hypothetical container terminal

Table 2. The travel time between transfer locations in the example (unit: seconds)

From \ To	A block	B block	C block	D block	E block	A w/p†	B w/p	C w/p	D w/p	E w/p
A	-	30	60	90	120	205	195	235	225	265
B	80	-	30	60	90	175	165	205	195	235
C	110	80	-	30	60	145	135	175	165	205
D	140	110	80	-	30	175	165	145	135	175
E	170	140	110	80	-	205	195	175	165	145
A w/p	205	175	145	175	205	-	50	90	80	120
B w/p	215	185	155	185	215	10	-	80	70	110
C w/p	235	205	175	145	175	30	20	-	50	90
D w/p	225	215	185	155	185	40	30	10	-	80
E w/p	265	235	205	175	145	60	50	30	20	-

† w/p: working position in the apron

There are three types of events that AGVs experience during a ship operation: The starting event, when an AGV starts to move for the first operation; the event when an AGV begins to receive a container from a CC or when an AGV begins to transfer a container to a CC; and the stopping event, when an AGV completes all of its assigned tasks.

For the operation of AGVs, the following notations are introduced:

V : The set of AGVs.

e_j^O : The starting event of AGV j, $j \in V$.

e_j^F : The stopping event of AGV j, $j \in V$.

e_i^k : The event that corresponds to the beginning of a pickup (or release) of a container from (onto) an AGV for the task related to the i^{th} operation of CC k. This is the same event as e_i^k in Figure 2.

$l(e_i^k)$: The location where the event e_i^k occurs. $l(e_i^O)$ represents the initial location of AGV j. $l(e_i^k)$ represents the position where the i^{th} container of CC k will be transferred. $l(e_i^F)$ is the location where AGV i completes its final delivery task.

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

c_{ki}^{lj} : The time required for an AGV to be ready for e_j^l after it experiences e_i^k. 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 CC k by CC k. c_{ki}^{lj} includes the travel time from the apron to the location of the next container (the j^{th} container of CC l) in a marshalling yard, the release time of the container by an ASC, and the travel time of the AGV to CC l. Note that c_{ki}^{lj} does not depend on the event time, y_i^k or y_j^l, but depends only on the route from $l(e_i^k)$ to $l(e_j^l)$. Thus, c_{ki}^{lj} is a constant value.

In this paper, we address the problem of solving a static scheduling problem where AGVs must be assigned to complete a known set of tasks. It is assumed that there is an available fleet of AGVs that are to be scheduled for a set of known tasks.

Let S and D be the sets of e_j^O and e_j^F, $j \in V$, respectively. And let T be the set of e_i^k, where $k \in K$ and $i = 1,\cdots,m_k$. Then, a feasible dispatching decision is 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 which becomes 1 if e_i^k is assigned to e_j^l, $k \in K'$ and $l \in K''$. For k, $l \in K$, the assignment of e_i^k to e_j^l implies that the AGV just delivered the i^{th} container of CC k is scheduled to deliver the j^{th} container of CC l.

Let α be the travel cost per unit time of an AGV, and β be the penalty cost per unit time for the delay of the completion time. It is assumed that $\alpha << \beta$. And let m_O and m_F equal to $|V|$. Then, the dispatching problem can be formulated as follows:

$$\text{Minimize} \quad \alpha \sum_{k\in K'} \sum_{i=1}^{m_k} \sum_{l\in K''} \sum_{j=1}^{m_l} t_{ki}^{lj} x_{ki}^{lj} + \beta \sum_{k\in K} (y_{m_k}^k - s_{m_k}^k)^+ \tag{1}$$

$$\text{Subject to} \quad \sum_{l\in K''} \sum_{j=1}^{m_l} x_{ki}^{lj} = 1, \qquad \text{for } \forall k \in K' \text{ and } i = 1,\cdots,m_k \tag{2}$$

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

$$y_j^l - (y_i^k + c_{ki}^{lj}) \geq M(x_{ki}^{lj} - 1), \quad \text{for } \forall k \in K', l \in K,\ i = 1,\cdots,m_k, \text{ and } j = 1,\cdots,m_l \tag{4}$$

$$y_{i+1}^k - y_i^k \geq s_{i+1}^k - s_i^k, \qquad \text{for } \forall k \in K \text{ and } i = 1,\cdots,m_k - 1 \tag{5}$$

$$y_i^k \geq s_i^k, \qquad \text{for } \forall k \in K' \text{ and } i = 1,\cdots,m_k \tag{6}$$

$$x_{ki}^{lj} = 0 \text{ or } 1, \qquad \text{for } \forall k \in K', \ \forall l \in K'', \ i = 1, \cdots, m_k, \text{ and } j = 1, \cdots, m_l \qquad (7)$$

where $s_i^O = 0$ for all $i \in V$.

Because $\alpha << \beta$, the sum of delays of CC operations will be minimized first. For the same value of the sum of the delays, the total travel distance of the AGVs 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 the fact that two events that are served consecutively by the same AGV must be set apart by at least the time required by an AGV for travel and load transfer between the two events. That is, x_{ki}^{lj} can be 1 only if $y_j^l - y_i^k \geq c_{ki}^{lj}$. Note that $y_i^O = 0$ for all $i \in V$ and x_{ki}^{Fj}, $k \in K$, is not restricted by constraint (4). Constraint (5) implies that two events that are served by the same CC must be set apart by at least the time required for the CC to perform all the movements between the two events. Constraint (6) represents that the actual event time is always more than or equal to the earliest possible event time.

An example problem is solved to illustrate the solution. Four AGVs are assumed to participate in a ship operation that consists of the first four tasks for each CC shown in Table 1. Using LINDO®, formulations (1) – (7) are solved. Let CC 1 and CC 2 be respectively located at A w/p and C w/p in the apron

For the problem shown in Table 1, the optimal solution is (y_1^{1*}, y_2^{1*}, y_3^{1*}, y_4^{1*}, y_1^{2*}, y_2^{2*}, y_3^{2*}, y_4^{2*}) = (310, 530, 550, 760, 390, 640, 660, 840). $x_{O1}^{21*} = x_{O2}^{11*} = x_{O3}^{23*} = x_{O4}^{13*} = x_{11}^{12*} = x_{12}^{F2*} = x_{13}^{14*} = x_{14}^{F4*} = x_{21}^{22*} = x_{22}^{F1*} = x_{23}^{24*} = x_{24}^{F3*} = 1$, and all the other $x_{ki}^{lj*} = 0$. The solution can be interpreted as follows:

AGV 1: Idle at the apron (in front of CC 1) → task 1 of CC 2 (delivering an outbound container from block B to CC 2) → task 2 of CC 2 (delivering an inbound container from CC 2 to block B). Therefore, the total travel time $= t_{O1}^{21} + t_{21}^{22} + t_{22}^{F1} = 585$.

AGV 2: Idle at the apron (in front of CC 1) → task 1 of CC 1 (delivering an inbound container from CC 1 to block C) → task 2 of CC 1 (delivering an outbound container from block B to CC 1). Therefore, the total travel time $= t_{02}^{11} + t_{11}^{12} + t_{12}^{F2} = 465$.

AGV 3: Idle at the apron (in front of CC 2) → task 3 of CC 2 (delivering an outbound container from block B to CC 2) → task 4 of CC 2 (delivering an inbound container from CC 2 to block C). Therefore, the total travel time $= t_{03}^{23} + t_{23}^{24} + t_{24}^{F3} = 595$.

AGV 4: Idle at the apron (in front of CC 2) → task 3 of CC 1 (delivering an outbound container from block C to CC 1) → task 4 of CC 1 (delivering an inbound container from CC 1 to block C). Therefore, the total travel time $= t_{04}^{13} + t_{13}^{14} + t_{14}^{F4} = 475$.

Table 3. Delays of event times in the example

CC number	Event	Event time without delay (s_i^k)	Event time considering delays (y_i^k)	Delay
1	e_1^1	0	310	310
	e_2^1	220	530	310
	e_3^1	240	550	310
	e_4^1	450	760	310
2	e_1^2	0	390	390
	e_2^2	250	640	390
	e_3^2	270	660	390
	e_4^2	450	840	390

The total travel time of the 4 AGVs is 585 + 465 + 595 + 475 = 2120 time units. The sum of the delays of the CCs is 700 time units, where the delay of CC 1 is 310 (= 760 – 450) and the delay of CC 2 is 390 (= 840 – 450). Table 3 shows the resulting delays of various events in the example.

3. A HEURISTIC ALGORITHM

In this section, a heuristic algorithm for solving the dispatching problem is suggested. Suppose that y_i^k's (which are equal to s_i^k's in the initial stage), $k = 1, \ldots, n$, $i = 1, \ldots, m_k$, are given and the events are sequenced in the increasing order of y_i^k. We will denote the j^{th} event in the sequence as "event (j)", the event time of event (i) as y_i, the time required for an AGV to be ready for event (j) after it experiences event (i) as c_{ij} (which corresponds to the notation of c_{ki}^{lj}), the pure travel time from the location of event (i) to the location of event(j) as t_{ij}, and the decision variable for the assignment of event (i) to event (j) as x_{ij}. Let T_ξ be a subset of T, respectively, which includes only the first ξ events in the sequence. Then, the constraint subset ξ of constraints (2) – (4) can be written as follows:

(Constraint subset ξ)

$$\sum_{i \in S \cup T_\xi} x_{ij} = 1, \qquad \text{for all } j \in D \cup T_\xi \tag{8}$$

$$\sum_{j \in D \cup T_\xi} x_{ij} = 1, \qquad \text{for all } i \in S \cup T_\xi \tag{9}$$

$$y_j - (y_i + c_{ij}) \geq M(x_{ij} - 1) \qquad \text{for all } i \in S \cup T_\xi \text{ and } j \in T_\xi \tag{10}$$

$$x_{ij} = 0 \text{ or } 1 \qquad \text{for all } i \in S \cup T_\xi,\ j \in D \cup T_\xi \tag{11}$$

In the algorithm, for given values of y_i^k's, the feasibility is checked one by one from the constraint subset 1 to the constraint subset m. 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 constaint (10). During iterative procedures of the following algorithm, it is tries to minimize the second term of objective function (1) by increasing y_i^k's by the least possible amount. However, after a feasible solution to equation constraint subset m, which is equisalent to constraints (2) and (3), is found, the total travel time of AGVs, which is the first term of objective function (1), will be minimized by solving the assignment problem.

The heuristic solution procedure is summarized in the following:

Step 1: (Initializing)
Set $y_i^k = s_i^k$, $i = 1, \cdots, m_k$ for $k \in K$. And set $y_i^O = 0$ for all $i \in V$. $\xi = 0$.

Step 2: (Next task)
$\xi = \xi + 1$. If $\xi > m$, go to step 5. Otherwise, sequence the events in the increasing order of y_i^k and go to step 3.

Step 3: (Feasibility checking)
Check the existence of a feasible solution to constraints (8) – (11). If there is a feasible solution, go to step 2. Otherwise, go to step 4.

Step 4: (Delaying event times)

Let $\pi_{i^*\xi} = \min_{i \in S \cup T_{\xi-1}} [\max\{c_{i\xi} - (y_\xi - y_i), 0\}]$. Let the event time of event (ξ) be y_λ^γ. Then $y_j^\gamma = y_j^\gamma + \pi_{i^*\xi}$ for all $j \geq \lambda$. Go to step 3.

Step 5: (Task assignment)

Evaluate t_{ij}. Solve the assignment problem with the objective of minimizing the total travel distance and constraint subset m. Stop.

Feasibility checking

For given values of y_i^k's, the feasibility can be checked by solving a maximum-cardinality-matching problem in the bipartite graph [5]. When the maximum cardinality is the same as $|S \cup T|$, it can be concluded that the constraint subset ξ, (8) - (11), has a feasible solution. Otherwise, it is infeasible.

Delaying event times

In order to make an infeasible constraint subset feasible, one or more x_{ij} must be additionally allowed to be 1 in constraint (10). Note that, according to the solution procedure, constraint subset ξ must be made feasible before proceeding to check the feasibility of constraint subset ξ+1. The questions, which must be answered for making constraint subset ξ feasible are "which x_{ij} should be additionally allowed to be 1," "which y_i's should be increased," and "how much should they be increased in order to make the intended x_{ij} be allowed to be 1." In this heuristic algorithm, the event time of event (ξ) is delayed by the minimum amount possible for making at least one x_{ij} to be allowed to be 1. In the process, when the current constraint subset becomes feasible, the iterations are stopped, and we proceed to the next stage.

The two main steps of feasibility checking and delaying events can be interpreted as a heuristic process of solving the following problem:

Minimize $\quad (y_\lambda^\gamma - s_\lambda^\gamma)^+ \qquad (12)$

Subject to $\quad$ constraints (2) – (7).

It is assumed that the dual-cycle ship operations in Table 1 are performed by four AGVs in the pooled dispatching manner. During the entire process of the algorithm, y_i^k's are increased, and certain $x_{i\xi}$'s are allowed to be 1 in order to make the constraint subset ξ feasible. Using the heuristic algorithm described above, the result obtained is the same as the optimal solution obtained from solving equations (1) – (7), in terms of the total delay time of CCs and the total travel time. The numerical description given below is a step-by-step illustration of the algorithm for the example.

Step 1. (Initialization) From Table 1, ($y_1^1, y_2^1, y_3^1, y_4^1, y_1^2, y_2^2, y_3^2, y_4^2$) = (0, 220, 240, 450, 0, 250, 270, 450). Set $\xi = 0$.

Step 2. (Next task) $\xi = \xi + 1 = 1$.

Step 3. (Feasibility checking) y_i^k is sorted as e_1^1, e_1^2, e_2^1, e_3^1, e_2^2, e_3^2, e_4^1, and e_4^2. Calculate c_{ki}^{lj}'s. There is no feasible solution to constraint subset 1.

Step 4. (Delaying event times) We will denote the i^{th} AGV in the initial state (location) as $\bar{i}$. Then, $\pi_{i1} = (\pi_{\bar{1}1}, \pi_{\bar{2}1}, \pi_{\bar{3}1}, \pi_{\bar{4}1})$ = (310, 310, 350, 350). Thus, $\pi_{i^*1} = \pi_{\bar{1}1}$ = 310, which results in ($y_1^1, y_2^1, y_3^1, y_4^1, y_1^2, y_2^2, y_3^2, y_4^2$) = (310, 530, 550, 760, 0, 250, 270, 760).

Step 3. (Feasibility checking) y_i^k is sorted as e_1^2, e_2^2, e_3^2, e_1^1, e_4^2, e_2^1, e_3^1, and e_4^1. There is a feasible solution to constraint subset 1.

Step 2. (Next task) $\xi = \xi + 1$.

Step 3. (Feasibility checking) There is no feasible solution to constraint subset 2.

Step 4. (Delaying event times) $\pi_{i^*2} = (\pi_{\bar{1}2}, \pi_{\bar{2}2}, \pi_{\bar{3}2}, \pi_{\bar{4}2})$ = (390, 390, 430, 430). Thus, $\pi_{i^*2} = \pi_{\bar{1}2}$ = 390, which results in ($y_1^1, y_2^1, y_3^1, y_4^1, y_1^2, y_2^2, y_3^2, y_4^2$) = (310, 530, 550, 760, 390, 640, 660, 840).

Step 3. (Feasibility checking) y_i^k is sorted as e_1^1, e_1^2, e_2^1, e_3^1, e_2^2, e_3^2, e_4^1, and e_4^2. There is a feasible solution to constraint subset 2.

This process should be repeated until ξ becomes greater than 8.

4. NUMERICAL EXPERIMENTS

In order to compare the optimal solutions from formulations (1) – (7) with those from the heuristic algorithm, 10 randomly generated problems are solved. Formulations (1) – (7) were solved by a software called LINDO®, on a personal computer with a pentium II MMX 266 processor. Also, in all the experiments described in this section, the data for the travel time among P/D points and the release and pickup times of CCs and ASCs are the same as those in the example in section 2. In each generated problem, it was assumed that 2 CCs and 2 AGVs per each CC are used and 5 dual-cycle operations are assigned to each CC. Also, the pooled dispatching strategy is adopted for the experiments. The travel times between locations in Table 2 are used for the test.

Table 4. Comparison of the heuristic method with the mixed integer programming model

No.	Mixed integer programming		Heuristic algorithm		Sum of operation times (E)	Performance	
	A†	B‡	C†	D‡		(D - B) / E	C / A
1	1850	810	0.13	910	1090	0.092	0.0001
2	3460	730	0.11	730	980	0.000	0.0000
3	2014	510	0.11	650	1130	0.124	0.0001
4	2235	530	0.10	530	1230	0.000	0.0000
5	1015	540	0.12	540	1180	0.000	0.0001
6	1307	1110	0.16	1400	860	0.337	0.0001
7	1315	990	0.12	1160	970	0.175	0.0001
8	451	610	0.10	660	1170	0.043	0.0002
9††	-	-	0.12	500	1210	-	-
10	3241	560	0.09	560	1080	0.000	0.0000
Average	1876.4	710.0	0.12	764	1090	0.086	0.0001

† : computational time (unit: seconds) ‡ : objective value

†† : This problem could not be solved by LINDO® because of a memory shortage.

Table 4 shows the results of the experiment. From column A of Table 4, it can be seen that the computational time of LINDO® is too excessive to be used in real time. Comparing column B with column D, it was found that the heuristic algorithm provides results comparable to those obtained by LINDO®. Dividing the difference between the two methods in the objective value by the sum of the operation times, the ratio ((D – B) / E) is less than 10% on the average. Considering that the computational time of the heuristic algorithm is negligible, compared with that of LINDO®, it can be concluded that the heuristic algorithm shows very satisfactory performance.

Various numerical experiments were conducted to compare different types of seaside and landside operations and dispatching rules. The heuristic algorithm suggested in this paper was used for dispatching AGVs.

For one CC, the number of AGVs varies from 2 to 5, and the numbers of tasks for the test problems are 30. For each combination of conditions, ten problems are generated randomly. The dual-cycle seaside operation is expected to reduce the total operation time of a CC and the completion time of the entire ship operation. In order to quantify the ratio of the dual-cycle operation to the entire ship operation, the concept of "length of runs" is introduced. A "length of runs" is the number of consecutive ship operations of the same type (loading or discharging). A short length of runs indicates a high ratio of dual-cycle operations to all of a ship's operations. In order to test the effects of the average length of runs on the total operation time and the completion time of an entire ship operation, ten different series of operation times for a single CC are generated at each different level of the average length of runs.

Figure 4 shows that the total operation time increases as the average length of runs increases. The ratio of the minimum total operation time to the maximum total operation time was 84%. The completion time of a ship operation also decreases as the average length of runs decreases. However, when the number of AGVs is far below the level required (when the number of AGVs is one in Figure 5), a high ratio of dual-cycle operations does not reduce the completion time, as shown in Figure 5.

In the experiment, the number of AGVs varies from 1 to 5, and the numbers of tasks for the test problems are 15, 20, 25, and 30. The number of CCs varies from 2 to 4. For each combination of conditions, ten problems are generated randomly. Thus, the total number of problems solved is 600. In order to measure the effects of different strategies, four scenarios considered in the experiment are as follows:

(1) Scenario A: Each CC performs the ship operation in a single-cycle, and the pooled dispatching of AGVs is applied to the ship operation.
(2) Scenario B: Each CC performs the ship operation in a single-cycle, and the dedicated dispatching of AGVs is applied to the ship operation.
(3) Scenario C: Each CC performs the ship operation in a dual-cycle, and the pooled dispatching of AGVs is applied to the ship operation.
(4) Scenario D: Each CC performs the ship operation in a dual-cycle, and the dedicated dispatching of AGVs is applied to the ship operation.

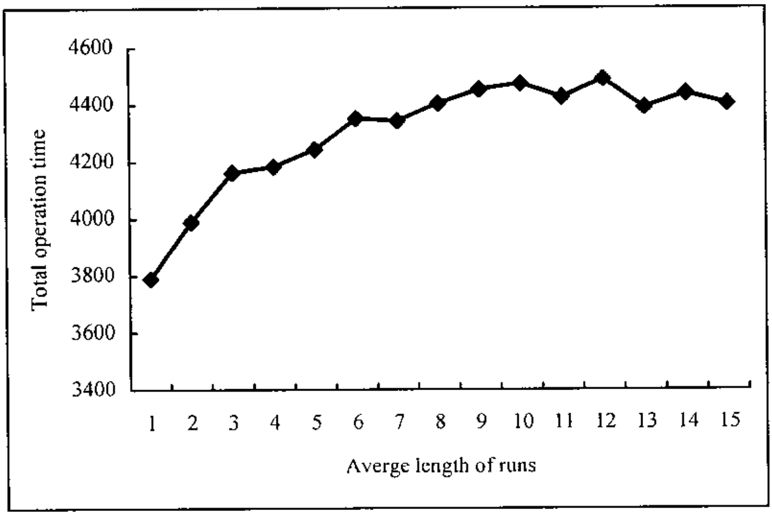


Figure 4. The effects of the average length of runs on the total operation time of a CC

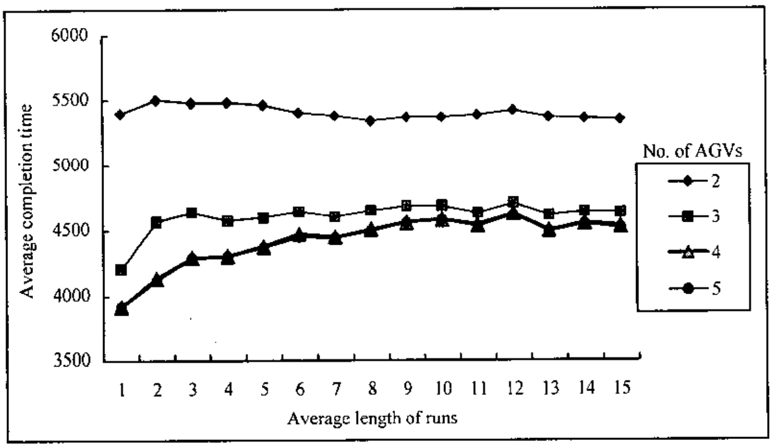


Figure 5. The effects of the average length of runs on the average completion time

According to Figure 6, when the number of AGVs exceeds a level required for a ship operation, the differences among the different strategies are negligible. However, when the number of AGVs is not sufficient for a ship operation, the performance of the pooled dispatching of AGVs significantly outperforms that of the dedicated dispatching of AGVs. The two dispatching strategies also show a larger difference in the case of the single-cycle operation of CCs than in the case of the dual-cycle operation of CCs.

Figure 7 shows how the average completion time changes as the number of

AGVs changes for each of the four different scenarios. As a whole, the average completion time in case of the dual-cycle operation of a CC was much lower than that of the single-cycle operation. Also, the difference in the average completion time between the pooled and the dedicated dispatching method was higher in case of the single-cycle operation of a CC than in the dual-cycle operation of a CC. The difference is due to the fact that the dual-cycle operation in a yard is possible in scenario D, but not in scenario B.

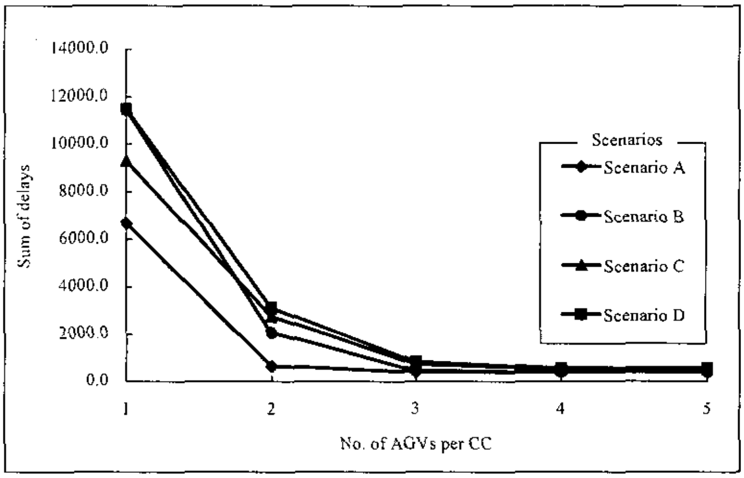


Figure 6. A comparison of the average delay under different dispatching strategies

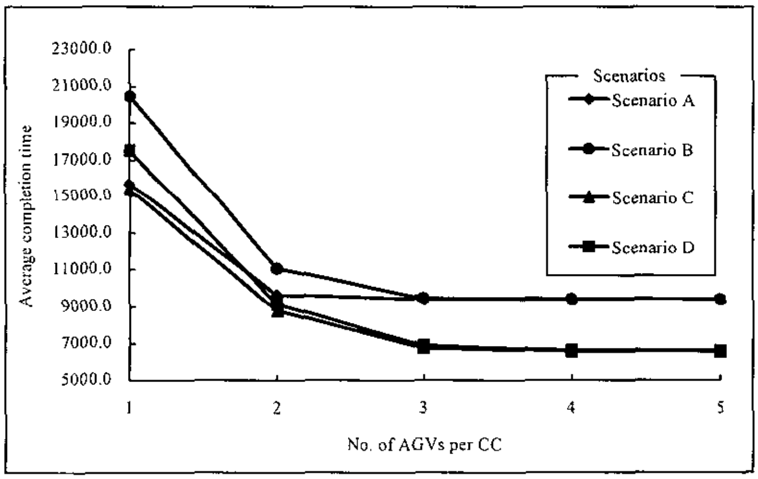


Figure 7. Changes in the average completion time for four operation scenarios

From Figure 7, it can be seen that, compared with the single-cycle operation, the dual-cycle operation of a CC can significantly reduce the completion time of a ship operation. As in the case of the sum of delays, the effects of the pooled dispatching of AGVs are higher in the case of the single-cycle operation of a CC than that in the case of the dual-cycle operation.

In Figure 8, it is shown that the total travel distance is reduced by adopting the pooled strategy in the case of the single-cycle seaside operation, while there is no difference between the two discharging strategies in the case of the dual-cycle seaside operation.

The distribution of containers in a yard is expected to be one of the important factors that affect the efficiency of ship operations. The study examined how the number of blocks, where containers for ship operations are located, affects the sum of delays. It was assumed that two CCs, and three AGVs per CC, are in operation. The number of blocks per CC varies from 1 to 5. For each combination of the number of blocks and operation scenarios, 10 problems are generated randomly and solved.

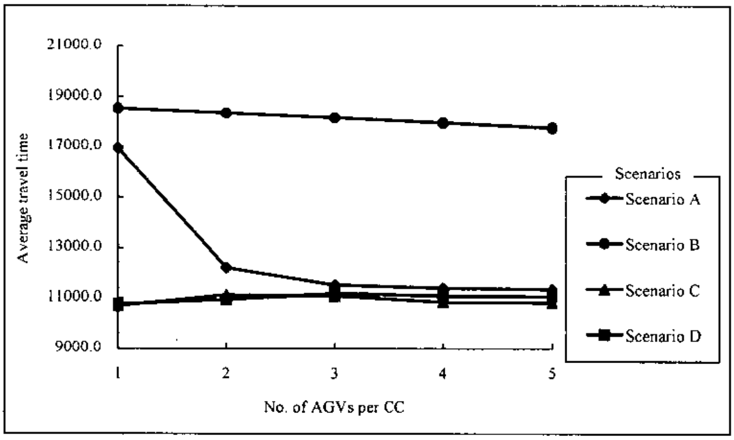


Figure 8. Changes in the average travel time for four operation scenarios

Figure 9 shows how the sum of delays changes as the number of blocks increases. When the ship operation is performed in a single-cycle, the number of blocks does not affect the sum of the delays. However, in the case of the dual-cycle seaside operation, the sum of the delays increases as the number of blocks increases. Considering that the completion time of the ship operation is shorter in the case of the dual-cycle seaside operation than in the case of the single-cycle seaside operation, it is logical that the increases in the travel distance resulting from the increased number of blocks is directly reflected in the sum of the delays in scenarios C and D.

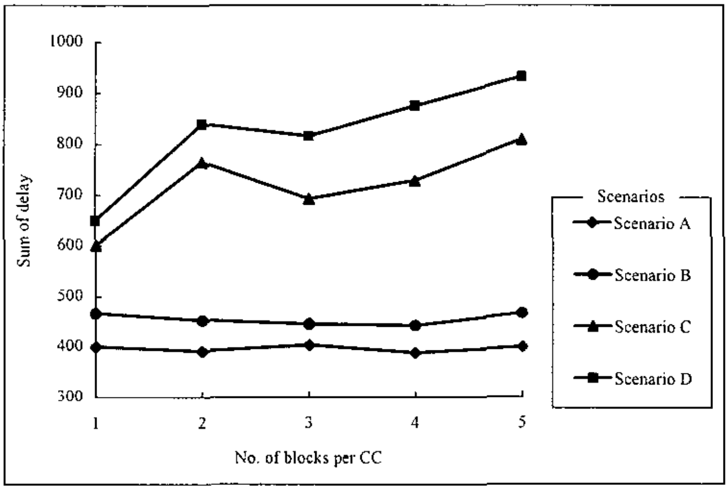


Figure 9. Changes in the average delay time as a function of the number of blocks

5. SUMMARY AND CONCLUSIONS

This paper discusses how to assign container-delivery tasks to AGVs during the ship operation in automated container terminals. Because there is no temporary buffering function between a CC and an AGV during the seaside operation, a delay in the arrival of an AGV directly causes a delay by a CC in the ship operation. A previous study [8] on the dispatching of AGVs for a CC is extended to a case of multiple CCs. A mathematical formulation for the dispatching problem is suggested and a heuristic algorithm is developed. It was found that the pooled dispatching strategy contributes significantly to reducing delays in the ship operation. The effects of pooled dispatching on the travel distance of AGVs were higher in the case of the single-cycle operation of a CC than in the case of the dual-cycle operation of a CC.

In this study, all the operation times are assumed to be deterministic. However, in the real world, they are stochastic. Thus, it is necessary to examine the performance of the algorithm described in this paper under a stochastic environment.

Acknowledgement: This study is partially supported by funds (986-1000-002-2) of the Korea-Japan Basic Scientific Promotion Program (KOSEF).

REFERENCES

[1] Anwar, M. F. and R. Nagi, "Integrated scheduling of material handling and manufacturing activities for just-in-time production of complex assemblies," *International Journal of Production Research* 36 (1998), 653-681.

[2] Durrant-Whyte, H. F., "An Autonomous Guided Vehicle for Cargo Handling Applications," *International Journal of Robotics Research* 15 (1996), 407-440.

[3] Egbelu, P. J. and J. M. A. Tanchoco, "Characterization of automatic vehicle dispatching rules," *International Journal of Production Research* 22 (1984), 359-374.

[4] Egbelu, P. J., "Pull versus push strategy for automated guided vehicle load movement in a batch manufacturing system," *Journal of Manufacturing Systems* 6 (1987), 209-221.

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

[6] Evers, J. J. M. and S. A. J. Koppers, "Automated guided vehicle traffic control at a container terminal," *Transportation Research - A* 30 (1996), 21–34.

[7] Ihsan, S. and D. L. Hommertzheim, "Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system," *International Journal of Production Research* 30 (1992), 1059–1079.

[8] Kim, K. H. and J. W. Bae, "A Dispatching Method for Automated Guided Vehicles to Minimize Delays of Containership Operations," *International Journal of Management Science* 5 (1999), 1–26.

[9] Klein, C. M. and J. Kim, "AGV dispatching," *International Journal of Production Research* 34 (1996), 95–110.

[10] Russell, R. and J. M. A. Tanchoco, "An evaluation of vehicle dispatching rules and their effect on shop performance," *Material Flow* 1 (1984), 271-280.

[11] Yim, D. and R. J. Linn, "Push and pull rules for dispatching automated guided vehicles in a flexible manufacturing system," *International Journal of Production Research* 31 (1993), 43 - 57.