Industrial Engineering
& Management Systems
Vol 15, No 1, March 2016, pp.19-31
ISSN 1598-7248 | EISSN 2234-6473 |
http://dx.doi.org/10.7232/iems.2016.15.1.019

Flow Path Design for Automated Transport Systems in Container Terminals Considering Traffic Congestion

Ivan Kristianto Singgih, Soondo Hong, Kap Hwan Kim*
Department of Industrial Engineering, Pusan National University, Busan, Korea

(Received: November 2, 2015 / Revised: February 5, 2016 / Accepted: March 7, 2016)

ABSTRACT

A design method of the network for automated transporters mounted on rails is addressed for automated container terminals. In the network design, the flow directions of some path segments as well as routes of transporters for each flow requirement must be determined, while the total transportation and waiting times are minimized. This study considers, for the design of the network, the waiting times of the transporters during the travel on path segments, intersections, transfer points below the quay crane (QC), and transfer points at the storage yard. An algorithm, which is the combination of a modified Dijkstra's algorithm for finding the shortest time path and a queuing theory for calculating the waiting times during the travel, is proposed. The proposed algorithm can solve the problem in a short time, which can be used in practice. Numerical experiments showed that the proposed algorithm gives solutions better than several simple rules. It was also shown that the proposed algorithm provides satisfactory solutions in a reasonable time with only average 7.22% gap in its travel time from those by a genetic algorithm which needs too long computational time. The performance of the algorithm is tested and analyzed for various parameters.

Keywords: Automated Transport System, Container Terminals, Network Design, Congestion

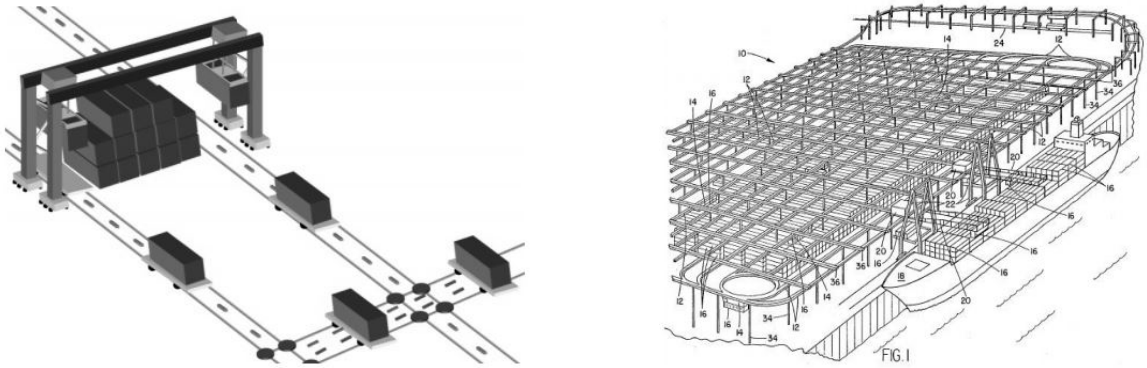
* Corresponding Author, E-mail: kapkim@pusan.ac.kr

1. INTRODUCTION

There have been new conceptual designs to overcome limitations of throughput capacities of traditional container terminals. Some of them proposed rail-based transport systems for delivering containers between the storage yard and quay cranes, where transporters move on a path guided by a rail network which provides a fixed path. Typical examples are the linear motor conveyance system (LMCS), overhead grid rail (GRAIL), and SPEEDPORT system (Kim *et al*., 2012). LCMS and GRAIL systems are illustrated in Figure 1.

In LCMS, transporters, powered by a linear motor, are used and move on rails-based guide path network. LMCS has advantages of a high positioning accuracy, a high reliability, and a robustness of the handling equipment. However, because there are a limited number of routes for transporters, the routing flexibility of the transporters is relatively low compared with truck- or AGV-based systems. In GRAIL, electric shuttles are used for the storage operations in the yard and for the delivery operations between the yard and quayside. The shuttles move above the stacks of containers and carry containers via the overhead rails. This system saves the space wasted on aisles and avoids the interference of container transporters with container stacks on the ground and with the traffic of manually operated trucks.

Figures 2 and 3 illustrate a typical rail-mounted transport system which may be used for delivering containers between QCs and the yard. In the rail-mounted



(a) Linear motor conveyance system (b) GRAIL system

Figure 1. Two conceptual transport systems proposed for automated container terminals (Kim *et al*., 2012).

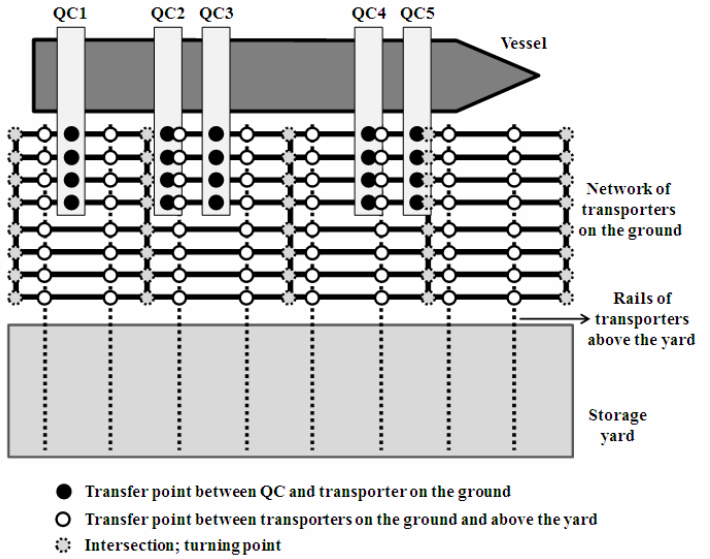


Figure 2. Top view of a new conceptual rail-mounted transport system with installed rails on the ground and above the storage yard.

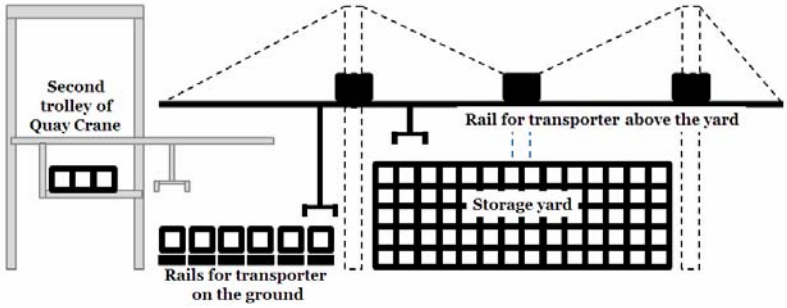


Figure 3. Side view of a new conceptual rail-mounted transport system with installed rails on the ground and above the storage yard.

transport system, which is illustrated in Figures 2 and 3, containers are delivered by transporters between the yard and QCs, which are moving on the rail installed on the ground. In the storage yard of the conceptual design, transporters are moving on the rail installed above the storage yard. During the ship operation for inbound containers, transporters pick up containers from the yard and transfer them onto transporters on the ground and then they deliver containers to pre-determined positions (transfer points) under QCs. The transportation of an outbound container is performed in the opposite direction.

For the efficient utilization of transporters, the de-

sign of guide path network is important. Rail network design is usually done before the construction stage of the terminal. Network usage design is done during the operation of the terminal. The network usage design includes the determination of the flow direction of each path segment of the physical rail network and the positions of parking slots. This study addresses the determination of the (transporter) flow direction of each path segment. The flow direction affects the travel distance and time of transporters significantly. The determination of the flow direction of each path segment should be based on the flow requirements of containers.

One problem for determining the flow directions of path segments that quay cranes (QCs) move frequently from one position to another during the discharging and loading operation in container terminals. As a result, flows of transporters for delivering containers between QCs and the yard blocks change from time to time. Thus, this paper assumes that the flow directions of path segments change whenever the flow requirements change, which includes the berthing of a new vessel, the deberthing of the vessel, the change of QC task from loading (discharging) to discharging (loading), or the movement of a QC from one ship-bay to another. This study assumes that the transporters' supervisor, in this case computer software, alters the flow directions of path segments based on changes in the flow requirements. For determining the directions of path segments of the guide path network, the route of transporters for satisfying a flow requirement, which has a specific departure position and destination, must be simultaneously determined.

Most of previous studies which are related to the guide path network design, have studied AGV guide path design. Various objectives, network structure, decision variables, and solution methods, have been studied, which are explained in detail below. The earlier studies have been conducted by Gaskins and Tanchoco (1987), Kaspi and Tanchoco (1990), and Kim and Tanchoco (1993). In these earlier studies, 0-1 integer programming models were developed while optimal solution methods like branch and bound methods were proposed.

During the determination of lane directions, the consideration of the transporters' empty travels is important, because the existence of empty travels affects significantly the total time required to complete the flow requirements or the congestion during the loaded transporters' movements. Sinriech and Tanchoco (1991), Kaspi *et al*. (2002), Lim *et al*. (2002a), Lim *et al*. (2002b), Guan *et al*. (2011), and Singgih and Kim (2015) studied the empty travels of transporters.

However, most researches did not consider the congestions, which transporters have to experience during their travel to destinations at intersections, merging positions, bi-directional path segments, and transfer (I/O) positions. Ignoring the congestion can lead to the errors in the estimation of the required travel time, which results in inefficient guide path design. Flow path design problem considering congestions has been studied by Vosniakos *et al*. (1989), Herrmann *et al*. (1995), Lim *et al*. (2002a), Lim *et al*. (2002b), Zhang *et al*. (2009), Zhang *et al*. (2011), Jeon *et al*. (2011), and Singgih and Kim (2015). Vosniakos and Davies (1989) analyzed three different AGV layouts (a bi-directional line, a bi-directional loop, and a uni-directional loop), which can be selected to serve an flexible manufacturing system (FMS). A control algorithm, which avoids the blockages on the track, was developed to improve the solution of the layout with uni-directional loop. The blocking percentage, caused by transporter interference, was used as a measure of network congestion. Herrmann *et al*. (1995) considered the design of material handling flow paths in a discrete parts manufacturing facility. A capacitated network design model was formulated and two efficient heuristics were proposed, while the flow through each arc was limited to prevent traffic congestion and account for the capacity of the arc.

Lim *et al*. (2002a), Lim *et al*. (2002b), and Jeon *et al*. (2011) used the total travel time, including waiting and interference time of transporters, as the decision criteria for determining the directions of the segments. The Q-learning technique was applied to estimate the travel times of transporters, which included the estimation of the waiting times required by the transporters. Lim *et al*. (2002a) and Singgih and Kim (2015) proposed a construction algorithm, which determined the direction of path segments in turn, starting from the one with the greatest difference between the total travel distances obtained by setting the directions in both directions, in which a method for estimating the waiting time based on queuing theory was proposed. Lim *et al*. (2002b) combined the Q-learning approach with a beam search in order to determine the path segments' directions. Zhang *et al*. (2009) searched for routes in a multi-commodity flow problem in a manufacturing or warehousing facility, which alleviated delays caused by congestions. A greedy upper bounding and Lagrangean relaxation algorithm were developed. Zhang *et al*. (2011) developed a model which simultaneously optimizes the layout and flow routing in a manufacturing facility layout design. They showed a benefit of the simultaneous consideration when confronted the workflow congestion and showed that ignoring the congestion consideration could result in a significantly poor design. Jeon *et al*. (2011) developed a simulation program, which showed that the Q-learning algorithm performs better than an approach considering shortest distance routes.

Some studies attempted to determine segments' directions in a network under the assumption that the length of time required to travel through each segment was known and deterministic. Other studies solved the problem of determining the segments' directions by combining the problem with other problems such as the facility location problems (Drezner and Wesolowsky, 2003).

Unlike from these problems, the traffic load planning problems for transportation were also addressed. Miandoabchi and Farahani (2011), Miandoabchi *et al*.

(2012a), and Miandoabchi *et al*. (2012b) analyzed a road network, in which a segment can be divided into several lanes for the travels of bus and cars. In these researches, the objectives included not only minimizing the total travel distance or time but also maximizing demand share between transportation modes, the demand coverage of the bus network, and the number of satisfied demands. Gallo *et al*. (2010) studied the urban network design problem, which is related to designing the directions of existing roads and the signal settings (the cycle length and effective green times), and proposed a scatter search algorithm and a meta-heuristic approach to solve the problem.

A machine layout problem was addressed by Al-Sultan and Bozer (1998) and Seo and Egbelu (1999). Al-Sultan and Bozer (1998) simultaneously considered the configuration of the network path and the machine assignment problem, while Seo and Egbelu (1999) studied an integrated planning model which solved the machine selection and operations sequencing, before dealing with the guide path design.

A flow path design in AGV system was studied by Shen and Lau (1997). In this system, the AGV system was analyzed in a queuing system, with the transporters and materials representing servers and customers, respectively. Shen and Lau (1997) considered the waiting time of a delivery request before being served by an AGV, during the determination of segments' directions, while in this study, the waiting times of transporters during the travel are aimed to be minimized.

The studies, stated above, considered a segment's direction as the decision variable, whereas other studies simultaneously decided whether or not to include some segments in a guide path network. Studies, which considered the inclusion of segments into the network, were Gaskins and Tanchoco (1987), Kim and Tanchoco (1993), Drezner and Wesolowsky (2003), Miandoabchi and Farahani (2011), Miandoabchi *et al*. (2012a), and Miandoabchi *et al*. (2012b).

The main contributions of this study are as follows: (1) this study proposes a method for designing the guide-path network of transporters considering congestion of transporters during the travel without using the simulation; (2) a heuristic procedure is suggested so that a near optimal design may be obtained in a short time enough to be used in a real time; (3) a shortest travel time route for each container flow is provided in the network design process. Even though Singgih and Kim (2015) proposed a similar approach, this study considers various types of operations which may be observed in container terminals and compared the performance of the proposed algorithm with a genetic algorithm, which obtains almost optimal solutions and proposed an improved heuristics determining directions of multiple path segments' at the same time in order to reduce the calculation time so that the real-sized problem may be solved in a real time.

2. PROBLEM DEFINITION AND A MATHEMATICAL FORMULATION

The performance measure of guide-path networks described in this study is the total expected travel time including waiting times during the deliveries of containers. A guide path network, shown by Figure 2, is used to represent the layout of the terminal. This study defines the problem of the guide-path design by using, so called, a critical resource network, in which nodes represent critical resources for transporter travels. Critical resources on the guide path network are stations or path segments on the network in which there may have queues of transporters and whose examples are pickup/delivery (P/D) stations and intersections. Each arc represents a path between two adjacent critical resources. The guide path network can be represented by the critical resource network as shown in Figure 2. In Figure 2, intersections, and transfer points at QC and storage yard are represented as nodes. The connections among them are represented by arcs. In all nodes, the service and waiting times are estimated, which are considered later when finding the shortest time path.

The following mathematical formulation is based on the critical resource network, while assuming that each flow requirement is performed through a single route and consists of loaded and empty travels with the same pair of source and destination nodes. One of the contributions of this study is that the mathematical formulation straightforwardly expresses the inclusion of vehicle waiting times during the total travel time calculation as the objective in this flow path design problem. The waiting time calculation method and the route selection based on the determined path segments directions are also described.

The following notations are used to describe the model:

• *Parameters and indices*

n = Number of nodes

r_k = Flow requirement k from pickup node s_k to delivery node d_k. The flow requirement is expressed by the starting position and the destination, which is represented by a QC position or a yard position, and the delivery rate (the number of moves) per unit time between the two positions.

s_k = Source node of flow requirement k

d_k = Destination node of flow requirement k

d_{ij} = Pure travel time on from node i to node j without delay

μ_j = Service rate of node j on the flow path network, which indicates the number of transporters to pass the node per unit time

• *Sets*

V = Set of nodes

F = Set of flow requirements

A = Set of all the edges, (i, j) for which $i \leq j$, in the original critical resource network. Note that edges

do not have any direction

B = Set of directed arcs. If $(i, j) \in A$, then $<i, j> \in B$ and $<j, i> \in B$.

A_c = Set of the edges for which at most one between two arcs $<i, j>$ and $<j, i>$ is allowed to be included in the final critical resource network. Note that $A_c \in A$.

• *Decision variables*

X_{ij}^k = 1, if directed arc $<i, j>$ is included in the route for flow requirement k; otherwise, 0 (decision variable)

Z_i^j = 1, if directed arc $<i, j>$ is included in the critical resource network; otherwise, 0 (decision variable)

The objective function is to minimize the total travel and waiting time of transporters, that is

$$Minimize \sum_{k \in F} r_k \left(\sum_{<i,j> \in B} X_{ij}^k (d_{ij} + W_j) \right), \quad (1)$$

where W_j represents the expected waiting time of a transporter at node j.

The constraint set includes the followings:

$$X_{ij}^k \le Z_{ij} \quad \text{for all } <i, j> \in \text{B}, \ k \in F \quad (2)$$

$$Z_{ij} + Z_{ji} \le 1 \quad \text{for } (i, j) \in A_c \quad (3)$$

$$\sum_{j \in V - \{s_k\}} X_{s_k j}^k = 1 \quad \text{for all } k \in F \quad (4)$$

$$\sum_{i \in V - \{d_k\}} X_{i d_k}^k = 1 \quad \text{for all } k \in F \quad (5)$$

$$\sum_{i \in V - \{j\}} X_{ij}^k = \sum_{l \in V - \{j\}} X_{jl}^k \quad \text{for all } j \in V, \ k \in F \quad (6)$$

$$W_j = \frac{\sum_k \sum_i X_{ij}^k r_k}{\mu_j \left(\mu_j - \sum_k \sum_i X_{ij}^k r_k \right)} \quad \text{for all } j \in V \quad (7)$$

$$X_{ij}^k, Z_{ij} = 0 \text{ or } 1 \quad \text{for all } <i, j> \in B, \ k \in F \quad (8)$$

Constraint (2) implies that transporters for each flow requirement can move through arc $<i, j>$ only when arc $<i, j>$ is included in the guide path network. Constraint (3) allows only one direction for a certain path segment on the network. Constraints (4), (5), and (6) indicate the flow conservation of each flow requirement through the network. Constraint (7) defines the expected waiting time at each critical node from M/M/1 model, where μ_j represents the service rate at node j that will be explained later and $\sum_k \sum_i X_{ij}^k r_k$ represents the arrival rate of transporters at node j.

3. A CONSTRUCTION ALGORITHM FOR DESIGNING RAIL NETWORKS

The input data are the critical resource network, the flow requirement between nodes, and the travel times on critical resources such as path segments or stations. The construction algorithm in this study attempts to determine the directions of path segments one by one. Each segment whose direction is not determined is assumed to have a pair of segments with opposite directions.

The following notations are used to describe the algorithm:

U = Set of edges whose directions are not yet determined,

D = Set of edges whose directions are determined during the solution procedure,

$T(U, D)$ = Expected total travel time when transporters move on the guide-path network consisting of (U, D) to satisfy all of the flow requirements in F during a shift under the assumption that all the edges in U have path segments directed to both directions.

The construction algorithm determines the directions of v guide path segments at a time until the directions of all the segments are fixed. In order to find the next set of path segments whose directions will be fixed, the algorithm identifies the shortest time route for each flow requirement. Considering the routes of flow requirements, the waiting time of transporters at each critical resource (intersection or P/D station) is estimated. Then, the shortest time route of each flow requirement is determined considering the updated waiting times. This procedure repeats until no more changes in the design are found. The construction algorithm in this study is an extended version of the one by Singgih and Kim (2015) and can be summarized as follows:

Step 0: (Initialize) $U = A$ and $D = \varnothing$.

Step 1: If $U = \varnothing$, stop; otherwise, let $X = U$. Find the shortest time of each flow requirement and estimate the waiting time on each node of the network with (U, D).

Step 2: If $X = \varnothing$, go to Step 4;otherwise, select an arc, (i, j) from X and let $X = X\text{-}\{(i, j)\}$. Construct two networks: one with $(U\text{-}\{(i, j)\}, D+\{<i, j>\})$ and the other with $(U\text{-}\{(i, j)\}, D+\{<j, i>\})$. If the network with $(U\text{-}\{(i, j)\}, D+\{<i, j>\})$ is infeasible, then insert $<j, i>$ into D. If the network with $(U\text{-}\{(i, j)\}, D+\{<j, i>\})$ is infeasible, then set $U = U\text{-}\{(i, j)\}$ and $D = D+\{<j, i>\}$. Go to Step 1. If both constructed networks are feasible, then go to Step 3.

Step 3: By finding the shortest time paths of all the flow requirements, evaluate the total expected travel times during a shift on both networks: $T[U\text{-}\{(i, j)\}, D+\{<i, j>\}]$ and $T[U\text{-}\{(i, j)\}, D+\{<j, i>\}]$. Let $\delta_{ij} = T[U\text{-}\{(i, j)\}, D+\{<i, j>\}] - T[U\text{-}\{(i, j)\}, D+\{<j, i>\}]$. Go to Step 2.

Step 4: Find $(s, t) = \arg\max_{i,j}\{|\delta_{ij}|\}$. $U = U-\{(s, t)\}$. If $\delta_{st} \geq 0$ and the resulting network from updating $D = D+\{<t, s>\}$ is feasible, then set $D = D+\{<t, s>\}$. Otherwise, $D = D + \{<s, t>\}$. Set $i = i +1$. Set $\delta_{st} = 0$. If $i < v$ and $U \neq \varnothing$, then go to the beginning of this step; otherwise, go to Step 1.

Setting the directions of multiple path segments simultaneously aims at reducing the computational time. During this process, if a selected path segment has only one feasible direction, then the direction of the path segment is automatically fixed first, then the next path segment is selected for the determination of the direction.

The shortest time path can be found in parallel with the estimation of the expected waiting time at each node on the network. The iterative search for the shortest time path and the calculation of waiting times are performed until it holds that |(the updated value of the total expected waiting times)−(the total expected waiting times in the previous iteration)|/(the total expected waiting times in the previous iteration) $\leq \theta$, in which θ is a certain threshold for terminating the iterative search.

Dijkstra's algorithm is used to calculate the shortest time path. To find the shortest time path instead of the shortest length path, the waiting times in the nodes, which transporters pass through, are also considered. Fu *et al*. (2006) mentioned that Dijkstra's algorithm is a label-setting algorithm, where the node with the smallest label is selected in order to find the shortest paths to the nodes. The procedure to find the shortest time paths of all the flow requirements may be described as follows:

Step 0: Set the expected waiting times of all the nodes to be zero.
Step 1: Determine the shortest time path of each flow requirement (Find Shortest Path) considering the expected waiting time at each node by using the Dijkstra's algorithm.
Step 2: Calculate the expected waiting time at each node by using M/M/1 queuing model. Check whether there exists a significant change in the expected waiting time at each node. If yes, then go to Step 1; otherwise, go to Step 3.
Step 3: Return to the shortest time route and the expected waiting time at each node.

When applying the Dijkstra's algorithm, the label value of a new node to be labeled, which is the expected travel from the source node to node v, is calculated by

$$\min_{u\in L}\left\{\begin{array}{l}\text{shortest travel time from the}\\ \text{source node to node } u(u \in L)\\ +\text{travel time from node } u\\ \text{to node } v + \text{waiting time at node } v\end{array}\right\}, \qquad (9)$$

where L is the set of labeled nodes.

The waiting time at a node q, W_q, is calculated using Eq. (10).

$$W_q = \frac{\lambda_q}{\mu_q(\mu_q - \lambda_q)} \qquad (10)$$

where λ_q and μ_q are defined as the number of arrivals per unit time and the service rate, which is the maximum number of transporters per unit time that can pass the node, respectively, at node q, when the server is busy, as explained by Gross and Harris (1985).

4. COMPARISON OF PERFORMANCES AMONG VARIOUS ALGORITHMS

4.1 Genetic Algorithm

As mentioned, a genetic algorithm is used to evaluate the performance of the algorithm. A list of bi-directed edges, which need to be set into uni-directed arcs, are expressed in $(x_1, x_2, x_3, \cdots, x_n)$, where x_i is a binary variable, which corresponds to the direction of bi-directed edge i. The value of x_i is 1, if the bi-directed edge is changed into a uni-directed arc with the direction from the node with small index to the node with large index; otherwise 0, for the opposite direction. The genetic algorithm is used to find the best combination of the arcs' directions, which minimizes the total travel and waiting times of transporters. The genetic algorithm used in this study is described in the following:

• *Initialization*
Randomly create a given number of chromosomes.

• *Selection*
In each population, some chromosomes are selected to survive in the next population and some of them are selected in order to be processed by a mutation or crossover operator to form new chromosomes. The number of chromosome candidates selected in each population is determined as follows: 10%, 50%, and 40% of the total number of chromosomes in a population using best selection, crossover, and mutation operation, respectively. For each candidate, a uniformly distributed random number between 0 and 1 is selected. If the selected random number is smaller than the crossover and mutation rates, the crossover and mutation is performed, respectively. Crossover and mutation rates are set to 0.8 and 0.05, respectively, considering that large values of crossover rate (0.5-1.0) and small values of mutation rate (0.001-0.05) are commonly employed in practice, as stated by Srinivas and Patnaik (1994).

• *Mutation*
A chromosome is randomly selected from the previous population. A random number of genes to be mu-

tated in the chromosome are selected. Then for each selected gene, a random binary value is generated. At the end of the mutation process, the chromosome's feasibility is checked. If the chromosome is infeasible, then the mutation operation is restarted. The infeasibility occurs when any node is not reachable from another node.

• *Crossover*

Two chromosomes are selected and the two-point crossover is done, by selecting random cutting points and exchanging the values of genes between those two cutting points to form two new offsprings. The feasibilities of the offsprings are checked and the crossover process is restarted if the infeasibility is detected.

The number of generations and the population size are set to 50 and 100, respectively. The algorithm is terminated after the same solution is obtained in consecutive 20 generations or the computational time limit, about 2 hours, is reached. Because the algorithm in this study needs to be run whenever the container flows change, too long computational time is not permissible.

The genetic algorithm has been coded in Java. All the tests have been carried out on an Intel® Core™ i3-2100 CPU 3.1 GHz with 2.048 GB of RAM.

4.2 Numerical Experiment

A part of the guide path network on the ground in Figure 2 was considered. Various sizes of networks are tested. In this numerical experiment, we solved 5 groups of problems, each of which consists of 4 problems, with different sizes of transporter networks and flow requirements. The number of horizontal rails was set to be between 2 or 4 and that of vertical rails was set to be one among 12, 24, 26, or 36. The number of the flow requirements was one among 30, 48, 60, and 72, as shown in Table 1.

For example, networks with 2 horizontal rails, 4 vertical rails, and 10 bi-directed path segments, are given in Figure 4(a). A flow requirement is defined as a number of containers, which must be transported from a source node to a destination node using a vehicle in the given network. A flow requirement may represent either the empty or the loaded travel(s) of vehicle(s).

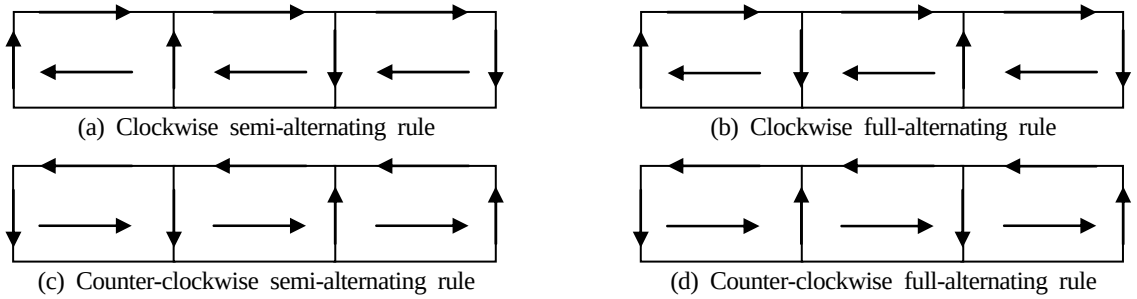
For each group of problems in Tables 1-4, four problems with these randomly generated parameters are tested. The input parameters include the position of each QC, the type of operation assigned to each QC, the source and destination nodes and flow rate (between 1-5 containers/flow) of each flow requirement.

The aim is to determine the directions of each vertical and horizontal rail in the transporter network. Five QCs are used to load/discharge containers to/from the vessel. In the experiment, it was assumed that the loaded travel time has the weight of 0.8 and the empty travel has the weight of 0.2. In the considered network of the transporter on the ground, the distance between horizontal rails is set to be 4 *m* and the distance between vertical rails is set to be 1 *m*.

The performance of the algorithm in this study is also compared with those by several simple rules, which are explained in Singgih and Kim (2015), and a genetic algorithm. Simple rules, which are formed by combining segments which form some loops in order to allow the access to all nodes, are shown by Figure 4.

In the experiment, the service rate is assumed to be constant, as follows: service rate for transfer points of QC, transfer points at storage yard, and the intersection, are 35, 40, and 50 containers per hour. The performance and the computational time comparisons between the proposed algorithm, the genetic algorithm, and the simple rules in 20 problems are shown in Tables 1-4. The proposed algorithm performs better than the simple rules in all problems, but the genetic algorithm obtains the best solutions in all problems, as long as a feasible solution is obtained. The average objective value difference between the solution obtained by the proposed algorithm and the genetic algorithm is 7.22%. The result shows that the proposed algorithm performs well. The genetic algorithm provides good results, but cannot be used in practice due to the required long computational time.

In Problems VL1-VL4, the genetic algorithm cannot obtain any feasible solution even after about 6600 seconds of the experiment. Note that the numbers of bi-directed path segments in VL1-VL4 are smaller than the ones in Problems L1-L4, which can be solved using the genetic algorithm. However, in Problems VL1, VL2, VL3, and VL4, the network consists of 4 horizontal rails and has more intersections, which connect 4 path segments,



(a) Clockwise semi-alternating rule (b) Clockwise full-alternating rule

(c) Counter-clockwise semi-alternating rule (d) Counter-clockwise full-alternating rule

Figure 4. Simple rules (Singgih and Kim, 2015).

Table 1. Performance comparisons among various algorithms using networks with 2 horizontal lanes

Problem	Problem size*	Total weighted loaded and empty travel times						
				Simple rules (Singgih and Kim, 2015)				Smallest among (B~F)-A (%)
		Proposed algorithm (A)	Genetic algorithm (B)	Clockwise semi-alternating rule (C)	Clockwise full-alternating rule (D)	Counter-clockwise semi-alternating rule (E)	Counter-clockwise full-alternating rule (F)	
VS1	12/34/30	110.756	**99.608**	177.766	161.291	192.195	151.480	-11.19
VS2		178.017	**163.489**	252.182	236.091	300.434	236.103	-8.89
VS3		180.110	**163.431**	277.689	240.725	234.509	231.192	-10.21
VS4		145.077	**131.307**	200.389	174.701	214.845	184.351	-10.49
S1	24/70/48	385.339	**364.505**	522.000	456.177	464.236	441.674	-5.72
S2		531.648	**500.711**	652.218	605.547	599.526	581.523	-6.18
S3		483.458	**448.741**	597.034	542.396	606.539	532.661	-7.74
S4		297.373	**280.414**	439.289	389.452	434.538	379.900	-6.05
M1	26/76/60	576.475	**538.829**	703.370	645.424	664.724	649.995	-6.99
M2		654.050	**611.078**	784.078	723.162	769.369	718.032	-7.03
M3		809.482	**763.585**	933.760	884.940	976.360	879.449	-6.01
M4		1,062.432	**1,000.203**	1,117.041	1,095.230	1,141.773	1,105.187	-6.22
L1	36/106/72	994.178	**941.431**	1,154.214	1,070.041	1,086.336	1,064.879	-5.60
L2		949.986	**905.545**	1,150.210	1,045.996	1,154.539	1,026.194	-4.91
L3		927.583	**868.350**	1,095.040	1,001.343	998.581	986.871	-6.82
L4		980.792	**929.652**	1,094.036	1,085.670	1,122.848	1,090.461	-5.50

* Number of horizontal rails/bi-directed path segments/flow requirements (including the empty travels).

Table 2. Performance comparisons among various algorithms to solve VL1-VL4 problems with 4 horizontal lanes, 12 vertical lanes, 80 bi-directed path segments, and 30 flow requirements

Problem	Total weighted loaded and empty travel times						
			Simple rules (Singgih and Kim, 2015)				Smallest among (B~F)-A (%)
	Proposed algorithm (A)	Genetic algorithm (B)	Clockwise semi-alternating rule (C)	Clockwise full-alternating rule (D)	Counter-clockwise semi-alternating rule (E)	Counter-clockwise full-alternating rule (F)	
1	129.944	No feasible solution**	215.122	190.746	229.611	**181.097**	28.25
2	196.749	No feasible solution	281.699	265.620	329.797	**265.615**	25.93
3	201.474	No feasible solution	307.329	270.453	264.050	**260.882**	22.77
4	169.404	No feasible solution	230.262	**204.591**	244.712	214.232	17.20

** No feasible solution could be found before the computational time of 6600 seconds.

while in other groups of problems (Problems VS1-L4), the network consists of only 2 horizontal rails, which does not have any 4-sided intersections. In networks with more 4-sided intersections, there is higher probability to create infeasible solutions when the genetic algorithm is used. A solution is infeasible when no ingoing or outgoing path segment exists in an intersection.

Solving the problem using genetic algorithm requires too long computational time, which is not practical. It was found that the proposed algorithm can solve the problem in a shorter time with a good quality, compared with genetic algorithm.

5. APPLICATION OF THE PROPOSED ALGORITHM TO THE NEW CONCEPTUAL RAIL-MOUNTED TRANSPORT SYSTEM

In the numerical experiment, the layout in Figure 2 was

Table 3. Computational time comparisons among various algorithms to solve flow path design problem using network with 2 horizontal lanes (in seconds)

Problem	Proposed algorithm	Genetic algorithm	Simple rules (Singgih and Kim, 2015)			
			Clockwise semi-alternating rule	Clockwise full-alternating rule	Counter-clockwise semi-alternating rule	Counter-clockwise full-alternating rule
VS1	1	11	< 1	< 1	< 1	< 1
VS2	1	12	< 1	< 1	< 1	< 1
VS3	1	12	< 1	< 1	< 1	< 1
VS4	1	10	< 1	< 1	< 1	< 1
S1	1	253	< 1	< 1	< 1	< 1
S2	1	252	< 1	< 1	< 1	< 1
S3	1	206	< 1	< 1	< 1	< 1
S4	1	260	< 1	< 1	< 1	< 1
M1	42	314	< 1	< 1	< 1	< 1
M2	49	345	< 1	< 1	< 1	< 1
M3	53	302	< 1	< 1	< 1	< 1
M4	58	374	< 1	< 1	< 1	< 1
L1	181	4,101	< 1	< 1	< 1	< 1
L2	154	2,740	< 1	< 1	< 1	< 1
L3	169	4,138	< 1	< 1	< 1	< 1
L4	189	3,600	< 1	< 1	< 1	< 1

Table 4. Computational time comparisons for VL1-VL4 problems with 4 horizontal lanes, 12 vertical lanes, 80 bi-directed path segments, and 30 flow requirements (in seconds)

Problem	Proposed algorithm	Genetic algorithm	Simple rules (Singgih and Kim, 2015)			
			Clockwise semi-alternating rule	Clockwise full-alternating rule	Counter-clockwise semi-alternating rule	Counter-clockwise full-alternating rule
1	32	> 6,600	< 1	< 1	< 1	< 1
2	35	> 6,600	< 1	< 1	< 1	< 1
3	90	> 6,600	< 1	< 1	< 1	< 1
4	31	> 6,600	< 1	< 1	< 1	< 1

used. The speed of transporters on the ground was assumed to be 3 m per sec. The directions of path segments in the same vertical rail are set to be the same. The reason is to maintain smooth flows of containers and thus shorter container transportation times by the transporters through the vertical rails on which, otherwise, a higher congestion is expected. A real-sized automated container terminal is considered, which is serving a vessel using 5 QCs. The transportation of containers between the quay side and the transfer points, close to the storage yard, are performed by using transporters.

The rail network for transporters moving on the ground consists of 9 vertical rails, 6 horizontal rails, which pass the transfer point of QCs, and other 8 horizontal rails, close to the storage yard. The rail network for transporters moving above the yard consists of 24 vertical rails. In the transporter on the ground's network, 9 vertical rails and 6 horizontal rails exist, while the directions of 121 path segments must be determined. Each path segment consists of several partitions, which must have the same direction and connect QC TPs, transporter above the yard's TPs and vertical rails in the transporter on the ground's network. The distance between two adjacent horizontal rails is 2.5 *m*, while the distance between two adjacent vertical rails is 6.3 or 12.5 *m*, in the network of transporters on the ground.

The control system for transporters changes the directions of path segments whenever the flow requirements of the containers change. The flow requirements can be changed when the position of any QC is changed, or any QC completes its current operation (for example, discharging operation) at the current bay and starts a new operation (for example, loading operation) at the same bay. Because the decision must be done in real-time, the time required to determine the directions must be very short.

5.1 Estimating the Expected Service Rate Based on the Proportions of Various Types of Operations at Each Node

The arrival rate at a node is calculated by adding the

arrival rates of all flow requirements which have the shortest time paths through the corresponding node. The method to estimate the expected waiting time in each node depends on the type of node and its own service rate. This study estimates the service rate based on the proportions of various types of operations performed at each individual node, which is later calculated using Eqs. (11)-(16).

The input parameters used as the service times at various types nodes are shown by Table 5, while formulas for calculating the required times for performing various operations in each type of nodes are provided in Table 6. In Table 6, the service time at the transfer points between a transporter on the ground and a transporter above the yard is estimated based on the required time for the hoist of transporter above the yard to move vertically and release or pick up a container.

The required time to pass an intersection needs to be calculated based on various operations performed the transporter on the intersection. The transport system under consideration has 5 types of nodes. Each type of nodes has its unique set of operations as shown in Table 7.

Service rate of each node is calculated using a general equation, as shown by Eq. (11):

Given the service time for each type of operation in Table 7, the service rates of all types of nodes are calculated by using Eqs. (12)-(16).

Table 7. Five types of nodes and operations performed at each type of nodes

Type of nodes	Performed operations				
	QC service	Transfer a container between transporters	Horizontal movement	Vertical movement	Turning
1				V	V
2		V	V		
3	V		V		
4			V	V	V
5		V	V		

$$\mu_1 = \frac{3,600}{(t_v n_v + t_t n_t)/(n_v + n_t)} \tag{12}$$

$$\mu_2 = \frac{3,600}{(t_c n_c + t_h n_h)/(n_c + n_h)} \tag{13}$$

$$\mu_3 = \frac{3,600}{(t_q n_q + t_h n_h)/(n_q + n_h)} \tag{14}$$

$$\mu_4 = \frac{3,600}{(t_h n_h + t_v n_v + t_t n_t)/(n_h + n_v + n_t)} \tag{15}$$

$$\mu_5 = \frac{3600}{(t_c n_c + t_h n_h)/(n_c + n_h)} \tag{16}$$

$$\mu = \frac{3600}{\left[\frac{\sum_{i \in T}\{(\textit{service time of type i operation}(\text{sec})) \times (\textit{number of type i operations performed in an hour})\}}{\text{total number of operations performed in an hour}}\right]} \tag{11}$$

Table 5. Travel time of transporter on ground between adjacent nodes

Type of travel	Explanation on the travel	Travel time (seconds)
Horizontal movement	Travel by a transporter on the ground to move from a node to the next adjacent node horizontally	3 or 5
Vertical movement	Travel by a transporter on the ground to move from a node to the next adjacent node vertically	1
Turning time	Travel by a transporter on the ground to change direction from horizontal to vertical movement, or vice versa	30

Table 6. Service times for types of operations at each node (for service rate calculation)

Type of operation	Explanation of operation	Service time (seconds)
QC's service time (t_q)	A QC transfers a container to a transporter for loading/discharging operations	103
Transporter's transfer time (t_c)	A transporter transfers a container to another transporter	45
Transporter's occupancy time during the horizontal movement (t_h)	Occupancy time of the transporter for the horizontal movement during which no other vehicle is allowed to enter the node	5, 7, and 9 (short, medium, and long distances)
Transporter's occupancy time during the vertical movement (t_v)	Occupancy time of the transporter for the vertical movement during which no other vehicle is allowed to enter the node	2
Transporter's occupancy time during the turning movement (t_t)	Occupancy time of the transporter for the turning movement during which no other vehicle is allowed to enter the node	30

where:
n_q = number of QC operations performed at the node
n_c = number of container transfer operations between transporters on the ground and above the yard
n_h = number of transporters passed the node in the horizontal direction
n_v = number of transporters passed the node in the vertical direction
n_t = number of transporters which turn (change their directions from a horizontal to a vertical movement, or vice versa) at the node

In the network of transporters on the ground, the values of t_q, t_c, t_h, t_v, and t_t are obtained from Table 6.

5.2 Determining the Directions of Path Segments

The problems are characterized by the operation performed by the QCs, the combination of source and designated nodes, and the amount of containers to deliver per unit time. Flow requirements of QCs 1 and 2 in Problem 1 are shown in Tables 8 and 9. QCs are currently performing loading or discharging operations. For discharging operation, designated vertical rails of transporter above the yard, which are located close with the target QC's position, are selected, while for loading operations, source vertical rails are already predetermined in the planning stage.

Using Djikstra's algorithm, the problem is solved. Table 10 shows the total weighted loaded and empty travel times, and the total waiting times, while Table 11 summarizes the required computational times. In this experiment, the threshold value (θ), which determines the number of iterations required during the search of shortest time path and calculation of total waiting times, is set to be 0.0005. The total waiting time denotes the

Table 8. Flow requirements of quay crane 1 (discharging)

From [QC, TP No] \ To [Horizontal, Vertical Rail No]	[1, 1]	[6, 1]	[8, 1]	[1, 3]	[3, 3]	[2, 4]	[7, 4]	[9, 4]	[4, 5]	[8, 6]	[12, 8]
[1, 1]	4	1									
[1, 2]			4	2							
[1, 3]				2	1	3					
[1, 4]						1	4	1			
[1, 5]									4	2	
[1, 6]										2	4

Table 9. Flow requirements of quay crane 2 (loading)

From [Horizontal, Vertical Rail No] \ To [QC, TP No]	[2, 1]	[2, 2]	[2, 3]	[2, 4]	[2, 5]	[2, 6]
[5, 3]	4					
[4, 6]	2	2				
[9, 6]		3				
[3, 12]			4			
[13, 12]			2	2		
[12, 17]				4		
[13, 17]					4	
[2, 21]					2	2
[4, 21]						4

Table 10. Performance of the proposed algorithm for various numbers of simultaneous determinations of directions of multiple path segments (n)

Problem	Total weighted travel times (sec)				Total waiting times			
	n = 1	n = 10	n = 50	n = 100	n = 1	n = 10	n = 50	n = 100
1	1,260	12,655	14,113	14,388	9.53	9.71	10.56	10.66
2	11,232	11,827	13,124	13,219	7.95	8.31	9.72	9.84
3	10,175	11,347	12,249	12,441	6.20	7.08	7.30	7.31
4	11,697	12,470	12,768	13,221	7.83	7.79	8.30	8.26

Table 11. Computational time for various numbers of simultaneous determinations of directions of multiple path segments (n)

Problem	n = 1	n = 10	n = 50	n = 100
1	2,612	476	226	150
2	2,978	441	274	337
3	2,453	400	209	173
4	3,153	495	264	219

summation of the waiting times at all nodes in the transporter network.

As shown by Table 11, as the value of n, which is the number of path segments whose directions are determined simultaneously (before searching for next shortest time paths), decreases, the total weighted travel and waiting times decreases. The reason is that as n decreases, the waiting times are updated more often and as a result, better directions of path segments could be obtained. However, with small values of n, longer computational time is required. Determining directions of more path segments simultaneously results in worse total weighted travel and waiting times, but shorter computational times, which can support real-time decisions.

The threshold value (θ) determines the number of iterations required during the search of shortest time path. Table 12 shows the effect of the threshold value to the obtained solution. Using a smaller θ was expected to produce shorter total weighted loaded and empty travel times, because the shortest time path search and waiting time calculations are performed in more iterations, even more computational times are required. However, the experiment result shows that the θ value does not consistently affect the total weighted travel time and the computational time. The final solution, obtained by using the proposed algorithm, is shown in Figure 5. The values of the search parameters are set θ = 0.0005 and n = 50.

6. CONCLUSIONS

In this study, the problem of determining the directions of path segments in a service network design, considering the usage of automated transporters mounted on rails, was addressed. The physical network was given and some bi-directed path segments in the physical network are decided to be uni-directional. An algorithm, which was combined with a modified Dijkstra's algorithm to find the shortest time path and queuing theory to calculate the waiting times at nodes, was proposed, and results from the numerical experiment was provided. Based on the experiment, the proposed algorithm was able to solve the problem in a short time, with a good performance, compared with a genetic algorithm and

Table 12. Performance of the proposed algorithm for various values of threshold (θ) (n = 50)

Problem	Total weighted travel times			Total waiting time			Computational time		
	θ = 0.00005	θ = 0.0005	θ = 0.05	θ = 0.00005	θ = 0.0005	θ = 0.05	θ = 0.00005	θ = 0.0005	θ = 0.05
1	14,112.98	14,112.98	14,255.81	10.56	10.56	10.83	201	226	187
2	13,154.46	13,123.64	13,123.64	9.92	9.72	9.71	284	274	276
3	12,248.66	12,248.66	12,063.07	7.30	7.30	7.45	202	209	240
4	12,767.53	12,767.53	12,767.53	8.30	8.30	8.30	271	264	221

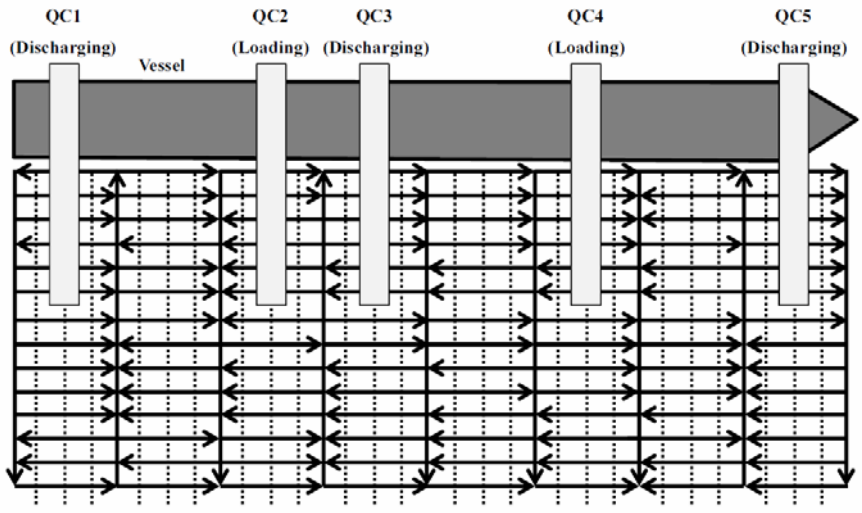


Figure 5. Directions of segments in the terminal as result of the proposed algorithm.

some simple rules. Studies, which combine the proposed flow path design with routing algorithms in a simulation model, must be conducted in order to evaluate the performance of the proposed algorithm, while being integrated with other algorithms.

ACKNOWLEDGMENTS

This work was supported by the Technological Development of Low-carbon Automated Container Terminals Project funded by the Ministry of Oceans and Fisheries, Korea (Project Number: 201309550003).

REFERENCES

Al-Sultan, K. S. and Bozer, Y. A. (1998), Network configuration and machine layout in fixed-path material handling systems, *Annals of Operations Research*, **76**, 21-54.

Drezner, Z. and Wesolowsky, G. O. (2003), Network design: selection and design of links and facility location, *Transportation Research Part A*, **37**(3), 241-256.

Fu, L., Sun, D., and Rilett, L. R. (2006), Heuristic shortest path algorithms for transportation applications: state of the art, *Computers and Operations Research*, **33**(11), 3324-3343.

Gross, D. and Harris, C. (1985), *Fundamentals of Queuing Theory: Second Edition*, John Wiley and Sons, New York.

Gallo, M., D'Acierno, L., and Montella, B. (2010), A meta-heuristic approach for solving the urban network design problem, *European Journal of Operational Research*, **201**(1), 144-157.

Guan, X., Dai, X., and Li, J. (2011), Revised electromagnetism-like mechanism for flow path design of unidirectional AGV systems, *International Journal of Production Research*, **49**(2), 401-429.

Herrmann, J. W., Ioannou, G., and Minis., I. (1995), Design of material flow networks in manufacturing facilities, *Journal of Manufacturing Systems*, **14**(4), 277-289.

Jeon, S. M., Kim, K. H., and Kopfer, H. (2011), Routing automated guided transporters in container terminals through the Q-learning technique, *Logistics Research*, **3**(1), 19-27.

Kaspi, M. and Tanchoco, J. M. A. (1990), Optimal flow path design of unidirectional AGV systems, *International Journal of Production Research*, **28**(6), 1023-1030.

Kaspi, M., Kesselman, U., and Tanchoco, J. M. A. (2002), Optimal solution for the flow path design problem of a balanced unidirectional AGV system, *International Journal of Production Research*, **40**(2), 389-401.

Kim, K. H., Phan, M.-H. T., and Woo, Y. J. (2012), New conceptual handling systems in container terminals, *Industrial Engineering and Management Systems*, **11**(4), 299-309.

Lim, J. K., Lim, J. M., Yoshimoto, K., Kim, K. H., and Takahashi, T. (2002a), A construction algorithm for designing guide paths of automated guided vehicle systems, *International Journal of Production Research*, **40**(15), 3981-3994.

Lim, J. K., Lim, J. M., Yoshimoto, K., Kim, K. H., and Takahashi, T. (2002b), Designing guide-path networks for automated guided vehicle system by using the Q-learning technique, *Computers and Industrial Engineering*, **44**(1), 1-17.

Gaskins, R. J. and Tanchoco, J. M. A. (1987), Flow path design for automated guided vehicle systems, *International Journal of Production Research*, **25**(5), 667-676.

Miandoabchi, E. and Farahani, R. Z. (2011), Optimizing reserve capacity of urban road networks in a discrete network design problem, *Advances in Engineering Software*, **42**(12), 1041-1050.

Miandoabchi, E., Farahani, R. Z., Dullaert, W., and Szeto, W. Y. (2012a), Hybrid evolutionary metaheuristics for concurrent multi-objective design of urban road and public transit networks, *Networks and Spatial Economics*, **12**(3), 441-480.

Miandoabchi, E., Farahani, R. Z., and Szeto, W. Y. (2012b), Bi-objective bimodal urban road network design using hybrid metaheuristics, *Central European Journal of Operations Research*, **20**(4), 583-621.

Seo, Y. and Egbelu, P. J. (1999), Integrated manufacturing planning for an AGV-based FMS, *International Journal of Production Economics*, **60**/**61**, 473-478.

Shen, Y. C. and Lau, L. K. (1997), Planning of flow path to minimize expected waiting time for free-ranging automated guided vehicle systems, *IEEE International Conference on Systems, Man, and Cybernetics*, **4**, 3738-3743.

Singgih, I. K. and Kim, K. H. (2015), Service network design for rail-mounted transporters, *ICIC Express Letter*, **9**(4), 1025-1032.

Sinriech, D. and Tanchoco, J. M. A. (1991), Intersection graph method for AGV flow path design, *International Journal of Production Economics*, **29**(9), 1725-1732.

Srinivas, M. and Patnaik, L. M. (1994), Adaptive probabilities of crossover and mutation in genetic algorithms, *IEEE Transactions on Systems, Man and Cybernetics*, **24**(4), 656-667.

Vosniakos, G. C. and Davies, B. J. (1989), On the path layout and operation of an AGV system serving and FMS, *International Journal of Advanced Manufacturing Technology*, **4**, 243-262.

Zhang, M., Batta, R., and Nagi, R. (2009), Modeling of workflow congestion and optimization of flow routing in a manufacturing/warehouse facility, *Management Science*, **55**(2), 267-280.

Zhang, M., Batta, R., and Nagi, R. (2011), Designing manufacturing facility layouts to mitigate congestion, *IIE Transactions*, **43**, 689-702.