European Journal of Operational Research 127 (2000) 565–573

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Theory and Methodology

Positioning automated guided vehicles in a loop layout

A.J.R.M. Gademann [a,b], S.L. van de Velde [c,*]

[a] *Center for Production, Logistics and Operations Management, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands*

[b] *ORTEC Consultants B.V., Postbus 490, 2800 AL Gouda, The Netherlands*

[c] *Rotterdam School of Management, Erasmus University, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands*

Received 1 July 1998; accepted 1 June 1999

Abstract

We address the problem of determining the home positions for m automated guided vehicles (AGVs) in a loop layout where n pickup points are positioned along the circumference ($m < n$). A home position is the location where idle AGVs are held until they are assigned to the next transportation task. The home positions need to be selected so as to minimize an objective function of the response times, where the response time for a pickup point is defined as the travel time to the pickup point from the nearest home location.

For the unidirectional flow system, where all AGVs can move in one direction only, we first point out that the problem of minimizing an arbitrary regular cost function can quite straightforwardly be solved in $O(n^2)$ time if $m = 1$ and in $O(mn^m)$ time if $m \geqslant 2$, which is polynomial for a fixed number m of AGVs. For $m \geqslant 3$, we can do better, however: we derive a generic $O(mn^3)$ time and $O(mn)$ space dynamic programming algorithm for minimizing any regular function of the response times. For minimizing maximum response time a further gain in efficiency is possible: this problem can be solved in $O(n^2)$ time if $m = 2$ and $O(n^2 \log n)$ time if $m \geqslant 3$. Our results improve on earlier published work, where it was suggested that problems with $m \geqslant 2$ are NP-hard.

For the bidirectional flow system, where the AGVs can move in both directions, the problem of determining the home locations is inherently much more difficult. Important objective functions like average response time and maximum response time can nonetheless still be minimized by the same types of algorithms and in the same amount of time as their unidirectional counterparts, once restrictive conditions apply such that the case $m = 1$ can be solved in polynomial time. One such restrictive condition is that each AGV travels at constant speed. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Location; Transportation; Automated guided vehicles; Loop layout; Dynamic programming

1. Introduction

Material handling is a significant part of the manufacturing process, both in terms of cost (Allegri, 1994; Tompkins and White, 1984) and

* Corresponding author. Tel.: +31-10-408-2025; fax: +31-10-452-3595.

E-mail addresses: a.j.r.m.gademann@cplom.utwente.nl, ngademann@ortec.nl (A.J.R.M. Gademann), s.velde@fac.fbk.eur.nl (S.L. van de Velde).

0377-2217/00/$ - see front matter © 2000 Elsevier Science B.V. All rights reserved.
PII: S 0 3 7 7 - 2 2 1 7 (9 9) 0 0 3 4 1 - 0

time. Indeed, the processing time of a typical job is only 5% of the manufacturing leadtime (Han and McGinnis, 1989); the remainder of the leadtime is spent in storage and in transportation by a material handling system.

Technological progress in hardware and software has led to the replacement of many conventional non-automated material handling systems, such as forklifts and rolling carts, with automated material handling systems, such as conveyor belts and automated guided vehicle systems. The task of automated guided vehicles (AGVs) is to pick up parts or items at certain points, usually storage depots and shop floor workstations, and to drop them off at others. AGV systems offer several advantages over conveyor belts and forklifts, such as higher flexibility, less space utilization, more safety and lower operating costs. The investment costs of AGV systems are relatively high, however, and the investment seems to be worthwhile only if the system is used intensively and efficiently (Ganesharajah et al., 1997).

The performance of an AGV system is generally a decreasing function of the service time, that is, the time between the time that a part becomes available for transportation and the delivery time. The service time consists accordingly of two components: waiting time and the travel time between the pickup point and the delivery point. In addition to AGV technology, various tactical design and operational control issues affect the performance of the system, including the design of the path layout, the location of pickup and delivery points, the number of AGVs in the system, and the scheduling and routing of the AGVs. We refer to Co and Tanchoco (1991), King and Wilson (1991), Johnson and Brandeau (1996) and Ganesharajah et al. (1997) for an overview and discussion of the various issues.

The topic of this paper is one such operational issue: the positioning of idle AGVs in a loop layout. AGV idleness is unavoidable when an AGV delivers a part and there is no pickup request to which the AGV can be assigned next. The positions of idle AGVs thus affect the empty travel times to the pickup points, and as such play a role in the waiting time of the parts. Accordingly, a clever positioning of idle AGVs may improve the performance of an AGV system. A *loop layout* is a single circuit where the pickup and delivery points are positioned along the circumference. It is an important practical layout: it offers greater efficiency and flexibility than a single line layout and is simple enough to avoid AGV interference problems and shop locking problems, which often occur in complex layouts (Egbelu and Tanchoco, 1984; Kusiak, 1985; Ganesharajah et al., 1997). Furthermore, loop layouts also form the components of so-called *tandem loops* of *tandem configurations*, which are suitable for more complex shop floor layouts (Bozer and Srinivasan, 1991, 1992).

The problem of positioning idle AGVs in a loop layout was first studied by Egbelu (1993), who suggested that the following three objectives may be used to determine the *home locations* of idle AGVs:

- minimizing the maximum response time of any pickup point, where the *response time* for a pickup point is the empty travel time from the nearest home location;
- minimizing the average response time;
- distributing the idle vehicles evenly in the network.

More specifically, Egbelu (1993) considered the problem of determining the home locations of m AGVs in a *loop layout* with n pickup points to minimize the first objective, that is, to minimize the maximum response time. He differentiated between two types of loop layouts: those with a *unidirectional* flow system, where all AGVs move in one and the same direction; and those with a *bidirectional* flow system, where the AGVs can travel in both directions. Furthermore, he commented on the practical complications of multi-AGV bidirectional flow systems, as AGVs moving in opposite directions may give serious interference and control problems.

Egbelu (1993) showed that the problem of minimizing maximum response time is solvable in polynomial time in case of a single AGV ($m = 1$), both in case of a unidirectional and a bidirectional flow system. He suggested that the problem is NP-hard if $m \geqslant 2$ for either type of system. For the multi-AGV unidirectional problem, he gave an integer non-linear programming formulation and presented a heuristic for its solution. For the bidirection problem, he proposed two heuristics.

Kim (1995) considered the problem of positioning a single idle AGV to minimize the average response time. He showed that both the static version of the problem, in which $m = 1$, as well as a specific dynamic version of the problem, where $m - 1$ AGVs have already been located, can be solved quite straightforwardly in $O(n^2)$ time.

In this article, we establish that the problems of minimizing maximum response time and minimizing average response time are solvable in polynomial time for *any* number of AGVs, in either type of flow system, thereby refuting Egbelu's suggestion that these problems are NP-hard if $m \geqslant 2$.

More specifically, our contribution is as follows. For the unidirectional flow system, we first point out in Section 3.1 a property of a class of optimal solutions for minimizing an arbitrary *regular* function of the response times and observe that as a result of this property *any* such function can quite straightforwardly be minimized in $O(n^2)$ time if $m = 1$ and in $O(mn^m)$ time if $m \geqslant 2$, which is polynomial when m is fixed. We call a response function regular if it is non-decreasing in the response times.

We can do better for $m \geqslant 3$, however. In Section 3.2, we derive a generic $O(mn^3)$ time and $O(mn)$ space dynamic programming algorithm for minimizing any regular function of the response times. A still further gain is possible for minimizing maximum response time. In Section 3.3, we show that this problem is solvable in $O(n^2 \log n)$ time for $m \geqslant 3$.

The bidirectional flow system is addressed in Section 4. First of all, it appears that in general it is not possible to determine the optimal home locations in polynomial time, even if $m = 1$. However, in specific situations where there is a polynomial subroutine for solving the case $m = 1$, any regular function of the response times can still be minimized in polynomial time by the same type of algorithm as its unidirectional counterpart by calling this subroutine a polynomial number of times. All AGVs traveling at constant speed is one such a specific situation. Indeed, in this case, average response time can still be minimized in $O(mn^3)$ time and $O(mn)$ space and maximum response time can still be minimized in $O(n^2 \log n)$ time.

Section 5 concludes our paper with some final remarks. Before we proceed to the heart of the paper, however, Section 2 presents a formal problem description, the notation used, and the assumptions made.

2. Problem description, notation, and assumptions

In this paper, we address the problem of determining the home locations of m AGVs circulating in a loop layout with n pickup and delivery points. To avoid trivialities, assume that $n > m$ and that all pickup points are distinct.

Without loss of generality, assume that the n pickup points have been consecutively numbered in the clockwise direction. It is immaterial at which pickup point the numbering was started. For matter of notational convenience, we also introduce pickup points 0 and $n + 1$ and let by definition pickup point 0 be equal to pickup point n and pickup point $n + 1$ be equal to pickup point 1. Let t_{ij} denote the travel time between pickup points i and j $(i = 1, \ldots, n,\ j = 1, \ldots, n)$. By default, let $t_{ii} = 0$ for $i = 1, \ldots, n$.

Let q be the number of distinct positive travel times. Note that $q \leqslant n(n - 1)$. Furthermore, let $t_{[k]}$ denote the kth smallest of these distinct travel times $(k = 1, \ldots, q)$. Accordingly, $t_{[k]}$ corresponds to the travel time between at least one pair of pickup points, say, (s_k, e_k) $(k = 1, \ldots, q)$. If $t_{[k]}$ corresponds to more than one pair, then without loss of generality we let (s_k, e_k) be the pair of pickup points that corresponds to the lexicographical minimum.

Note that we require no explicit knowledge of and make no restrictive assumptions about the accelerating, decelerating, and traveling speeds of the AGVs and the distances between the pickup points – we assume that they have been used in a preprocessing phase to compute the traveling times t_{ij} $(i = 1, \ldots, n,\ j = 1, \ldots, n)$.

Any positioning ω of the AGVs specifies for each pickup point j $(j = 1, \ldots, n)$ a response time $T_j(\omega)$, which is defined as the travel time from the nearest home location. The home location of an

AGV does not necessarily have to be one of the pickup points. When there is no ambiguity, $T_j(\omega)$ is abbreviated to T_j $(j = 1, \ldots, n)$. Each pickup point j $(j = 1, \ldots, n)$ has a response cost function f_j, where $f_j(t)$ denotes the *cost* incurred if the response time for pickup point j is equal to t. We consider only *regular* cost functions, i.e., throughout it is assumed that each $f_j(t)$ is a nondecreasing function of t, for $j = 1, \ldots, n$.

Furthermore, like Egbelu (1993), we address only the *static* variant of the problem, in which all m AGVs are idle at the same time and no pickup requests are available. This variant is relevant, for instance, for single shift operations, where the AGVs need to be strategically positioned at the beginning of the shift, or for situations in which all AGVs are idle from time to time. In *dynamic* variants of the problem, some AGVs are idle at their home locations, some are idle and on the way to pickup a request, while the rest are busy transporting items. These variants can be defined in different ways and lead to different optimization problems. The dynamic variant considered by Kim (1995) can be reduced to the static variant, and for this type of dynamic variant our results are of direct use. For more complicated dynamic variants, however, it seems that a stochastic optimization approach is required.

Not only do we consider the static variant, we also assume to have no deterministic knowledge about future tasks, when and where they will arise. Under this assumption, it may be possible, however, that we know the probability $P_j \geqslant 0$ that the next task is issued by the pickup station j $(j = 1, \ldots, n)$.

The objective is to minimize the response cost, measured either by a regular minmax objective function $f_{\max} = \max_{1 \leqslant j \leqslant n}\{f_j(T_j)\}$, or by a regular minsum objective function $\sum_{j=1}^{n} f_j = \sum_{j=1}^{n} f_j(T_j)$. Important regular objective functions that we consider explicitly are the *maximum response time* $T_{\max}$, defined as $T_{\max} = \max_{1 \leqslant j \leqslant n}\{T_j\}$, and *average response time* $\overline{T}$, defined as $\overline{T} = (1/n)\sum_{j=1}^{n} T_j$ – note that these objective functions were studied by Egbelu (1993) and Kim (1995). In case we know the probabilities P_j $(j = 1, \ldots, n)$, then relevant objective functions are *expected maximum response time* and *expected average response time*. They are in fact special cases of regular minmax and minsum objective functions, however, and for this reason we will not consider them explicitly.

3. The unidirectional flow system

3.1. Preliminaries

For minimizing the maximum response time, Egbelu (1993) observes that there exists an optimal solution in which the home positions of the AGVs coincide with m distinct pickup points. This observation, however, applies to *any* regular function of the response times. To see this, consider any optimal solution in which some AGV is located in between two pickup points, say, j and $j + 1$ – because of unidirectionality, relocating this AGV to pickup point $j + 1$ will definitely improve the response time of this AGV, and hence never decrease the objective function value. Furthermore, consider any optimal solution in which two or more AGVs are positioned at the same pickup point. Relocating one of them to a pickup point where no AGV is stationed will never decrease the objective function value.

For the special case $m = 1$, we can accordingly minimize any regular function of the response times in $O(n^2)$ time by evaluating the objective function for all possible n home locations. For $m \geqslant 2$, the above property implies that we may restrict ourselves in our search for an optimal solution to all positionings ω that can be represented by a string of m pickup points $(w_1, w_2, \ldots, w_m)$ with $w_i < w_{i+1}$ for $i = 1, \ldots, m$. We say then that the AGV positioned at pickup point w_i $(i = 1, \ldots, m)$ *covers* the *zone* from w_i up to and including $w_{i+1} - 1$. For any given assignment ω of AGVs to pickup points, we can compute the objective function value in only $O(m)$ time, if we have evaluated and stored the partial sums $\sum_{j=l}^{k} f_j(t_{l,j})$ and partial maxcost coefficients $\max_{l \leqslant j \leqslant k}\{f_j(t_{l,j})\}$ for $k = 1, \ldots, n$, $l = 1, \ldots, k$ in a preprocessing step. This preprocessing requires only $O(n^2)$ time.

These properties imply that any regular function can be minimized in $O(mn^m)$ time, which is polynomial for fixed m. This is achieved by simply enumerating all $\binom{n}{m}$ possible ways of assigning m

AGVs to n pickup points, evaluating the objective value of each assignment, which takes $O(m)$ time, and storing the best solution. Actually, this naive procedure is the best possible for $m = 2$. For $m \geqslant 3$, however, there exists a faster algorithm, as is shown in Section 3.2.

3.2. Minimizing an arbitrary regular function

This section presents a polynomial-time dynamic programming algorithm for minimizing an arbitrary regular function of the response times, which runs in $O(mn^3)$ time and $O(mn)$ space. The algorithm uses a *forward* enumeration scheme in which AGVs are successively assigned to the current partial assignment. Consider any assignment of $k \geqslant 1$ AGVs to the pickup points $i, i+1, \ldots, j-1, j$ subject to the condition that the first AGV is positioned at pickup point i and the kth AGV is positioned at pickup point j, that is, a partial assignment with $w_1 = i$ and $w_k = j$. We define such an assignment to be in *state* (i, j, k). Of course, to assign the remaining $(m-k)$ AGVs to the pickup points $j+1, \ldots, n-1, n, 1, \ldots, i-1$, we need to consider only an assignment with minimum objective value among all assignments in state (i, j, k).

Let ω be an assignment with minimum objective value in state (i, j, k) with $k \geqslant 2$. To achieve this state from a previous state, we must decide which pickup points are covered by the $(k-1)$th AGV to create ω. Accordingly, the previous state must be $(i, l, k-1)$ for some $i+k-2 \leqslant l < j$, in which case the $(k-1)$th AGV covers the pickup points $l, \ldots, j-1$. The result is then that the total cost increases by $\sum_{h=l}^{j-1} f_h(t_{l,h})$ in case of an arbitrary regular minsum cost function. If we have an arbitrary regular minmax function, then the maximum cost of the pickup points $l, \ldots, j-1$ is $\max_{l \leqslant h \leqslant j-1}\{f_h(t_{l,h})\}$.

This optimality principle leads to a polynomial-time dynamic programming algorithm for minimizing any regular objective function. Let $F_i(j,k)$ be the minimum objective value for assigning k AGVs to the zone $i, \ldots, j$ subject to $w_1 = i$ and $w_k = j$. We are now ready to give the dynamic programming recursion. The initialization is

$$F_i(j,k) = \begin{cases} 0 & \text{if } j = i \text{ and } k = 1, \\ \infty & \text{otherwise.} \end{cases}$$

If the objective function is of the minmax type, then the recursion for $k = 2, \ldots, m$, $j = i+k-1, \ldots, n$ is given by

$$F_i(j,k) = \min_{i+k-2 \leqslant l < j} \left\{ \max\left\{ F_i(l, k-1), \max_{l \leqslant h < j}\{f_h(t_{l,h})\} \right\} \right\}.$$

If the mth AGV has been assigned to pickup point j $(j = i+m-1, \ldots, n)$, then it needs to cover the pickup points $j+1, \ldots, n, 1, \ldots, i-1$, since we set $w_1 = i$, that is, we located the first AGV to pickup point i. Thus, the maximum cost of the pickup points $j, \ldots, n, 1, \ldots, i-1$ is equal to $\max_{j \leqslant h \leqslant n, 1 \leqslant h < i}\{f_h(t_{j,h})\}$. We have then that the optimal solution value for $w_1 = i$, that is, given that the first AGV is located at pickup point i, is equal to

$$F_i^* = \min_{i+m-1 \leqslant j \leqslant n} \left\{ \max\left\{ F_i(j,m), \max_{j \leqslant h \leqslant n, 1 \leqslant h < i}\{f_h(t_{j,h})\} \right\} \right\}.$$

If the objective function is of the minsum type, then the recursion for $k = 2, \ldots, m$, $j = i+k-1, \ldots, n$ is given by

$$F_i(j,k) = \min_{i+k-2 \leqslant l < j} \left\{ F_i(l, k-1) + \sum_{h=l}^{j-1} f_h(t_{l,h}) \right\}.$$

If the mth AGV has been assigned to pickup point j $(j = i+m-1, \ldots, n)$, then it needs to cover the pickup points $j, \ldots, n, 1, \ldots, i-1$, since we set $w_1 = i$. The total response cost for pickup points $j, \ldots, n, 1, \ldots, i-1$ is thus equal to

$$\sum_{h=j}^{n} f_h(t_{j,h}) + \sum_{h=1}^{i-1} f_h(t_{j,h}).$$

Accordingly, the optimal solution value given $w_1 = i$ is then equal to

$$F_i^* = \min_{i+m-1 \leqslant j \leqslant n} \left\{ F_i(j,m) + \sum_{h=j}^{n} f_h(t_{j,h}) + \sum_{h=1}^{i-1} f_h(t_{j,h}) \right\}.$$

For both types of objective functions, the overall optimal solution value is equal to $\min_{1 \leqslant i \leqslant n-m+1} F_i^*$ and the corresponding optimal home locations are found by backtracing.

To implement these algorithms efficiently, the partial sums $\sum_{h=l}^{j} f_h(t_{l,h})$ and the maxcost coefficients $\max_{l \leqslant h \leqslant j}\{f_h(t_{l,h})\}$ are evaluated and stored for $l = 1, \ldots, n$ and $j = l+1, \ldots, n$ in a preprocessing step, which takes $O(n^2)$ time. Computing each $F_i(j,k)$ value takes $O(n)$ time, and since there are $O(mn^2)$ different $F_i(j,k)$ values, the entire algorithm can be implemented to run in $O(mn^3)$ time and $O(mn)$ space. In Section 3.3, it is shown that a further gain in efficiency is possible for the problem of minimizing maximum response time.

3.3. Minimizing maximum response time

This section presents a straightforward $O(n^2 \log n)$ algorithm for minimizing maximum response time in a unidirectional flow system for any $m \geqslant 2$. Note that for any assignment of AGVs to pickup points the maximum response time $T_{\max}$ is easily computed as

$$T_{\max} = \max_{1 \leqslant j \leqslant m} \{t_{w_j, w_{j+1}-1}\}.$$

If $T^*_{\max}$ denotes the minimum maximum response time, then $T^*_{\max} = t_{[k]}$ for some a priori unknown k $(1 \leqslant k \leqslant q)$. (Recall that $t_{[k]}$ is the kth smallest travel time and that q is the number of distinct positive travel times.) Since we do not know k beforehand we need to guess it – this guessing proceeds in a systematic manner.

First of all, note that the problem of minimizing the maximum response time can be viewed as a finite series of decision problems of the type "is $T^*_{\max} \leqslant t_{[k]}$?" for a *given* k, where k is repeatedly adjusted by binary search over the values $1, \ldots, q$. If this decision problem can be solved in polynomial time, then the optimization problem can be solved in polynomial time, since k need be adjusted no more than $\lceil \log n(n-1) \rceil$ times. The notation $\lceil x \rceil$ refers to the smallest integer greater than or equal to x.

The decision problem for any given k can be solved in $O(m \log n)$ time in the following way. We position the first AGV at pickup point s_k and the second AGV at pickup point $e_k + 1$ – hence, this first AGV covers a zone of length exactly $t_{[k]}$. Each next AGV is assigned to the pickup point that is as far away from the last assigned AGV as possible but no further away than $t_{[k]}$. Two cases need to be distinguished:

- The circuit can be covered with no more than m AGVs to guarantee the maximum response time $t_{[k]}$ – hence, the question "is $T^*_{\max} \leqslant t_{[k]}$?" has an affirmative answer.
- The travel time between the home location of the mth AGV and the last pickup point before s_k, where the first assigned AGV is located, is larger than $t_{[k]}$ – in this case, the answer to the question is no.

Since each next pickup point can be determined in $O(\log n)$ time by binary search over the remaining pickup points and $m-2$ pickup points need to be assigned, it takes $O(m \log n)$ time to solve the decision problem. The entire procedure requires $O(m \log^2 n)$ time, since we need to solve no more than $\lceil \log n(n-1) \rceil$ decision problems, once the travel times have been sorted in non-decreasing order. The sorting requires $O(n^2 \log n)$ time, however, and it is therefore more time-consuming than solving the series of decision problems. Accordingly, this algorithm solves the problem to optimality in $O(n^2 \log n)$ time for any $m \geqslant 2$. Recall, however, that we have shown in Section 3.1 that the case $m = 2$ can be solved in $O(n^2)$ time by explicit enumeration.

For an arbitrary regular minmax function, essentially the same algorithm can be applied to solve the problem to optimality. The only difference is that binary search for the optimal solution value takes place over a different interval. Nonetheless, if the optimal solution value is an integer whose logarithm is polynomially bounded in the size of the input, then the problem is still solvable in polynomial time.

4. The bidirectional flow system

4.1. Preliminaries

For any arbitrary regular function of the response times in a bidirectional flow system, there is an optimal solution in which each AGV covers a certain zone between two pickup points such that these zones have no pickup point in common.

The search for an optimal positioning of AGVs can therefore be restricted to those that can be represented by a string of m pickup points $(w_1, \ldots, w_m)$ with $w_i < w_{i+1}$ for $(i = 1, \ldots, m)$. This string indicates that the ith AGV covers the zone $w_i, \ldots, w_{i+1} - 1$ for each i $(i = 1, \ldots, m)$. In the remainder, we refer to such a string of m distinct pickup points as a *partitioning* of the loop layout into m distinct zones.

Unfortunately, minimizing an arbitrary regular function of the response times in a bidirectional flow system is in general much harder than in a unidirectional flow system. The reason is that we may no longer restrict ourselves to home locations that concur with pickup points. Consider for example the situation with a single AGV and $n = 2$ pickup points, which are located on a single line layout; note that a single line layout is a special case of a loop layout. Since the AGV may move in both directions, the optimal home location could be anywhere in between, and to find this location we need to minimize the given response cost function of a single variable, i.e., the position on the line. Here we come across two problems, which also arise for the general home location problem with m AGVs and n pickup points in a loop layout:

- There are nonlinear functions that cannot be minimized in polynomial time. There exist for instance functions that can be handled only by methods that converge in an *infinite* number of steps, such as the Golden Section Method (Wilde and Beightler, 1967).
- Since the home location could be anywhere between the two pickup points, we would need to know the travel times between *any* arbitrary point on the line and the two pickup points.

Hence, in general, there is little that can be done analytically.

4.2. A well-solvable case

If we assume that the optimal positioning of an AGV in any given zone requires polynomial time, that is, the cost of each partitioning can be computed in polynomial time, then the positioning problem can be solved in polynomial time for any number of AGVs. In fact, the bidirectional home location problems can then be solved by essentially the same algorithms as their unidirectional counterparts.

For $m = 1$ and $m = 2$, the fastest method is still to enumerate all $\binom{n}{m}$ partitionings, optimally locate the m AGVs within each zone, evaluate the cost of each partitioning, and store the best one. Accordingly, the problem can be solved in $O(n^2)$ if $m = 1$ or $m = 2$, once the costs of all partitionings have been computed.

For $m \geqslant 3$, we can solve the problem by essentially the same type of dynamic programming algorithm as its unidirectional counterpart. Consider a partial partitioning of the loop in which $w_1 = i$ and $w_k = j$, i.e., a partitioning where $k \geqslant 1$ AGVs cover the zone between i and j, that is, the pickup points $i, i+1, \ldots, j-1, j$. We define such a partitioning to be in state (i, j, k). To position the remaining $m - k$ AGVs we need to consider only a partitioning with minimum objective value among all partitionings in this state. In analogy to the unidirectional case, we can derive a partitioning in state (i, j, k) $(k \geqslant 2)$ only from some partitioning in a previous state $(i, l, k-1)$ for some $i + k - 2 \leqslant l < j$, where $w_1 = i$ and $w_{k-1} = l$. We only give the dynamic programming recursion for an arbitrary regular minsum objective function $\sum_{j=1}^{n} f_j$ – the recursion for an arbitrary regular minmax objective function proceeds in a similar fashion.

Let now $F_i(j, k)$ be the minimum objective value for positioning k AGVs such that $w_1 = i$ and $w_k = j$. Furthermore let $c_{i,j}$ denote the minimum total cost if a single AGV covers the zone $i, i+1, \ldots, j-1, j$ $(i = j+1, \ldots, n,\ j = 1, \ldots, n)$; by default, we let $c_{1,0} = 0$. The initialization of the dynamic programming recursion is

$$F_i(j,k) = \begin{cases} 0 & \text{if } j = i \text{ and } k = 1, \\ \infty & \text{otherwise,} \end{cases}$$

and the recursion for $k = 2, \ldots, m,\ j = i + k - 1, \ldots, n$ is given by

$$F_i(j,k) = \min_{i+k-2 \leqslant l < j} \left\{ F_i(l, k-1) + c_{l,j-1} \right\}.$$

If the mth AGV has been assigned to pickup point j ($j = i + m - 1, \dots, n$), then the total response cost for pickup points $j, \dots, n, 1, \dots, i - 1$ is equal to $c_{j,n} + c_{1,i-1}$, since $w_1 = i$. Hence, the optimal solution value given $w_1 = i$ is then equal to

$$F_i^* = \min_{i+m-1 \leqslant j \leqslant n} \{F_i(j, m) + c_{j,n} + c_{1,i-1}\}.$$

The overall optimal solution value is equal to $\min_{1 \leqslant i \leqslant n-m+1} F_i^*$ and the corresponding optimal home locations are found by backtracing. Hence, once the cost coefficients $c_{i,j}$ have been computed, the entire algorithm requires $O(mn^3)$ time and $O(mn)$ space.

4.2.1. Constant traveling speed

If each AGV travels at constant speed between any pair of adjacent home locations, then each AGV can be optimally positioned in *constant* time both for average response time and maximum response time, once its zone has been specified.

For minimizing average response time, there is an optimal solution in which the AGVs are stationed at m distinct pickup points. This means that the cost coefficients $c_{i,j}$ can be computed in $O(n^2)$ time in a preprocessing step; hence, average response time in a bidirectional flow system can be minimized in $O(n^2)$ time if $m = 1$ or $m = 2$ and $O(mn^3)$ time and $O(mn)$ space if $m \geqslant 3$, just as in a unidirectional system.

For the problem of minimizing maximum response time, each AGV must be located *exactly* halfway between the two end stations of each zone. This means that

$$T_{\max}^* = \frac{1}{2} \max_{1 \leqslant i \leqslant m} \{t_{w_i, w_{i+1}-1}\},$$

where $T_{\max}^*$ is the minimum maximum response time. Hence, $T_{\max}^* = (1/2)t_{[k]}$ for some a priori unknown k ($k = 1, \dots, q$). The problem of finding $T_{\max}^*$ and a corresponding optimal partitioning can be solved by essentially the same procedure as the one described in Section 3.3 for the unidirectional case. Accordingly, the problem of minimizing maximum response times in a bidirectional loop layout with constant speed can be solved in $O(n^2 \log n)$ time.

5. Conclusions

Our results constitute a complete complexity mapping of determining the home locations of idle AGVs in a loop layout with either a unidirectional or a bidirectional flow system. For the unidirectional flow system, any regular function can be minimized in polynomial time; Table 1 gives an overview of time complexities. Determining the home locations of AGVs in a bidirectional flow system is much harder. In general, this problem is not solvable in polynomial time, even in case of a single AGV. However, under a restrictive assumption on the objective function, which seems to be quite mild from a practical point of view, the home location problem can still be solved in polynomial time. Furthermore, if we assume constant traveling speed of the AGVs, then minimizing maximum response time and minimizing average response time in a bidirectional flow system require as much time and space as their unidirectional counterparts.

It is likely that determining the home locations of idle AGVs in more complex layouts is more

Table 1
Overview of time complexities

Objective function	Unidirectional		
	$m = 1$	$m = 2$	$m \geqslant 3$
$f_{\max}$	$O(n^2)$	$O(n^2)$	$O(mn^3)$
$\sum_{j=1}^n f_j$	$O(n^2)$	$O(n^2)$	$O(mn^3)$
$T_{\max}$	$O(n)$[a]	$O(n^2)$	$O(n^2 \log n)$
$\sum_{j=1}^n \overline{T}_j$	$O(n^2)$[b]	$O(n^2)$	$O(mn^3)$

[a] See Egbelu (1993).
[b] See Kim (1995).

difficult – it would be interesting to see for what types of layouts these problems can still be solved in polynomial time. Finally, we have only addressed the static setting of the home location problem, in which all AGVs are assumed to be idle at the same time. Part of our future research is the dynamic setting of the problem, in which some AGVs are idle and others are busy transporting items.

References

Allegri, T.H., 1994. Material Handling: Principles and Practice. Van Nostrand Reinhold, New York.

Bozer, Y.A., Srinivasan, M.M., 1991. Tandem configurations for automated guided vehicle systems and the analysis of single vehicle loops. IIE Transactions 23, 72–82.

Bozer, Y.A., Srinivasan, M.M., 1992. Tandem AGV systems: a partitioning algorithm and performance comparison with conventional AGV systems. European Journal of Operational Research 63, 173–191.

Co, G.C., Tanchoco, J.M.A., 1991. A review of research on AGVS vehicle management. Engineering Costs and Production Economics 21, 35–42.

Egbelu, P.J., 1993. Positioning of automated guided vehicles in a loop layout to improve response time. European Journal of Operational Research 71, 32–44.

Egbelu, P.J., Tanchoco, J.M.A., 1984. Characterization of automatic guided vehicle dispatching rules. International Journal of Production Research 22, 359–374.

Ganesharajah, T., Hall, N.G., Sriskandarajah, C., 1997. Scheduling, design and decomposition decisions in AGV-served flowshop loops, Working Paper, University of Toronto, Toronto, Canada.

Han, M.-H., McGinnis, L.F., 1989. Control of material handling transporter in automated manufacturing. IIE Transactions 21, 184–190.

Johnson, M.E., Brandeau, M.L., 1996. Stochastic modeling for automated material handling system design and control. Transportation Science 30, 330–350.

Kim, K.H., 1995. Positioning of automated guided vehicles in a loop layout to minimize the mean vehicle response time. International Journal of Production Economics 39, 201–214.

King, R.E., Wilson, C., 1991. A review of automated guided-vehicle systems design and scheduling. Production Planning and Control 2, 44–51.

Kusiak, A., 1985. Material handling in flexible manufacturing systems. Material Flow 2, 79–95.

Tompkins, J.A., White, J.A., 1984. Facilities Planning. Wiley, New York.

Wilde, D.J., Beightler, C.S., 1967. Foundations of Optimization. Prentice-Hall, Englewood Cliffs, NJ.