OR Spectrum (2003) 25: 25–44

OR Spectrum

A dispatching method for automated guided vehicles by using a bidding concept*

Jae Kook Lim[1], **Kap Hwan Kim**[2], **Kazuho Yoshimoto**[3], **Jun Ho Lee**[2], **and Teruo Takahashi**[1]

[1] Institute of Asia-Pacific Studies, Waseda University, Sodai-Nishiwaseda bldg. 6F, 1-21-1 Nishiwaseda Shinjuku-ku Tokyo, 169-0051, Japan
(e-mail: {jklim; taka12}@mn.waseda.ac.jp)
[2] Department of Industrial Engineering, Pusan National University, Jangjeon-dong, Kumjeong-ku, Pusan 609-735, Korea
(e-mail: kapkim@pusan.ac.kr; chiwoo@sktelecom.com)
[3] Department of Industrial & Management Systems Engineering, Waseda University, 3-4-1 Okubo Shinjuku-ku Tokyo 169-8555 Japan
(e-mail: kazuho@yoshi.mgmt.waseda.ac.jp)

Abstract. A dispatching method is suggested for automated guided vehicles by using an auction algorithm. The dispatching method in this study is different from traditional dispatching rules in that it looks into the future for an efficient assignment of delivery tasks to vehicles and also in that multiple tasks are matched with multiple vehicles. The dispatching method in this study is distributed in the sense that the dispatching decisions are made through communication among related vehicles and machines. The theoretical rationale behind the distributed dispatching method is also discussed. Through a simulation study, the performance of the method is compared with that of a popular dispatching rule.

Key words: Automated guided vehicle – Distributed dispatching method – Simulation

1 Introduction

The dispatch of automated guided vehicles (AGVs) can be defined as the assignment of vehicles to a delivery task so that some performance objectives of a shop are achieved. The most popular strategy for dispatching vehicles is to match a delivery

* The research was financially supported by the Sasakawa Scientific Research Grant from The Japan Science Society. The original version of the simulation program is provided by Professor Jae Yeon Kim at Dong Yang University, Korea.
Correspondence to: J. K. Lim

request with an idle vehicle whenever a delivery request is issued or whenever a vehicle becomes idle. When the dispatching decision is triggered by an occurrence of a delivery task, one of vehicles idle at that time is selected for the new delivery task. Also, when a vehicle becomes idle, one of the waiting delivery-tasks is chosen for the new idle vehicle. Traditional dispatching rules for the former case and the latter case are called "task-initiated dispatching rules" and "vehicle-initiated dispatching rules," respectively (Egbelu and Tanchoco, 1984).

Although the traditional dispatching rules are simple to use, one drawback of these rules is that they are myopic in a sense that only one vehicle is considered in case of the vehicle-initiated rules and only one delivery task is considered in case of the task-initiated rules. For example, suppose that a new idle vehicle was assigned to a delivery task at a workstation (let it be workstation A) because workstation A was located nearest to the vehicle among all the workstations with delivery tasks. However, suppose that, just after the (first) vehicle was assigned, another (second) vehicle became idle at a location nearer to workstation A than the first vehicle did. In this case, if the dispatching decision had been made considering where and when the second vehicle would have become idle, the second vehicle must have been assigned to workstation A instead of the first vehicle.

To overcome this problem, when a vehicle becomes idle and needs a delivery task to be assigned, all the vehicles must be considered simultaneously as well as all the delivery tasks. That is, the dispatching decision must be made in many-to-many basis instead of one-to-many basis. However, the optimal decision-making for all the vehicles is computationally impractical especially when the number of vehicles involved is large. Also, dispatching decisions must be made by a central processor that has complete information on states of all the vehicles and workstations. However, when the size of a material handling system is large, it is risky to depend on only a central controller because of possible breakdowns or overloading of the central controller. Also, it is not economical for the central controller to make the dispatching decision again for all the vehicles whenever a small change in the system state occurs.

Thus, the dispatching algorithm in this study attempts to satisfy the following desirable conditions:

(1) A high level of system performance must be obtained. For the high performance, the dispatching decision must be near optimal form the perspective of the empty travel distance of vehicles or the response time of vehicles to calls from workstations.
(2) The decision process for dispatching must be distributed so that the central controller is not overloaded and the entire material handling system becomes robust to various failure and breakdowns of some components and the central controller.
(3) The effect of small changes in the system's state must be localized. For example, when a new delivery task arrives, the changes in the dispatching decision are usually confined to several related vehicles. Thus, it is necessary to develop a method for identifying the related vehicles and revising the dispatching decisions only for the related vehicles without having to solve the entire assignment problem again.

Bartholdi and Platzman (1989) proposed the first-encountered-first-served (FEFS) rule, which, with a closed single-loop guide path, attempts to deliver tasks as quickly as possible for AGVs. Egbelu (1987) suggested the demand-driven rule in which AGVs are first dispatched to delivery tasks that are bound for input buffers whose lengths are below a threshold value. Kim et al. (1999) suggested an AGV dispatching method in which balancing workloads among different workstations is the first criterion for selecting the next delivery task. For selecting a vehicle, although the dispatching method proposed by Kim et al. (1999) looks beyond the times when vehicles become idle, their method basically selects in sequence one task among several tasks and then one vehicle among many vehicles sequentially. Thus, their method does not attempt to optimally match multiple tasks with multiple vehicles simultaneously. Sabuncuoglu, I. and Hommertzheim (1992) proposed a dynamic scheduling method for both vehicles and machines.

Klein and Kim (1996) compared multi-attribute dispatching rules with single-attribute dispatching rules and showed that the former rules are superior to the latter rules. Lee et al. (1996) and Bilge and Tanchoco (1997) treated the dispatching problem for AGVs with multi-load capacities. Taghaboni-Dutta (1997) suggested a dispatching rule based on an index of values added during the operations of a job in a shop.

Bilge and Ulusoy (1995) provided a simultaneous scheduling method for operation of machines and transfer of materials by AGVs. The scheduling problem was decomposed into two subproblems: a machine scheduling subproblem and a vehicle scheduling subproblem. An iterative procedure was suggested for each subproblem. Their paper is related to this study in that future delivery tasks and idle vehicles are considered for the vehicle scheduling. However, the problem solved in their study was limited in size because of the complexity of the suggested algorithm. Co and Tanchoco (1991) provided a good review about the various aspects of the operation of AGVs.

This study suggests a dispatching method based on a bidding concept. The bidding-based dispatching method (BDM) assumes a market system in which vehicles attempt to earn money as much as possible by performing delivery tasks with the highest possible price at the lowest possible costs, and delivery tasks attempt to pay charges as less as possible by hiring vehicle with the lowest possible opportunity cost.

It is shown in property 2 that BDM results in the optimal solution of an assignment problem in which cost coefficients correspond to empty travel times or response times. BDM considers currently busy vehicles as well as currently idle vehicles by looking ahead to the change of states of vehicles. Thus, BDM makes dispatching decisions on a many-to-many basis instead of a one-to-many basis, as was in the case of workstation-initiated rules or vehicle-initiated rules in previous studies. This property of BDM coincides with the first desirable condition of the dispatching algorithm.

However, when a decision is made about the assignment of multiple vehicles to multiple delivery tasks, it is time-consuming to reconsider all the dispatching decisions already made whenever a new delivery task arrives at the shop or whenever a vehicle becomes idle. In BDM, a newly idle vehicle submits a bid to the task giving the maximum margin to the vehicle, and then the task sends a cancellation notice to a previously assigned vehicle (let it be the second vehicle) so that the second vehicle looks for another task, and so on. When a new delivery task arrives at the shop, similar things happen. Thus, the dispatching procedure of BDM is distributed in that no central controller is necessary. Also, the effect of small changes in the system's state is localized because the changes in the dispatching decision are confined to several tasks and vehicles. Thus, second and the third desirable conditions of the dispatching algorithm are satisfied.

In the next section, an auction-based assignment algorithm and the concept of economic equilibrium are introduced. Section 3 suggests a process of bidding-based dispatching and a method of constructing bids. Section 4 describes results of a simulation study to evaluate the performance of the dispatching procedure described in this paper. Conclusions are provided in the final section.

2 An assignment problem and the economic equilibrium

In order to explain the rationale of the bidding-based dispatching method, the concept of economic equilibrium is first introduced. In this section, the dispatching problem is considered as an assignment problem in which multiple delivery tasks are matched with multiple vehicles and objective function is minimizing the total travel distance or the total response time of vehicles. Assuming that there are n vehicles and m delivery tasks to be matched with each other at a specific point in time, it is attempted to solve the assignment problem through a market mechanism (Bertsekas, 1990), viewing each task or a vehicle as an economic agent acting for its own best interest.

Let a_{ij} be the cost of vehicle i to perform task j. The cost may include only that for empty travel or for both empty and loaded travel. And, let x_{ij}= 1 if vehicle i is assigned to task j; 0, otherwise. Note that the number of vehicles and the number of delivery tasks are not usually the same. When the number of vehicles is larger than the number of tasks, dummy tasks will be added to the list of vehicles. On the other hand, when the number of tasks is larger than the number of vehicles, dummy vehicles will be added. The cost parameters of each dummy vehicle (task), a_{ij}, are set to be the same for all tasks (vehicles). In the following discussion, without the loss of generality, it is assumed that the number delivery tasks waiting for the assignment of a vehicle is larger than or equal to the number of assignable vehicles. Thus, it is assumed that dummy vehicles may be added for the formulation of the assignment problem.

Then, the minimum cost assignment problem can be formulated as follows:

(P1)

$$\text{Minimize} \sum_{i=1}^{m} \sum_{j=1}^{m} a_{ij}\, x_{ij}$$

subject to

$$\sum_{j=1}^{m} x_{ij} = 1, \text{ for all } i,$$

$$\sum_{i=1}^{m} x_{ij} = 1, \text{ for all } j,$$

$$x_{ij} \geq 0, \text{for all } i \text{ and } j.$$

The dual of (P1) becomes
(D)

$$\text{Minimize} \sum_{i=1}^{m} v_i - \sum_{j=1}^{m} p_j$$

subject to

$$v_i - p_j \quad \geq -a_{ij} \text{ for all } i \text{ and } j. \tag{1}$$

The concept of the complementary slackness is useful to introduce the relationship between the optimality of an assignment and the condition that values of prices and margins must satisfy as follows:

(**An optimality condition**) A solution x_{ij} and (v_i, p_j) is optimal if x_{ij} and (v_i, p_j) are feasible for (P1) and (D), respectively, and satisfy the complementary slackness conditions of linear programming, which can be stated as follows (Hillier and Lieberman, 1986):

$$(v_i - p_j + a_{ij})x_{ij} = 0 \text{ for all } i \text{ and } j.$$

The scalar p_j will be referred to as the price of delivery task j, which task j must pay to the corresponding vehicle performing the delivery task. Also, v_i can be interpreted as the profit margin of vehicle i that performed task j.

In the following, a concept of "equilibrium," – which is useful in devising a bidding-based dispatching method – is introduced:

(**Definition of the equilibrium**) Suppose that task j is assigned to vehicle i. Vehicle i will be *happy* if $v_i = p_j - a_{ij} = \max_{k=1,\ldots,m} \{p_k - a_{ik}\}$. We will say that a feasible assignment and a set of prices and margins are at *equilibrium* when all the vehicles are *happy*.

The following property guarantees the optimality of prices and margins that are at equilibrium:

Property 1: (The optimality of the equilibrium condition) For a feasible assignment, if the a set of prices and margins are at equilibrium, the assignment is optimal.

Proof. The fact that $v_i = p_j - a_{ij} = \max_{k=1,\ldots,m} \{p_k - a_{ik}\}$ for all i implies that $v_i \quad \geq p_k - a_{ik}$ for all i and k, which in turn means that the set of v_i and p_k is feasible for constraint (1). Also, the fact that $v_i = p_j - a_{ij}$ for all i and j with positive x_{ij} (based on the definition of the equilibrium) implies that the complementary slackness conditions are satisfied. Thus, the conclusion holds. □

For devising a bidding procedure which guarantees to be stopped within a finite number iterations, the concepts of "almost equilibrium" and "almost happy" will be introduced which are almost equivalent to those of "equilibrium" and "happy" in the following (Bertsekas, 1990):

(**Definition of almost equilibrium**) Suppose that task j is assigned to vehicle i. Vehicle i will be *almost happy* if $v_i = p_j - a_{ij} = \max_{k=1,\dots,m} \{p_k - a_{ik}\} - \varepsilon$ for a small positive real number ε. Task j will be *almost happy* if $p_j = v_i + a_{ij} = \min_{k=1,\dots,m} \{v_k + a_{kj}\} + \varepsilon$ for a small positive real number ε . We will say that a feasible assignment and a set of prices and margins are at *almost equilibrium* when all vehicles or all tasks are *almost happy*.

The following section describes a dispatching algorithm for obtaining an assignment of candidate trucks to candidate tasks that satisfies the conditions of the almost equilibrium.

3 A bidding-based procedure for dispatching vehicles

BDM assumes that the price of each delivery is determined through a bidding process. During the bidding process, each vehicle selects the delivery task that maximizes its own margin which is the price of the task minus the cost of performing the task, while each task chooses the vehicle that requests the least compensation which is the transportation cost required by the vehicle for performing the task plus the minimum margin requested by the vehicle.

For a given assignment of vehicles to delivery tasks, a set of prices and margins is said to be at "equilibrium" if a vehicle cannot increase its margin by changing its currently assigned task and a delivery task cannot decrease the compensation by changing its currently assigned vehicle (see Sect. 2 for more formal definition of the equilibrium).

In the dynamic situation assumed in this study, all loaded or idle vehicles are candidates for dispatching. Note that empty vehicles traveling to pick up a load is excluded from the set of candidates. Tasks in the output buffer space at each workstation are considered to be candidates for dispatching. A dispatching decision process is initiated whenever a vehicle is loaded or a new delivery order is issued. When a vehicle is loaded, the ASSIGN-TASK-TO-A-NEW-VEHICLE procedure is initiated to secure the next delivery task. In case a delivery order is issued, the ASSIGN-VEHICLE-TO-A-NEW-TASK procedure is initiated. Once either of the two procedures is initiated, a feasible assignment and prices almost at equilibrium – which is defined in Section 2 – are obtained. A dispatching decision on a vehicle or a delivery task is fixed and implemented in either of two following cases: when a vehicle becomes idle and it has an assigned delivery task or when a task is assigned an idle vehicle. Thus, once a vehicle starts an empty travel for implementing a delivery task, both the vehicle and the task will be excluded from the candidate list for dispatching.

During the ASSIGN-TASK-TO-A-NEW-VEHICLE procedure, prices of tasks and margins of vehicles decrease. However, during the ASSIGN-VEHICLE-TO-A-NEW-TASK procedure, prices of tasks and margins of vehicles increase. The prices of tasks are limited by a pre-specified upper bound of the price from the above, while, above zero, margins of vehicles change.

In the following, two procedures (ASSIGN-TASK-TO-A-NEW-VEHICLE, ASSIGN-VEHICLE-TO-A-NEW-TASK) will be introduced for optimally matching vehicles with delivery tasks. The optimality of the resulting assignment will be proved In Property 2. Note that the number of available vehicles (n) may not be the same as the number of available delivery tasks (m). Assume that $m > n$ without the loss of generality. Then, even after an assignment is determined, one or more tasks may not be assigned ("unassigned and inactive": UI) to a vehicle. Both the number of vehicles and the number of tasks in the "assigned (A)" state are n. However, when a vehicle (task) becomes a new candidate for an assignment, it is "unassigned" but has a potential to be assigned. The vehicle (task) is said to be "unassigned but activated (UA)." Also, during the assignment procedure, a less competitive vehicle (task) may have its assigned task (vehicle) taken away by another more competitive vehicle (task). Then, the former vehicle (task) becomes "unassigned but activated (UA)," while the latter vehicle (task) becomes "assigned (A)."

The bidding process (ASSIGN-TASK-TO-A-NEW-VEHICLE) for the case of the vehicle initiation (when a new vehicle becomes idle) is illustrated in Figure 1. Before the new vehicle becomes idle, three vehicles are assigned to three tasks. Thus, they are in state "A" except one task that is not assigned to any vehicle and so is in state "UI" (see Fig. 1a). When a vehicle (vehicle D) becomes idle, its initial state is "UA", and it seeks a delivery task to perform (see Fig. 1b). Among waiting tasks, vehicle D selects a task (task 3) giving the maximum margin at the current price. Then, vehicle D submits a bid with a price lower than the current price of task 3. Then, task 3 accepts the bid because the suggested price is lower than the current price, and informs the cancellation of assignment to the vehicle (vehicle C) that task 3 was previously assigned to. The state of vehicle D is changed from "UA" to "A" and the state of vehicle C is changed from "A" to "UA" (see Fig. 1c). Then, vehicle C whose state became "UA" searches for the task that gives the highest margin at the current price. In Figures 1c, it is task 4. Because task 4 was in state of "UI," no vehicle turns to "UA" and so the bidding process is terminated (see Fig. 1d). Note that tasks 1 and 2 and vehicles A and B were not involved in the entire dispatching process. That is, the effect of changes was confined only to related tasks and vehicles. And also note that no central controller was involved in the bidding process.

A similar process (ASSIGN-VEHICLE-TO-A-NEW-TASK) will be followed when a new delivery task appears. Figure 2 illustrates the bidding process. The new task selects the vehicle (vehicle 1) with the lowest price that is the sum of the travel cost for vehicle to perform the new task and the current margin of vehicle 1 for performing the currently assigned task (task 2 in this example). The entering task submits a bid – which suggests a margin higher than the current margin of vehicle 1 – to vehicle 1. Then, vehicle 1 sends a cancellation notice to the currently assigned task (task 2). Then, task 2 begins the same procedure as what the new entering did.

The following describes how to obtain a feasible assignment and a set of prices and margins that are at *almost equilibrium* through two bidding processes: ASSIGN-TASK-TO-A-NEW-VEHICLE and ASSIGN-VEHICLE-TO-A-NEW-TASK.

First, the procedure of ASSIGN-TASK-TO-A-NEW-VEHICLE is described in the following:

This process is triggered when a new vehicle becomes idle. Let A(i) be the set of tasks that can be assigned to vehicle i. Also, let the initial price of task j, p_j, be

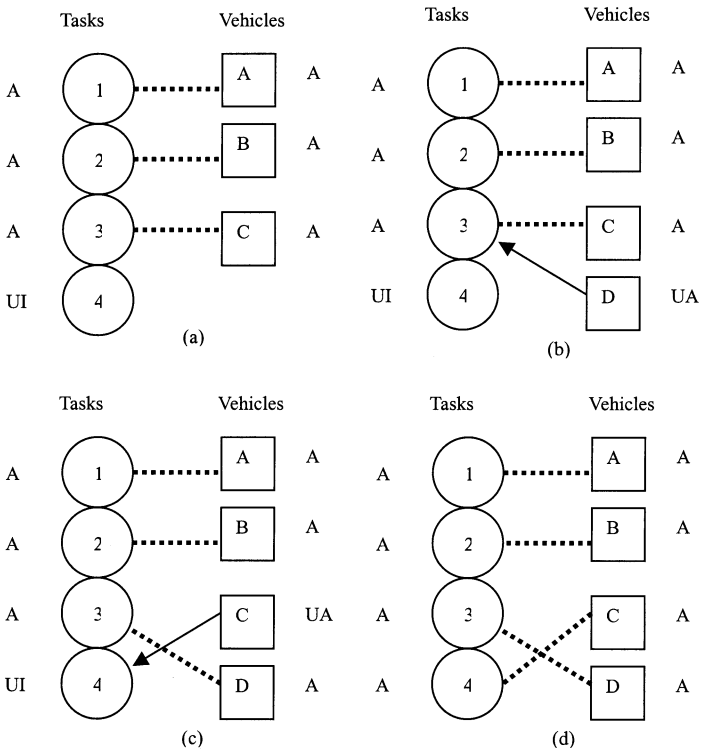

Fig. 1. An illustration of the ASSIGN-TASK-TO-A-NEW-VEHICLE

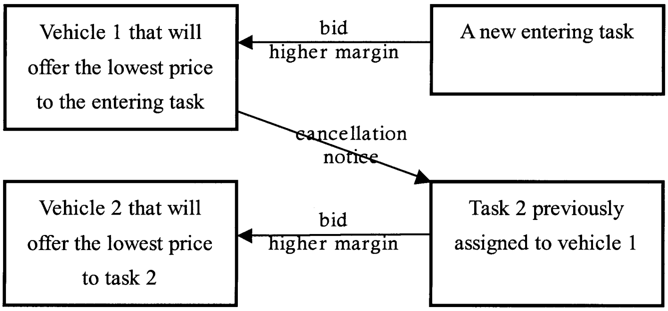

Fig. 2. Procedure of ASSIGN-VEHICLE-TO-A-NEW-TASK

a large number (p_0) for all j and the initial margin of vehicle i, v_i, be zero. The initial state of the new idle vehicle is set as UA.

Procedure for an unassigned vehicle: ASSIGN -TASK-TO-A-NEW-VEHICLE

The following procedure is repeated until no UA vehicle is found:

Preparing a bid by a UA vehicle

Let the UA vehicle be vehicle i.

Compute the current margin (profit) that vehicle i can earn by performing task $j \in A(i)$, which is given as

$$v_{ij} = \max\{p_j - a_{ij}, 0\}. \tag{2}$$

Find the best task j^* having the maximum value of

$$v_{ij^*} = \max{}_{j \in A(i)} v_{ij}.$$

(Task j^* will give vehicle i the maximum margin if it is assigned to vehicle i.)

If $v_{ij^*} = 0$, then it implies that there is no profitable task for vehicle i. (This happens when prices of tasks in A(i) are too low for vehicle i to obtain a positive margin by performing any task in A(i). In this case, vehicle i should remain idle.) Let $v_i = 0$ and the state of vehicle i be UI, and then stop.

Otherwise, find the highest margin offered by tasks other than j^*, which is given as

$$w_{ij^*} = \max{}_{j \in A(i), j \neq j^*} v_{ij} \tag{3}$$

If task j^* is the only task in A(i), then w_{ij^*} is set to be 0. (w_{ij^*} means the margin that vehicle i can earn when it is assigned to the second best task.)

Let the state of vehicle i be A. Compute the bid of vehicle i for task j^*. The price of the bid is given by

$$b_{ij^*} = p_{j^*} - v_{ij^*} + w_{ij^*} - \varepsilon. \tag{4}$$

(b_{ij^*} is the level of the price of task j^* that gives the same margin to vehicle i as the second best task of (3) does. Note that ε is related to the definition of "almost equilibrium.")

Submit the bid to task j^*.

Accepting the bid submitted to task j^*

If there is a vehicle that is currently assigned to task j^* (let it be vehicle i^*), make the state of vehicle i^* be UA. Assign task j^* to vehicle i and make the state of vehicle i be A. Let $p_{j*} = \min\{p_0, b_{ij^*}\}$ and $v_i = p_{j^*} - a_{ij}$. Inform the new assignment to vehicle i^* and i.

The following describes the ASSIGN-VEHICLE-TO-A-NEW-TASK, which is the procedure triggered when a new delivery task becomes available:

Procedure for unassigned tasks: ASSIGN-VEHICLE-TO-A-NEW-TASK

Let $V(k)$ be the set of vehicles that can be assigned to task k.

The following procedure is repeated until no UA task is found.

Preparing a bid by a UA task
Let the UA task be task k.

Compute the current minimum compensation required to induce each vehicle $i \in V(k)$, which is given by

$$c_{ik} = \min \{v_i + a_{ik}, p_0\}. \tag{5}$$

(Note that v_i represents the margin that vehicle i already secured.)

Find the best vehicle i^* having the minimum value

$$c_{i^*k} = \min{}_{i \in V(k)} c_{ik}.$$

If $c_{i^*k} = p_0$, then it implies that there is no vehicle that task k with the maximum price p_0 can afford. Then, let $p_k = p_0$ and the state of task k be UI, and then stop. (Task k will remain unassigned.)

Otherwise, find the minimum compensation for vehicles other than vehicle i^* by

$$d_{i^*k} = \min{}_{i \in V(k), i \neq i^*} c_{ik}. \tag{6}$$

(d_{i^*k} means the compensation that task k must pay so that it is to be assigned to the second best vehicle.)

If vehicle i^* is the only vehicle in $V(k)$, $d_{i^*k} = p_o$.

Let the state of task k be A.

Compute the bid of task k for vehicle i^*. The margin of the bid is given by

$$e_{i^*k} = v_{i*} - c_{i^*k} + d_{i^*k} + \varepsilon. \tag{7}$$

(e_{i^*k} is the level of the margin that vehicle i^* can earn when task k pays the same compensation to vehicle i^* as task k needs to do for inducing the second best vehicle of 7.)

Submit the bid to the manager of vehicle i^*.

Accepting the bid submitted to vehicle i^*
If there is a task that is currently assigned to vehicle i^* (let it be task j^*), make the state of task j^* be UA. Assign vehicle i^* to task k. Let $v_{i^*} = \min \{e_{i^*k}, p_0 - a_{i^*k}\}$ and $p_k = v_{i^*} + a_{iT^*k}$.

Inform the new assignment to tasks j^* and k.

Property 2: For a given set of candidate vehicles and tasks, procedures of ASSIGN-TASK-TO-A-NEW-VEHICLE and ASSIGN-VEHICLE-TO-A-NEW-TASK enable a feasible assignment to (P1), and a set of prices and margins that are almost at equilibrium in a finite number of iterations.

Proof. See Appendix.

The following provides a numerical example to illustrate step-by-step the distributed assignment procedure:

A scenario of the dynamic arrival of vehicles and delivery tasks is presented here. At the beginning, tasks 1 and 2 are waiting for vehicles. Next, AGV 1 and AGV 2 become available for assignment, one by one. The following shows how the assignment procedure is performed at each moment of the event. Table 1 shows

Table 1. Travel distance from locations of vehicles to pickup location of delivery tasks (unit: ft)

	Task 1	Task 2
AGV 1	360	260
AGV 2	700	200

Table 2. Assignment, prices, and margins for the example with two tasks and two AGVs

	Cost (assignment)		v_i
	Task 1	Task 2	
AGV 1	360 (1)	260 (0)	$240 - 2\varepsilon$
AGV 2	700 (0)	200 (1)	$300 - \varepsilon$
p_j	$600 - 2\varepsilon$	$500 - \varepsilon$	

the travel distance from the initial locations of vehicles to the pickup locations of the delivery tasks.

Let both the initial p_1 and p_2 be 1,000 ($= P_0$).

1) When AGV 1 becomes available,
$v_{11} = 1,000 - 360 = 640.v_{12} = 1,000 - 260 = 740$. Thus, j^*=2, and $w_{12^*} = 640.b_{12} = p_2 - v_{12} + w_{12} - \varepsilon = 900 - \varepsilon$. State of AGV 1, which was assigned to task 2, changes from UA to A. $p_2 = 900 - \varepsilon$ and $v_1 = 640 - \varepsilon$.
2) When AGV 2 becomes newly available,
$v_{21} = 1,000 - 700 = 300$, and $v_{22} = 900 - \varepsilon - 200 = 700 - \varepsilon$. Thus, $j^* = 2$ and $w_{22} = 300$. Also, $b_{22} = p_2 - v_{22} + w_{22} - \varepsilon = 500 - \varepsilon$. Task 2 is newly assigned to AGV 2. The state of AGV 1 changes from A to UA, while the state of AGV 2 becomes A. $p_2 = 500 - \varepsilon$, and $v_2 = 300 - \varepsilon$.
AGV 1 whose state became UA constructs a new bid, as follows: $v_{11} = 1000 - 360 = 640$, and $v_{12} = 500 - \varepsilon - 260 = 240 - \varepsilon$. Thus, $j^* = 1.w_{11} = 240 - \varepsilon$, and $b_{11} = p_1 - v_{11} + w_{11} - \varepsilon = 600 - 2\varepsilon$. AGV 1 is assigned to task 1. The state of AGV 1 changes from UA to A. $p_1 = 600 - 2\varepsilon, v_1 = 240 - 2\varepsilon$. Table 2 shows the final assignment, margins, and prices.

4 A simulation experiment

A simulation was conducted to evaluate the performance of the bidding-based dispatching method (BDM). Figure 3 and Table 3 show respectively the guide path layout and the flow requirement used for the experiment. Each department has the area of 100×100 ft^2, and stations on each path segment are located 30 ft away from the nearest intersection. Table 3 shows the sequence of workstations that each product must be processed on. It is assumed that all the processing times follow uniform distributions, with the parameters as shown in Table 4.

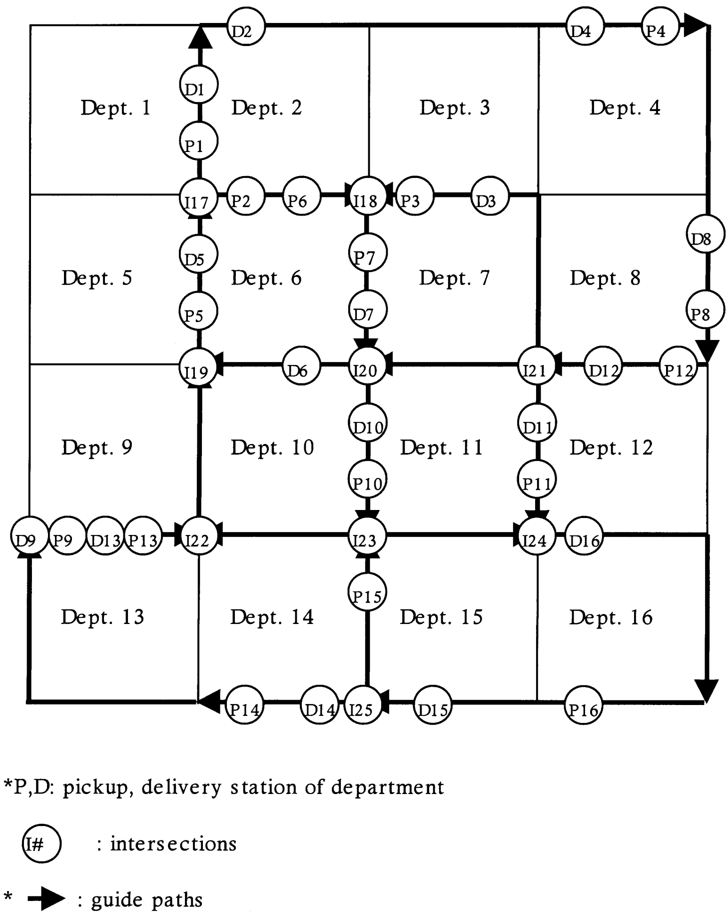

Fig. 3. Guide path and pickup and drop-off stations for vehicles

Table 3. Product mix and processing route

Products	Production mix (%)	Route (work station number)
Product 1	0.609	5-1-4-8-12-3-7-10-16
Product 2	0.229	6-7-10-16-14-9-13
Product 3	0.114	10-16-14-13-5-2-6
Product 4	0.048	5-8-11-16-15-1

The probability distributions of operation times shown in Table 4 were used in one of situations (Ex-3 in Table 5) assumed for the simulation. The average of the operation times shown in Table 4 is 4.5 minutes. However, the operation times were increased or decreased by multiplying every value in Table 4 by the same ratio for the simulation (See Table 5).

Table 4. Lower and uppper bounds of uniform distribution for the operation times at workstateion (unit: minutes)

Workstation	1	2	3	4	5	6	7	8
Bounds	[4,6]	[5,7]	[3,5]	[2,4]	[3,5]	[5,7]	[4,6]	[5,7]
Workstation	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]
Bounds	[4,6]	[4,6]	[1,3]	[3,5]	[6,8]	[3,5]	[2,4]	[1,3]

Table 5. Data for the first experiment

Situation number	The number of vehicles	Inter-arrival time of orders (min.)	Average processing time at workstations
Ex-3	3	5.00	4.50
Ex-4	4	3.75	3.38
Ex-5	5	3.00	2.70
Ex-6	6	2.50	2.25
Ex-7	7	2.14	1.93

The following assumptions are introduced for the simulation:

1) The capacities of input and output buffers are infinite.
2) The speed of vehicles is constant, and the transfer time for pallets between vehicles and workstations is zero. However, the result of the simulation would be the same even if a strict positive transfer time is assumed.
3) A vehicle can only move one unit-load at a time.
4) Vehicles travel on the shortest distance route.
5) The inter-arrival time of production orders follows an exponential distribution.
6) The empty travel time of a vehicle from the delivery location of task i to the pickup location of task j is used as the value of a_{ij}.

Idle vehicles and loaded vehicles are candidates for task assignment. That is, empty but assigned vehicles are excluded from candidates for task assignment. Delivery tasks become candidates for vehicle assignment only after delivery requests for them are issued. Even if the qualifying range for candidates of assignment is changed, BDM in this study remains valid.

The number of runs in the simulation was 10 for each problem. The simulation time was 20,000 minutes. The first 1,000 minutes were considered as a warm-up period and so the data collected during the warm-up period was discarded when various statistics were calculated. The simulation study was performed by using ARENA 3.5.

Although there exist many different types of weights (a_{ij}) of assignments for modeling the assignment problem, the deadhead travel time is a good candidate for the weight. Minimizing the total travel distance is expected to maximize the efficiency of the manufacturing system in the long run and also to provide a robust rule under various different situations.

As the reference rule for comparisons with BDM, the nearest-workstation rule was used as the vehicle-initiated rule, while the nearest-vehicle rule was used as the workstation-initiated rule (Egbelu, 1984), which was denoted as SDR (the shortest distance rule). That is, under the SDR, when a vehicle becomes idle, the workstation with a delivery task nearest to the new idle vehicle is selected for the next service, while, when a workstation calls for a vehicle, it chooses the idle vehicle nearest to the location of the workstation. Because BDM makes dispatching decisions based on the state of the system at the moment of the decision, it can be called a dynamic decision rule.

The following statistics were collected for comparing the two dispatching methods:

NOA: The Number of Orders Arrived at the shop
NOP: The Number of Orders Produced during the simulation period
RPA: The Ratio of the number of orders Produced to the number of orders Arrived (NOP/NOA). The higher value of RPA implies the higher throughput rate of the production system that is one of the ultimate performance measures of the material handling system.
RT: The Response Time, which is the time between a call for a vehicle and the arrival of a vehicle for the call. RT consists of the waiting time of a task until the assignment of a vehicle and the empty travel time of the assigned vehicle.
ETT: The Empty Travel Time from the delivery of the previous task to the arrival time at the corresponding pickup station
TS: The Time that an order stays in the production System. This performance measure is related to the work-in-process inventory.
AU: The average AGV Utilization that is the ratio of the sum of loaded travel time, “empty but assigned” travel time, and load transfer time to the total time spent by vehicles
AGVQ: The average number of delivery orders waiting for pickup by an AGV
NC: The Number of Communications occurred for one event that needs a dispatching decision in the case of BDM. Communication is necessary for sending bids, cancellation notice, new price, and new margin. Thus, it is desirable the dispatching process is completed with fewest possible communications.

Two experiments were conducted to evaluate the performance of BDM. In the first experiment, the number of vehicles was varied between 3 and 7. For Ex-3, data for the processing time, shown in Table 4, were used. However, processing times used for other situations were values in Table 4 multiplied by the ratio of the number of vehicles for Ex-3 to that of the corresponding situation. The inter-arrival time was also adjusted in the same way. The basic idea for the adjustment was to make the work-load of deliveries be proportional to the number of vehicles so that the work-load of deliveries per vehicle can be maintained at the same level, even in different situations. Also, the work-load on each workstation was maintained at

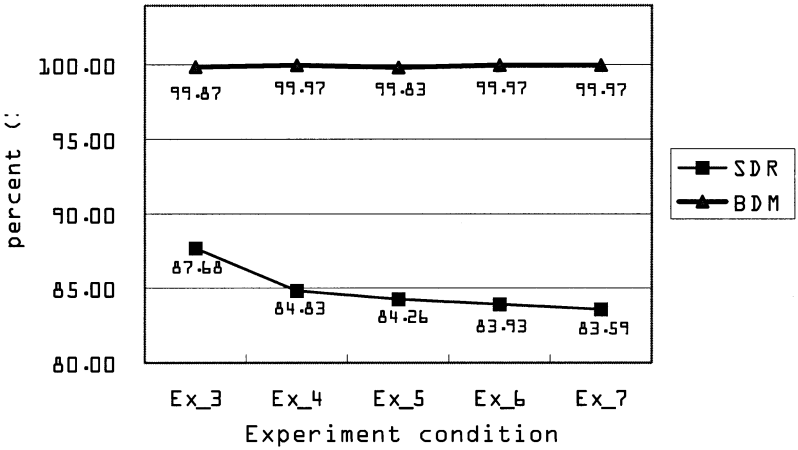

Fig. 4. Comparison of the percentage of completed orders between SDM and BDM

the same level so that the number of the works-in-process is maintained at a similar level. It was attempted to test the changes in performance measures for different numbers of vehicles and delivery tasks that participate to the auction process (the density of participants).

Table 6 shows the results of the experiment and each numeric value in Table 6 represents the average of results of the ten simulation runs. According to the results of the first experiment in Table 6, the production system using the SDR rule completed only 83–88 percent of all production orders issued during the planning horizon. In comparison, by using BDM (see Table 7), almost all production orders (over 99%) were completed during the same period. It was found that, when using SDR, the number of works-in-process waiting for vehicles (AGVQ) became higher and so the response time (RT) and the flow time (TS) were much longer than in the case using BDM. The poor performance of SDR is attributable to the longer empty travel time (ETT) in case of SDR than in the case of BDM. The last two columns of Table 7 show the maximum and the average numbers of communications, respectively. It is interesting to note that the average number of communications is less than 3, even when the number of vehicles is seven. Although the maximum number of communications went up to 120 when the number of vehicles became seven, it depends on the value of ε which was introduced to make the assignment algorithm terminate in a finite number of iterations and thus can be reduced by adjusting the value of ε. Also, note that in the result by BDM, the values of RT, ETT, and TS became smaller as the density of participants became higher. However, the results by SDR showed the trends opposite to those by BDM. The contrasting results comes from that BDM utilizes vehicles more efficiently than SDR does. Figure 4 shows the difference in the percentage of completed orders between two heuristic methods.

In the second experiment, the processing time and the inter-arrival time were maintained at the same level as Ex-5 in the first experiment. However, the number

Table 6. Result of the first experiment by using SDR

Situation	NOA	NOP	RPA	RT	ETT	TS	AU	Maximum AGVQ	Average AGVQ
Ex_3	3961	3473	87.7	203.0	1.63	1427	99.99	530	270.4
Ex_4	5350	4538	84.8	257.9	1.64	1761	100.00	859	457.5
Ex_5	6674	5623	84.3	257.5	1.65	1744	100.00	1104	568.4
Ex_6	7997	6712	83.9	263.7	1.66	1777	100.00	1359	696.7
Ex_7	9342	7809	83.6	277.3	1.66	1861	99.99	1623	853.5

Table 7. Result of the first experiment by using the BDM

Situation	NOA	NOP	RPA	RT	ETT	TS	AU	Max. AGVQ	Average AGVQ	Max NC	Average NC
Ex_3	4018	4013	99.9	2.0	1.10	119	93.84	10	1.3	38	1.6
Ex_4	5326	5325	100.0	1.7	1.05	56	92.40	11	1.3	57	1.8
Ex_5	6651	6640	99.8	1.5	1.01	70	91.35	10	1.2	75	1.9
Ex_6	7984	7982	100.0	1.4	0.97	66	90.26	11	1.3	91	2.0
Ex_7	9318	9315	100.0	1.4	0.95	58	89.56	12	1.5	120	2.2

Table 8. Result of the second experiment by using SDR

No. of AGVs	NOA	NOP	RPA	RT	ETT	TS	AU	Max AGVQ	Average AGVQ
3	6607	2285	34.6	1520.0	1.60	7036	100.00	4520	2362
4	6621	3857	58.3	794.5	1.63	4522	100.00	2891	1501
5	6674	5623	84.3	257.5	1.65	1744	100.00	1104	568
6	6652	6641	99.8	3.9	1.57	84	96.95	47	6
7	6686	6678	99.9	1.6	1.45	72.85	91.26	13	0.4

Table 9. Result of the second experiment by using BDM

No. of AGVs	NOA	NOP	RPA	RT	ETT	TS	AU AGVQ	Max AGVQ	Average Max	Max NC	Average NC
3	6684	5137	76.9	186.8	0.72	1517	100.00	1598	824	12	1.4
4	6686	6672	99.8	2.5	0.94	80	98.30	19	4	65	2.0
5	6651	6640	99.8	1.5	1.01	70	83.24	9	1	75	1.9
6	6681	6675	99.9	1.3	0.94	71	76.19	7	1	72	1.4
7	6590	6586	100.0	1.3	0.93	68	79.23	7	1	88	1.4

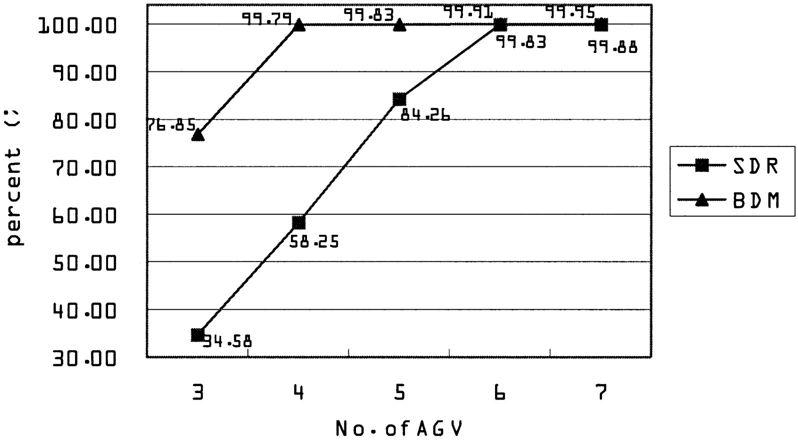

Fig. 5. Comparison of the percentage of completed orders between SDR and BDM for different number of vehicles

of vehicles was varied between 3 and 7. When the number of vehicles was high, the percentage of completed production orders was similar between the two dispatching methods. However, as shown in Tables 8 and 9, when the number of vehicles was low, the AGVs became the bottleneck of the production system, and the difference between two methods became significant. In Figure 5, the percentage of completed orders is compared between SDR and BDM for different number of vehicles

In certain layouts where some of the stations have favorable locations as opposed to others on account of reachability, it is known that SDR performs rather poorly, because calls from the unfavorable stations are usually not responded for very long durations. A common modification to the SDR for making up the drawback is to force a task to have the highest priority when the task had no response during a period longer than a specified time, which can be called "modified SDR." A dynamic algorithm as BDM would alleviate this problem, because multiple vehicles become candidates for assignment and thus there is a high chance for a vehicle to be assigned even to a distant station. However, it is still possible for a distant station to wait for a response for a longer period of time than other stations, because even in those rare occasions when a vehicle is assigned to a distant station, this decision will be quickly discarded as soon as a new task appears at a preferable station. For making up the possible drawback of BDM, it would be possible to modify BDM in a similar way to the modified SDR.

In this paper, a_{ij} was evaluated by using the empty travel time to pick up task j after performing task i. One of other alternatives for evaluating a_{ij} is to use the response time – which is the time from the current location of the vehicle performing task i to the pick-up location of task j – for minimizing the total response time of vehicles which is a possible objective of material handling systems. Also, there may be cases where delivery tasks have due times for pick-ups or deliveries. Typical examples are when workstations have a finite input or output buffer space for incoming or outgoing materials. In the case, the most important objective can be to minimize blocking or starvation of workstations because of the delayed transportation of materials by vehicles. Thus, each material may have a due time for pickup or delivery and the amount of delay – when the vehicle that is performing task i is assigned to task j – can be used as the value of a_{ij}.

5 Conclusion

A bidding-based method is suggested for dispatching automated guided vehicles. The dispatching method described in this study is different from other dynamic dispatching rules in that it looks into the future for an efficient assignment of delivery tasks to vehicles and that so it essentially assigns multiple tasks to multiple vehicles. The theoretical background of the bidding-based dispatching method is basically the auction method for the well-known assignment problem. The auction method developed by Bertsekas (1981) was modified to the dispatching method, and the rationale behind the bidding-based dispatching method is also provided.

A simulation was conducted to test the performance of the bidding-based dispatching method. It was shown that the bidding-based dispatching method significantly outperformed a popular dispatching rule (shortest distance rule) in throughput rate, flow time, response time, empty travel time, and average and maximum queue length of calls for vehicles. In addition, it was found that the number of communications among workstations and vehicles, which occurred during the dispatching decision process, was as low as three, even when the number of vehicles was seven. The low number of communications implies that the bidding-based dispatching process can be applied without a high load on the communication links among processors.

Because this paper introduces a new concept for distributed dispatching, only a limited number of practical factors were considered. However, in real systems, there may be more practical factors, such as types of guide paths, the size of buffer spaces, starvation, blocking, and deadlocks, which must be considered in the dispatching processing. Thus, further research is necessary on the method before it can be applied in practice. Also, empty vehicles traveling to pick up a load as well as the loaded vehicles and idle vehicles may be considered in the dispatching, which may be done in a future study. And, in this study, the minimization of the total empty travel time was considered as the objective function of the assignment problem. However, other objective functions may be used in different situations. More experiments are needed to study different objective functions.

Appendix: Proof of Property 2

The following four cases exist, any of which triggers either of the procedures:

Case 1: A new vehicle becomes available, and the number of available vehicles is smaller than or equal to the number of available tasks.

Case 2: A new vehicle becomes available, and the number of available vehicles is larger than the number of available tasks.

Case 3: A new task becomes available, and the number of available vehicles is smaller than or equal to the number of available tasks.

Case 4: A new task becomes available, and the number of available vehicles is larger than the number of available tasks.

Case 1: Each unassigned task (k) has the price of p_0, which is so high that every vehicle can get a positive margin when assigned to task k. When a vehicle becomes newly available, there are $(n-1)$ assigned vehicles, $(n-1)$ assigned tasks, and $(m-n+1)$ unassigned and inactivated tasks. The new vehicle arrives in an unassigned but activated (UA) state, and one vehicle remains "UA" until a vehicle becomes finally assigned to one of the $(m-n+1)$ UI tasks. As soon as a vehicle is assigned to one of the $(m-n+1)$ UI tasks, the procedure is terminated, and the results are n assigned vehicles, n assigned tasks, and $m-n$ UI tasks. At every iteration, the price of one assigned task decreases by $v_{ij^*} - w_{ij^*} + \varepsilon$ ($> \varepsilon >$0) from (4). Thus, in the long run, a UI task must be selected as the best task (j^*) by the UA vehicle. Thus, the procedure terminates in finite iterations. Once a vehicle submits a bid to a task, it means that the vehicle is almost happy about the price of the task (from 4) and remains so until the vehicle becomes "UA." It results from the fact that the price of no other task increases during the iteration. For the case of dummy vehicles, because the cost of assignment to every task is the same and the prices of unassigned tasks are p_0, which is high enough to make dummy vehicles have a positive margin, dummy vehicles must be almost happy for being assigned to any of the unassigned tasks.

Case 2: When the number of available vehicles is larger than the number of available tasks, $(n-m)$ vehicles become "unassigned" and cannot find a task that results in a positive margin. Because the prices of dummy tasks are the same and the cost of assigning a vehicle to every dummy task is the same, every unassigned vehicle must be almost happy for being assigned to any of the dummy tasks.

The proof for cases 3 and 4 can be derived as for cases 1 and 2 from the perspective of tasks instead of vehicles. Thus, the conclusion holds.

References

Bartholdi III JJ, Platzman LK (1989) Decentralized control of automated guided vehicles on a single loop. IIE Transactions 21: 76–81

Bertsekas DP (1990) The auction algorithm for assignment and other network flow problems: a tutorial. Interfaces 20(4): 133–149

Bertsekas DP (1981) A new algorithm for the assignment problem. Mathematical Programming 21: 152–171

Bilge Ü, Ulusoy G. (1995) A time window approach to simultaneous scheduling of machines and material handling system in an FMS. Operations Research 43(6): 1058–1070

Bilge, Ü, Tanchoco, JMA (1997) AGV systems with multi-load carriers. Journal of Manufacturing Systems 16(3): 159–173

Co CG., Tanchoco JMA (1991) A review of research on AGVS vehicle management. Engineering Costs and Production Economics 21: 35–42

Egbelu PJ, Tanchoco JMA (1984) Characterization of automatic vehicle dispatching rules. International Journal of Production Research 22: 359–374

Egbelu PJ (1987) Pull versus push strategy for automated guided vehicle load movement in a batch manufacturing system. Journal of Manufacturing Systems 6: 209–221

Hillier FS, Lieberman GJ (1986) Introduction to operations research, 4th edn., Chapter 6. Holden-Day

Kim CW, Tanchoco JMA, Koo PH (1999) AGV dispatching based on workload balancing. International Journal of Production Research 37(17): 4053–4066

Klein CM, Kim J (1996) AGV dispatching. International Journal of Production Research 34: 95–110

Lee J, Tangjarukij M, Zhu Z (1996) Load selecting of automated guided vehicles in flexible manufacturing systems. International Journal of Production Research 34 (12): 3383–3400

Sabuncuoglu I, Hommertzheim DL (1992) Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system. International Journal of Production Research 30: 1059–1079

Taghaboni-Dutta F (1997) A value-added approach for automated guided vehicle task assignment. Journal of Manufacturing Systems 16 (1): 24–34