INT. J. PROD. RES., 2002, VOL. 40, NO. 3, 745–760

Scheduling and routing algorithms for AGVs: a survey

LING QIU†, WEN-JING HSU†*, SHELL-YING HUANG†
and HAN WANG‡

Automated guided vehicles (AGVs) are now becoming popular in automated materials handling systems, flexible manufacturing systems and even container handling applications. In the past few decades, much research has been devoted to the technology of AGV systems and rapid progress has been witnessed. As one of the enabling technologies, scheduling and routing of AGVs have attracted considerable attention. Many algorithms for the scheduling and routing of AGVs have been proposed. However, most of the existing results are applicable to systems with a small number of AGVs, offering a low degree of concurrency. With a drastically increased number of AGVs in recent applications (e.g. in the order of a hundred in a container handling system), efficient algorithms are needed to resolve the increased contention of resources (e.g. path, loading and unloading buffers) among AGVs. This survey paper first gives an account of the emergence of the problem of AGV scheduling and routing. It then differentiates it from several related problems and classifies major existing algorithms for the problem. Finally, the paper points out fertile areas for future study of AGV scheduling and routing.

1. Introduction

Automated guided vehicles (AGVs) are becoming popular in automatic materials-handling systems, flexible manufacturing systems and even container-handling applications in seaports of late (Evers and Koppers 1996, Kim and Bae 1999, Ye *et al.* 2000).

Since the invention of AGVs, much research has been devoted to the technology of AGV systems, and rapid progress has been witnessed. As one of the enabling technologies, the scheduling and routing of AGVs has also attracted considerable attention (e.g. De Guzman *et al.* 1997, Huang and Hsu 1994, Kim and Bae 1999, Qiu and Hsu 2000a). Many algorithms about the problem have been proposed. In this paper, we first present a description of the problem, pointing out that scheduling and routing are two related aspects of the problem. We then discuss some common hazards in scheduling and routing of AGVs and techniques to handle them. Several similar problems are also compared and contrasted with the problem, which suggests that specially tailored approaches are needed. We then survey the existing major works on AGV scheduling and routing and give classifications. Finally, we recommend a few fertile areas for further study of this problem.

Revision received July 2001.

† Centre for Advanced Information Systems, School of Computer Engineering, Nanyang Technological University, Singapore 639798.

‡ School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798.

* To whom correspondence should be addressed. e-mail: hsu@ntu.edu.sg

International Journal of Production Research ISSN 0020–7543 print/ISSN 1366–588X online © 2002 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/00207540110091712

1.1. *Origin of the problem*

Like a computer system, an AGV system is normally composed of two main interacting subsystems: hardware and software (Qiu and Hsu 2001b). The former consists of the physical components such as AGVs, paths, controllers, sensors and guidance devices, etc. The latter embodies approaches or algorithms for systematically managing the hardware resources of an AGV system so that the whole system can work harmoniously with the highest efficiency.

However, as we have seen, in the past four decades, the progress of hardware of AGVs has far exceeded that of software. The lag of (controlling) software becomes an acute problem only in recent years when certain time-critical applications, e.g. container handling in seaports, require a great number of tasks to be completed in real time and hence involve a relatively large fleet (e.g. in the order of hundreds) of AGVs (Qiu and Hsu 2000a–c). Many hazards, such as congestion or deadlocks in AGV operations, arise from inadequate software of an AGV system. *Scheduling* and *routing* of AGVs, which seemed trivial in the earlier days when the number of vehicles was small, now becomes important and non-trivial.

It was said that one of the largest port operators has postponed its plan to deploy AGV systems in its new container port because no satisfactory solutions to the problem of scheduling and routing of AGVs have been found so far. Therefore, there are urgent requirements both in theory and in realistic applications to find solutions to the problem.

1.2. *Organization of the paper*

The remaining sections are arranged as follows. Section 2 gives the brief description for scheduling and routing of AGVs. Section 3 describes commonly encountered hazards arising in scheduling and routing of AGVs. We then differentiate the problem of AGV scheduling and routing from the conventional vehicle-routing problems, conventional path problems in graph theory and data routing, and argue why specialized approaches are needed for the problem. In section 4, the existing algorithms for the problem are surveyed and classified. Finally, section 5 gives concluding remarks and points out the directions for the future study of AGV scheduling and routing.

2. Description of the problem

The scheduling and routing of AGVs are actually two related aspects. We give the description of each aspect as follows.

2.1. *Scheduling*

The aim of AGV scheduling is to dispatch a set of AGVs to achieve the goals for a batch of pickup/drop-off (or P/D for short) jobs under certain constraints such as deadlines, priority, etc. The goals are normally related to the processing time or utilization of resources, such as minimizing the number of AGVs involved while maintaining the system throughput, or minimizing the total travel time of all vehicles (Akturk and Yilmaz 1996), and the likes.

2.2. *Routing*

Once the scheduling decision is made, the mission of routing is to find a suitable route (e.g. shortest-distance path, shortest-time path or minimal energy path)

(Daniels 1988) for every AGV from its origin to destination based on the current traffic situation.

The routing decision involves two issues. First, it should detect whether there *exists* a route that could lead a vehicle from its origin to destination. For instance, in the indirect transfer system (Bartholdi and Platzman 1989, Bozer and Park 1992, Bozer and Srinivasan 1991), if the destination is not in the same loop of the origin, there is no way for a vehicle to achieve its task without transferring its load to the other AGVs serving the other loops. Second, the route selected for the vehicle must be feasible, which means the route must be congestion-, conflict- and deadlock-free (Taghaboni and Tanchoco 1995), etc.

2.3. *Relations between scheduling and routing*

In some applications of AGV systems (Akturk and Yilmaz 1996, Bartholdi and Platzman 1989, Sinriech and Tanchoco 1994, Tanchoco and Sinriech 1992), only a few vehicles and jobs are involved. In this case, simple scheduling algorithms can serve the purpose. Jobs are usually handled in a first-come-first-serve (FCFS) fashion, and the nearest idle vehicle is usually chosen to serve a new job (De Guzman *et al*. 1997, Lin and Dgen 1994). Therefore, in this case scheduling seems very trivial and the research emphases are focused on routing.

In applications that involve many jobs, however, due to limitations of facility resources such as paths and loading/unloading buffers, a simple scheduling algorithm may not achieve a high system efficiency. For example, when we have only a limited number of vehicles but many jobs, jobs have to be divided into several batches based on related factors such as traffic capacity or waiting time, and accordingly AGVs have to be dispatched and routed concurrently for each of the batches of jobs (Qiu and Hsu 2000a–c, 2001a). In this case, the scheduling strategy must ensure that the conditions under which the given routing algorithm works are satisfied. For instance, no two vehicles are dispatched to a same destination in the same batch of jobs.

2.4. *Issues arising in the scheduling and routing of AGVs*

The following listed are the commonly encountered hazards when scheduling and routing AGVs: collisions, congestion, livelocks and deadlocks (Hsu and Huang 1994). Because of their negative consequences, all hazards must be eliminated during the operations of an AGV system.

3. Similar problems and the need of tailored approaches for AGV scheduling and routing

Intuitively, scheduling and routing of AGVs can be considered as a variation of the *vehicle routing problem* (VRP) (Bodin and Golden 1981, Bodin *et al.* 1983), which has been approached typically by using techniques such as linear programming (Dantzig 1963). However, there are significant distinctions between them, which motivates us to treat the problem separately.

- The path networks considered in VRP is usually of metropolitan scale. In this case, the length of a vehicle becomes insignificant when compared with the distance it travels. Hence, the vehicle can be considered as a moving point. However, in an AGV system, the distance between two depots is usually relatively short (in the order of tens or less of an AGV length), the segment of a path occupied by an AGV cannot be ignored.

- For VRP, the load capacity of a path is not a consideration; collisions or congestion (e.g. especially at a junction or intersection of paths) among vehicles are assumed to never occur. However, for an AGV system, limited by the path space, AGVs may congest or even collide with each other if the system is not well scheduled and routed.
- For VRP, the shortest distance path normally coincides with the shortest time path (but may not necessarily be the least cost path); while for an AGV system, due to the contention of the path, it is quite common that the shortest time path is not necessarily the shortest path.
- For VRP, the path network is predefined and unchangeable; whereas for an AGV system, sometimes the existing path layout can (or has to) be revised to meet a better scheduling or routing algorithm.

It should also be pointed out that even with many advanced technological features, the latest model of AGVs is still grossly inferior to human drivers in terms of sensory and decision-making capabilities (Hsu and Huang 1994, Huang and Hsu 1994). For example, a human driver can foresee the general 'trend' of other moving vehicles and avoid congestion, either by driving around an obstacle or by taking an alternative route. In contrast, what a typical AGV possesses is only rudimentary motion primitives and collision-prevention mechanisms. As a result, much of the responsibility that previously lies with the human drivers of ensuing congestion-free routing now totally falls on the algorithms of AGV scheduling and routing. For this reason, the problem is quite different from the conventional VRP. Therefore, appropriate and effective algorithms are always needed by an AGV system for scheduling and routing of vehicles.

The problem of scheduling and routing of AGVs is also different from the conventional path problems in graph theory such as the *shortest path problem, Hamiltonian-type problem* or *scheduling problem*. For example, a graph-theoretic problem usually concerns whether there *exists* an optimal path leading to a given destination node, while in an AGV system, *when* and *how* an AGV gets to its destination is also emphasized. In other words, scheduling and routing of AGVs is a time-critical problem, while a graph problem is usually not. Moreover, a good plan of scheduling and routing cannot disregard the system control mechanism and path layout; whereas a graph problem normally does not involve these two issues.

It is also quite natural to notice the similarities between the problem of scheduling and routing of AGVs and that of routing electronic data in a network. A possible analogy here is: AGVs are analogous to data packets, the paths (or tracks) to the data links, and the traffic control devices to the routers. Again, there are fundamental distinctions in the two problems such that one scheme in a system may not be applicable directly to the other (Hsu and Huang 1994, Huang and Hsu 1994). For example, the time for transferring electronic data packets is generally not taken as a function of distance between senders and receivers. However, in an AGV system, the time for transporting loads is usually a function of distance between origins and destinations. As another example, the sender can discard and then re-send a frame of electronic data when it fails due to the congestion in the data links. In contrast, the loads carried by AGVs neither have back-up copies nor can be discarded. These motivate us to consider the scheduling and routing of AGVs as a distinct problem by itself.

4. Taxonomy of the algorithms for the problem

According to the characteristics of the algorithms for the problem, the existing work could be classified into three general categories: (1) algorithms for general path topology, (2) path layout optimization and (3) algorithms for specific path topologies. It should be pointed out that because of the reasons stated in section 2, most of the algorithms reviewed adopt FCFS rules or other simple ones (Lin and Dgen 1994, Qiu and Hsu 2001a, Tanchoco and Sinriech 1992) for scheduling by default, i.e. they only solve just one half of the problem—routing. Works in the first category usually treat the problem as a graph theory problem, and use approaches such as Dijkstra's shortest path algorithm and *partitioning shortest path* (PSP) algorithm (Glover *et al.* 1985), etc., to get optimal routes. By using optimization techniques such as integer programming, works in the second category focus on the optimization of path network, in which even the routing control is generally very simple. For the third category, path networks are restricted to specific topologies such as single-loop, multi-loops or meshes, etc., and algorithms are developed to route and control AGVs in them. Moreover, there are some works dedicated to dispatching or scheduling of AGVs without consideration of routing (Akturk and Yilmaz 1996, Kim and Bae 1999, Klein and Kim 1996, Lee *et al.* 1996).

4.1. *Algorithms for general path topology*

Algorithms in this category focus mainly on the finding of the feasible routes for AGVs without considering the topological characteristics of path layout. In other words, these algorithms attempt to offer universal routing solutions (e.g. conflict-free shortest-time path).

As argued in section 3, the AGV routing problem is different from the conventional VRP. For VRP, vehicles will not run into collision because they are regarded as moving points when they run on the path network. However, the path (or lane) for AGVs usually has limited width which may not allow two or more AGVs to run side by side at a time. Therefore, a basic consideration is to give conflict-free and shortest-time routing solutions for AGVs. The papers reviewed in the following are representative works about this issue. The methods adopted may be classified into three categories: (1) static methods, where an entire path remains occupied until a vehicle completes the tour; (2) time-window-based methods, where a path segment may be used by different vehicles during different time-windows; and (3) dynamic methods, where the utilization of any segment of path is dynamically determined during routing rather than before routing as with cases (1) and (2).

4.1.1. *Static methods*

When the scale of an AGV system is small, it is quite natural to treat the problem with a static method (Broadbent *et al.* 1985, Daniels 1988, Egbelu 1987, Egbelu and Tanchoco 1986, Lim 1988, Lin 1986). The main advantage of the idea is its simplicity, while a disadvantage is its optimal solutions. We discuss how these algorithms work in the following.

The concept of conflict-free and shortest-time AGV routing was first presented by Broadbent *et al.* (1985). The routing procedure described employs Dijkstra's shortest path algorithm to generate a matrix, which describes the path occupation time of vehicles. Potential conflicts among vehicles, namely head-on, head-to-tail collisions and collisions at a path junction, are detected by comparing path occupation time. The potential conflicts are avoided *a priori* as follows: head-on conflicts are resolved by finding another shortest path excluding the congested segment; junction and head-to-tail conflicts are resolved by slowing down a new vehicle to let the previously scheduled ones proceed first. The time complexity of finding a path for a vehicle is $O(n^2)$, where n is the number of nodes (P/D stations or junctions of paths) of the path network.

Compared with unidirectional path AGV systems, there are obvious advantages of bidirectional path AGV systems in terms of utilization of vehicles and potential throughput efficiency. Egbelu and Tanchoco (1986) and Egbelu (1987) showed a marked improvement in productivity and a reduction in the number of vehicles required in bidirectional path AGV systems. The control of bidirectional path AGV systems can be complex because of the contention of multiple vehicles for the shared path segments. In this case, the routing algorithms must be able to eliminate potential head-on collisions among vehicles, and deadlocks as well. They also suggested the use of bidirectional path networks for routing AGVs (Egbelu and Tanchoco 1986). However, no algorithm is given to guarantee the optimal routes for vehicles in the paper.

Daniels (1988) first introduced an algorithm to route vehicles in a bidirectional flow path network, in which the PSP algorithm (Glover *et al.* 1985) is applied to find the shortest path for an AGV. The correctness and feasibility (in terms of time/space requirements) of the algorithm are theoretically proven. The algorithm can find a conflict-free and shortest-time route for a newly added AGV without changing the existing routes of the other vehicles. The computational complexity of finding a routing for every AGV is $O(n \times a)$, where n is the number of nodes and a is the number of arcs in the path network (arcs are the path segments and nodes are P/D stations or junctions of paths). However, when a path is allocated to a vehicle v, it is considered unusable by other newly added vehicles before v reaches its destination. In reality, the path segments of the route for v are only partially occupied by it during some time-windows. In other words, the algorithm does not allow vehicles to use path resource that could otherwise be shared during different time-windows. A newly added vehicle could run on the path segments within their free time-windows. Consequently, sometimes the algorithm may not find a path even if there exists one for a vehicle. Hence, the algorithm is only suitable for a system with a small path network and a small number of AGVs.

4.1.2. *Time-window-based methods*

In order to share the path network more efficiently, the time-window method is proposed and applied for routing of AGVs (Huang *et al.* 1989, Kim and Tanchoco 1991, 1993, Kolen *et al.* 1987, Rajotia *et al.* 1998b). The main contribution of time-window method is in the enhancement of the path utilization. In the following, we discuss some of the work.

Huang *et al.* (1989) proposed a labelling algorithm to find a shortest time path for routing a single vehicle in a bidirectional path network. A graph G is obtained from a given path network by representing each physical path segment as a node in G; two nodes in G are linked if and only if the corresponding path segments are adjacent to each other. By comparing the labels of every node, a shortest-time path could be obtained if it exists. The algorithm has the time complexity of $O(w^2 \log w)$ for a single vehicle, where w is the total number of time-windows of all nodes in the converted network. The main disadvantage of the algorithm is the unacceptably large amount of computation. The converted network has double number of arcs and at least double number of nodes more than that of the original path network, and every node has at least one time-window. (In a connected graph, the node number n and the arc number a satisfy $a \geqslant n - 1$.) Therefore, the data structures could require a lot of memory space. The actual time complexity for a single vehicle is $O((n + a)^2 \log(n + a))$, where n and a are the numbers of nodes and arcs in the original path network respectively.

Kim and Tanchoco (1991) also presented a conflict-free and shortest time algorithm for routing AGVs in a bidirectional path network. Their algorithm is based on Dijkstra's shortest path algorithm. It maintains, for each node, a list of time-windows reserved by scheduled vehicles and a list of free time-windows available for vehicles to be scheduled. The concept of time-window graph is applied, in which the node set represents the free time-windows and the arc set represents the reachability among free time-windows. Then the algorithm routes vehicles through the free time-windows of the time-window graph instead of the physical nodes of the path network. To get an optimal routing solution, the algorithm could take a large amount of time. Specifically, it requires $O(v^4 n^2)$ computation in the worst case, where v is the number of vehicles and n is the number of nodes. Therefore, it will be more suitable for a small system with few vehicles.

Kim and Tanchoco (1993) later gave the operational control of bidirectional path AGV systems for the conflict-free and shortest-time routing algorithm. The research employs a *conservative myopic* strategy, in which only one vehicle is considered at a time and all the previous decisions are strictly respected and a subsequent travel schedule is assigned only after the vehicle becomes idle. Using this strategy, to find a path from the current position of a vehicle to a pickup station and then to the drop-off station, the system controller need only call at most twice the conflict-free and shortest-time routing algorithm given in Kim and Tanchoco (1991) after the decision on vehicle dispatching is made.

4.1.3. *Dynamic methods*

The preceding algorithms rely on global information and make routing decision for each vehicle before the actual routing. They generally take a relatively long time of computation for an optimal route. In order to speed up the process of finding the routes for AGVs, dynamic routing techniques are proposed (Langevin *et al.* 1994, Taghaboni and Tanchoco 1995).

Taghaboni and Tanchoco (1995) proposed a dynamic routing technique, namely *incremental route planning*, which can route AGVs relatively quickly compared with some static algorithms. The algorithm selects the next node for the vehicle to visit so as to reach its destination based on the status of neighboring nodes and information about the global network. The next node is selected among all adjacent nodes such that it will result in the shortest travel time. The next node selection is repeated until the vehicle reaches its destination. The algorithm can work in both unidirectional and bidirectional path networks. However, compared with the *complete route planner* mentioned here, the incremental route planning can not achieve a high efficiency when the number of tasks and vehicles increases. The algorithm may also not obtain an optimal route and correctly predict the delays in some cases.

Based on dynamic programming, Langevin *et al.* (1996) presented an algorithm that could give an optimal integrated solution for planning the dispatching, conflict-free routing and scheduling of AGVs in a flexible manufacturing system. The algorithm defines a partial transportation plan as a schedule and a route for each vehicle satisfying a subset of the transportation tasks. States are defined corresponding to the partial transportation plans. The dynamic programming operations work on the states to find the best final state set which contains the optimal solutions. However, as stated in the paper, the number of states could be astronomical for a large system. To make it usable, a procedure is designed to eliminate some of the states. Even so, the computation can be very expensive. The fatal limitation is that the system considered contains only two vehicles, although the path network may be bidirectional. Therefore, the throughput of such a system may not scale up with the number of tasks. An extended version of the algorithm is also given in the study to deal with systems containing more than two vehicles. However, it still cannot guarantee the optimality of the solutions.

4.2. *Path optimization*

Owing to the heavy computation for the finding of the optimal routes in a general path network, many researchers propose the idea of the optimization of path layout (Gaskins and Tanchoco 1987, Goetz and Egbelu 1990, Kaspi and Tanchoco 1990, Kouvelis *et al.* 1992, Kouvelis and Kim 1992, Langevin *et al.* 1994, Rajotia *et al.* 1998a, Sinriech and Tanchoco 1991) or the distribution of P/D stations (Banerjee and Zhou 1995, Bodin *et al.* 1983, Kiran and Tansel 1989, Kiran *et al.* 1992). The optimization problem is usually formulated in integer programming models (Gaskins and Tanchoco 1987, Gaskins *et al.* 1989, Kaspi and Tanchoco 1990, Sinriech and Tanchoco 1991) as discussed in the following.

4.2.1. *0–1 Integer-programming model*

Gaskins and Tanchoco (1987) first formulated the path layout problem as a 0–1 integer programming model with considerations of the given facility layout and P/D stations. The objective is to find an optimal path network which will minimize the total travelling distance of loaded vehicles. However, the paper only considers unidirectional path network, which has lower utilization than bidirectional ones do (Egbelu and Tanchoco 1986). The distance travelled by unloaded vehicles is not taken into consideration, which may affect the routing control and system throughput. The main limitation of the study is that it only considers a fleet of AGVs with the same origin and destination every time. These AGVs run along the same route. Therefore, in this case routing control is trivialized since issues such as congestion, deadlocks and conflicts will never occur. Because routing control is not taken into account, when AGVs with different origins and destinations run simultaneously, conflicts may occur among them. For the formulated 0–1 integer programming model here, another limitation is that it has a low probability of obtaining a non-empty solution set. Moreover, for practical problems, the number of 0–1 variables needed for the model tends to be very large and computational efficiency becomes a critical issue.

Based on a 0–1 integer-programming model and the branch-and-bound method, Kaspi and Tanchoco (1990) proposed an alternative formulation of the AGV path layout problem described in Gaskins and Tanchoco (1987). The new approach can obtain the best path design provided that the locations of P/D stations and facility layout are given. The procedure, compared with the algorithm in (Kaspi and Tanchoco 1990), reduces the computation time at the cost of the quality of path design because not all of the possibilities are enumerated. However, the other drawbacks in the previous paper are still not fully overcome here.

4.2.2. *Intersection graph method*

Sinriech and Tanchoco (1991) presented an *intersection graph method* (IGM) for solving AGV flow path optimization model by (Kaspi and Tanchoco 1990). A procedure based on the technique of branch-and-bound is described wherein only a reduced subset of all nodes in the path network is considered, and only intersection nodes are used to find optimal solutions. With this improved procedure, the number of branches of the main problem is only half of that of the method described (Kaspi and Tanchoco 1990). The amount of computation is thus greatly reduced. Therefore, this approach could be adopted by a relatively large AGV system. However, because only the intersection nodes of the path network are used, the algorithm might miss some optimal solutions.

4.2.3. *Integer linear programming model*

The same problem formulated by Gaskins and Tanchoco (1987), Kaspi and Tanchoco (1990) and Sinriech and Tanchoco (1991) is also studied by Goetz and Egbelu (1990) with a different approach. They modelled and solved the problem of selecting the path as well as the location of P/D stations as an integer linear-programming problem. The objective is to minimize the total distance travelled by both loaded and unloaded AGVs. A heuristic algorithm is used in the study to reduce the size of the problem. The reduction makes the approach more amenable for use in the design of large path layouts. Then the problem of determining the optimal locations of P/D stations is formulated based on the reduced problem. The paper also exploits the structure of the problem to reduce the number of constraints required to solve the path layout problem. However, the issues of vehicle number that the system could have and the routing control on the optimized path network are not considered in the study, which are obviously related to the final goal of the system optimization. The path studied here is uni-directional, which, to some extent, results in low path utilization and system throughput (Egbelu and Tanchoco 1986).

4.3. *Algorithms for specific path topologies*

In realistic applications, path topologies are usually specific and regular. The commonly encountered path layouts are linear, loop/loops, mesh, etc. Algorithms for these specific path topologies usually achieve better effects than those for general path topology.

4.3.1. *Linear topology*

Linear path topology is a basic type of path layouts. Qiu and Hsu (2001a) presented a scheme to schedule and route a batch of AGVs concurrently on a bidirectional linear path layout. The proposed algorithms actually employ the idea of concurrent processing. Freedom of conflicts is provably guaranteed when AGVs are scheduled and routed to run along the linear tracks. All jobs can be completed within a very short time. The efficiency of the algorithms is not constrained by the size of the system. In the study, the scheduling scheme by default is to handle the jobs batch by batch. Therefore, to decrease the idle runs of vehicles, they also point out a promising direction for future study—how to schedule continuously and route vehicles to handle jobs. One of the shortcomings of their current scheme is the stringent (and unrealistic) synchronization requirements of vehicles, which should be relaxed to fill the needs of realistic applications.

4.3.2. *Loop topology*

The loop topology, including single-loops, multi-loops, SFT (segmented floor topology), etc., is a commonly adopted path layout for an AGV system (Banerjee and Zhou 1995, Barad and Sinriech 1998, Bartholdi and Platzman 1989, Bozer and Srinivasan 1991, De Guzman *et al.* 1997, Lin and Dgen 1994, Sinriech and Tanchoco 1991, 1992, 1993, 1994, 1997, Tanchoco *et al.* 1987, Tanchoco and Sinriech 1992). With a loop, usually only few vehicles run in the same direction within the loop; the routing control is very simple, but the system throughput may not be very high. In the following we discuss algorithms for this topology.

Tanchoco and Sinriech (1992) suggested an optimal closed single-loop path layout for an AGV system. An algorithm based on integer programming is given to find the optimal single-loop. With the single-loop path layout, the routing algorithm could be very straightforward. In this case, if all vehicles run in the same direction with uniform speed, no collisions may occur among them because there are no intersections in the optimal singe-loop path. As the paper claims, however, the system throughput drops down a little compared with conventional path systems. It allows at most ten vehicles to run simultaneously around the single-loop (Tanchoco and Sinriech 1992). Therefore, such an AGV system may not be very suitable for a large material handling system with a great number of vehicles and stations.

Lin and Dgen (1994) provided an algorithm for routing AGVs among non-overlapping closed loops within a tandem AGV system. P/D stations within each loop are served by a single dedicated vehicle. The transit areas located between two adjacent loops serve as an interface and allow loads to be transferred from one loop to another. If a load needs to be delivered to a station not located within the same loop, the load needs more than one vehicle to carry it to its destination. A task-list time-window algorithm is employed to find a shortest travel time path based on the current status to route a vehicle from one point to another. The computation for a routing decision is relatively small. However, since only one vehicle serves a loop, the throughput of the system is very low. Besides, the devices used for transferring loads from one loop to another usually increase the system cost. Therefore, the scale of such a system could not be very large. The paper did not address the design issue of the loops, which is crucially related to the system efficiency.

Similar to the multi-loop path layout in Lin and Dgen (1994), an alternative path topology—SFT (Barad and Sinriech 1998, Sinriech and Tanchoco 1991, 1992, 1994, 1997) is proposed and studied. The SFT can be used in conjunction with each of the following three network types: connected, partitioned and split-flow. The general SFT is comprised of one or more zones, each of which is separated into non-overlapping segments with each segment served by a single vehicle. Transfer buffers are located at both ends of every segment. Since there is only one vehicle moving in a segment, no path segment contentions will occur. The vehicle can move bidirectionally in the segment. Therefore, the routing control for such a path topology is very simple. The advantage of the SFT can be easily recognized by the lower value of flow $\times$ distance compared with that of other path topologies, such as single-loop, bidirectional and uni-directional conventional paths, etc. However, the transferring devices in the buffers are the additional cost of the overall system. The operations of transferring loads among segments may be time-consuming and sometimes cause delay of transportation.

4.3.3. *Mesh topology*

In recent AGV applications for container shipping and transportation at container terminals (e.g. (Evers and Koppers 1996, Qiu and Hsu 2000a, Ye *et al.* 2000)), the container stacking yards are usually arranged into rectangular blocks, which leads to a mesh-like path topology. Therefore, developing efficient algorithms for this topology becomes a focus.

Hsu and Huang (1994) and Huang and Hsu (1994) gave analysis of time and space complexities for some basic AGV routing operations in several specific bi-directional path topologies. The routing operations include simple delivery, distribution, scattering, accumulation, gathering, sorting and total exchange (or shuffling). All these operations are achieved in the following path topologies: linear array, ring, binary-tree and H-tree, star, 2D mesh, n-cube and cube-connected cycles, and complete graph. The upper bounds of time and space complexities for AGV routing in those path topologies are $\Theta(n^2)$ and $\Theta(n^3)$ respectively, where n is the number of nodes in the path topologies. However, the paper does not give the details of the routing algorithms and the techniques to avoid congestion, conflicts, deadlocks, etc. It also causes additional system cost if every node in the path network has an arbitration capability.

Qiu and Hsu (2000a–c) presented different methods to schedule and route AGVs in an $n \times n$ mesh-like path topology. The algorithms can schedule and route simultaneously up to $4n^2$ AGVs concurrently at one time. The routing process is fixed, which is either an adapted *bitonic* merging procedure (Qiu and Hsu 2000b,c) or a column-row-column permutation procedure (Qiu and Hsu 2000a); while the scheduling process schedules AGVs batch by batch based on the job arrivals. Qiu and Hsu (2000a) give an improved version of the solution. When running the algorithms, all these $4n^2$ vehicles can get to their destinations within $O(n \log n)$ rectilinear steps (defined as the distance between two neighboring junctions of paths in the mesh) (Qiu and Hsu 2000b,c) which was further decreased down to $3n$ steps (Qiu and Hsu 2000a) of well-defined physical moves. In all algorithms mentioned above, freedom of conflicts among AGVs is provably guaranteed. Although the size of the mesh does not affect the efficiency of the algorithms, when the number of AGVs is far less than $4n^2$, the path that an AGV travels might not be an optimal one. This is because in this case the AGVs are quite sparse compared with the mesh so that they can even go directly by the shortest paths towards their destinations without conflicts.

4.4. *Dedicated scheduling algorithms*

Most studies reviewed above pay more attention to routing but adopt simple scheduling rules implicitly, however, there are some work dedicating to scheduling of AGVs and jobs without considering the routing process (Akturk and Yilmaz 1996, Kim and Bae 1999, Klein and Kim 1996, Lee *et al.* 1996).

A typical work on scheduling of AGVs is by Akturk and Yilmaz (1996). They proposed an algorithm to schedule vehicles and jobs in a decision-making hierarchy based on the mixed-integer programming. The proposed *micro-opportunistic scheduling algorithm* (MOSA) combines two perspectives, namely job- and vehicle-based approaches, into a single algorithm, in which both the critical jobs and the travel time of unloaded vehicles are considered simultaneously. The scheduling problem is defined similarly to, but not identically as, the *time constrained vehicle routing problem* (TCVRP), which is proven to be NP-hard by (Kolen *et al.* 1987). The problem is solvable in polynomial time only because the objective is to minimize the deviation of the time-windows rather than the distance traveled by vehicles in TCVRP. However, the MOSA is only applicable for AGV systems with a small number of jobs and vehicles. This is because with the increase of job number and the size of vehicle fleet, the amount of computation for the solution could become unacceptably large.

Kim and Bae (1999) presented a model for scheduling of AGVs for multiple container-cranes. The primary objective is to minimize the delay of carrying out all loading and unloading operations. In other words, the goal is to minimize the waiting time of container-cranes rather than to minimize the idle running of AGVs. Issues related to AGV routing are not taken into considerations. Therefore, with the increase of the number of AGVs, congestion or even collisions of AGVs might occur at the operating area of container-cranes.

5. Future directions and concluding remarks

This paper has reviewed the existing representative results on the problem of routing and scheduling of AGVs. The algorithms are classified into three categories: (1) algorithms for general path topology, (2) optimization of path network and (3) algorithms for specific path topologies. Most of the existing results emphasize more on routing than on scheduling because usually simple scheduling strategies such as FCFS (Lin *et al.* 1994, Lin and Dgen 1994, Sinriech and Tanchoco 1991, 1992) or batch by batch (Qiu and Hsu 2000a–c, 2001a) are adopted. There are also algorithms dedicated to scheduling of AGVs without consideration of routing (Akturk and Yilmaz 1996, Kim and Bae 1999, Klein and Kim 1996, Lee *et al.* 1996).

5.1. *Appraisal of previous work*

Algorithms for general path topology theoretically prove the feasibility of finding the conflict-free and shortest-time routes for AGVs on both uni-directional and bidirectional path networks (Broadbent *et al.* 1985, Daniels 1988, Kim 1998, Lin 1986). However, because these algorithms treat the routing problem as a shortest path problem in graph theory, to find a route for a vehicle, the algorithms usually have to search every node and arc of the path network graph. Especially when the time-window method is applied, the computation needed for finding a solution increases quickly (Huang *et al.* 1989, Kim and Tanchoco 1991, 1993). As reviewed in the preceding sections, the least upper bound of these algorithms achieved is $O(n^2)$, where n denotes the number of nodes of the path network graph. Some of the algorithms may also miss the optimal solutions (Egbelu and Tanchoco 1986, Langevin *et al.* 1994, Taghaboni and Tanchoco 1995) because certain constraints are overly enforced. For example, the algorithm in Daniels (1988) treats the whole path being occupied by a vehicle. Before the previous one gets to its destination, a newly added vehicle is not allowed to use any segments of the path used by the previous vehicle. This may cause the failure of finding a feasible route for a new vehicle or unnecessary delays. These algorithms are only suitable for AGV systems with a small number of vehicles and small size of path networks.

Research on optimization of path network has so far focused on optimizing the configuration and arrangement of the existing facilities, such as P/D stations, path segments, etc. Usually the problem is formulated as an integer-programming problem, which normally takes a lot of computations to search for the optimal solutions (Gaskins and Tanchoco 1987, Goetz and Egbelu 1980, Kaspi and Tanchoco 1990). Routing control in this case is regarded as trivial. The number of vehicles considered is small (in the order of ten at most). Therefore, such systems cannot achieve a high throughput.

The algorithms for specific path topologies, on the other hand, give solutions for scheduling and routing of AGVs on certain specific path topologies such as closed single-loop, multi-loops, segmented path and mesh, etc. Conflicts and deadlocks are easily eliminated in some cases, which eventually simplify the routing control. Although most of these algorithms proposed for the loop path topology do not allow many vehicles to be routed simultaneously, the approaches are very novel and innovative which can be a point of reference when designing new routing algorithms.

5.2. *Suggestions for future research*

For future research, we suggest that the most fertile area should be in the development of new scheduling and routing algorithms for specific path topologies.

- In many applications, the AGV path networks are regular graphs, such as linear array, loop/loops, 2D mesh. For example, in a container-handling application, the container-stacking yards are usually arranged as rectangular blocks connected with mesh-like paths.
- Algorithms for specific path topologies usually have relatively lower computational complexity compared with those for general path topology.
- Developing algorithms for specific path topologies is relatively more feasible than those for general path topology. Meanwhile, because we can exploit the characteristics of these specific topologies, we could develop algorithms that are more efficient.

Algorithms with provable qualities such as freedom of conflicts among vehicles (e.g. Qiu and Hsu 2000a–c) also provide a direction for more research.

5.3. *Concluding remarks*

In spite of the pervasive applications of AGV systems, scheduling and routing of AGVs still calls for much attention. In the latest conferences on robotics and intelligent vehicles (Proceedings 1999, 2001) much study still focus on difficult issues such as automated driving of vehicles, intelligentization of vehicles, intelligent navigation mechanisms, robot vision, image processing, information fusion, etc; relatively fewer papers refer to scheduling and routing of AGVs. It should be pointed out that even if issues on vehicle automation or intelligentization are fully resolved, the problems of scheduling and routing will not disappear accordingly. Even if all vehicles are driven by human drivers, how can anyone guarantee that congestion or deadlocks will not occur without proper scheduling and routing? On the other hand, good scheduling and routing algorithms will enable us to apply the AGV systems using the current level of AGV technologies, without waiting for solutions of the difficult problems mentioned above.

As a case in point, presently at Nanyang Technological University, Singapore, there is an ongoing project on the application of AGVs to container handling. The main goal is to schedule and route AGVs within a mesh-like path topology of a container terminal. A satisfactory solution to the problem should use the currently available vehicle technology to decrease the system costs.

The AGV systems are intrinsically parallel and distributed systems. We feel that the routing and scheduling of these systems should be of interests to computer scientists/engineers, in addition to the mechanical or industrial engineers/researchers. This is still a fertile area where all can join to make contributions.

References

Akturk, M. S. and Yilmaz, H., 1996, Scheduling of automated guided vehicles in a decision-making hierarchy. *International Journal of Production Research,* **32**, 577–591.

Banerjee, P. and Zhou, Y., 1995, Facilities layout design optimization with single loop material flow path configuration. *International Journal of Production Research,* **32**, 183–204.

Barad, M. and Sinriech, D., 1988, A Petri net model for the operational design and analysis of segmented flow topology (SFT) AGV system. *International Journal of Production Research,* **36**, 1401–1426.

Bartholdi, J. J. and Platzman, L. K., 1989, Decentralized control of automated guided vehicles on a simple loop. *IIE Transactions,* **21,** 76–81.

Bodin, L. D. and Golden, B. L., 1981, Classification in vehicle routing and scheduling. *Network,* **11**(2), 97–108.

Bodin, L. D., Golden, B. L., Assad, A. and Ball, M., 1983, Routing and scheduling of vehicles and crews: the state of the art. *Computers and Operation Research,* **10**, 63–211.

Bozer, Y. A. and Park, J. H., 1992, New partitioning schemes for tandem AGV systems. In *Proceedings of the 1992 International Material Handling Colloquium*, Milwaukee, WI.

Bozer, Y. A. and Srinivasan, 1989, Tandem configuration for AGV systems offer simplicity and flexibility. *IE Magazine*, February, 23–27.

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

Broadbent, A. J., Besant, C. B., Premi, S. K. and Walker, S. P., 1985, Free ranging AGV systems: promises, problems and pathways. In *Proceedings of the 2nd International Conference on Automated Materials handling* (IFS/Springer), pp. 221–237.

Daniels, S. C., 1988, Real-time conflict resolution in automated guided vehicle scheduling. PhD thesis, Department of Industrial Engineering, Pennsylvania State University, USA.

Dantzig, G., 1963, *Linear Programming and Extractions* (Princeton: Princeton University Press).

De Guzman, M. C., Prabhu, N. and Tanchoco, J. M. A., 1997, Complexity of the AGV shortest path and single-loop guide path layout problems. *International Journal of Production Research,* **35**, 2083–2092.

Egbelu, P. J., 1987, The use of non-simulation approaches in estimation vehicle requirements in an automated guided vehicle based transport system. *Material Flow,* **4**, 17–32.

Egbelu, P. J. and Tanchoco, J. M. A., 1986, Potentials for bidirectional guidepath for automatic guided vehicle systems. *International Journal of Production Research,* **24**, 1075–1097.

Evers, J. J. M. and Koppers, S. A. J., 1996, Automatic guided vehicle traffic control at a container terminal. *Transportation Research Part A*, **30**, 21–34.

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

Gaskins, R. J., Tanchoco, J. M. A. and Taghaboni, F., 1989, Virtual flowpaths for free-ranging automated guided vehicle systems. *International Journal of Production Research,* **27**, 91–100.

Glover, F., Klingman, D. D. and Phillips, N. V., 1985, A new polynomially bounded shortest path algorithm. *Operations Research*, **33**(1), 65–73.

Goetz, W. G. and Egbelu, P. J., 1990, Guide path design and location of load pick-up/drop-off points for an automated guided vehicle system. *International Journal of Production Research*, **28**, 927–841.

Hsu, W.-J. and Huang, S.-Y., 1994, Route planning of automated guided vehicles. In *Proceedings of Intelligent Vehicles,* Paris, pp. 479–485.

Huang, S.-Y. and Hsu, W.-J., 1994, Routing automated guided vehicles on mesh-like topologies. In *Proceedings of International Conference on Automation, Robotics and Computer Vision*.

Huang, J., Palekar, U. S. and Kapoor, S. G., 1989, A labeling algorithm for the navigation of automation guided vehicles. In *Advances in Manufacturing Systems Engineering, Proceedings of the ASME Winter Annual Meeting*, San Francisco, CA, PED vol. 37, pp. 181–193.

Kaspi, M. and Tanchoco, J. M. A., 1990, Optimal flow path design of uni-directional AGV systems. *International Journal of Production Research,* **28**, 915–926.

Kim, C. W. and Tanchoco, J. M. A., 1991, Conflict-free shortest-time bi-directional AGV routing. *International Journal of Production Research,* **29**, 2377–2391.

Kim, C. W. and Tanchoco, J. M. A., 1993, Operational control of a bi-directional automated guided vehicle systems. *International Journal of Production Research,* **31**, 2123–2138.

Kim, K. H. and Bae, J. W., 1999, Dispatching automated guided vehicles for multiple container-cranes. In Seminar on Port Design and Operations Technology, Pan Pacific Hotel, Singapore, 4–5 November.

Kiran, A. S. and Tansel, B. C., 1989, Optimal pickup point location on material handling networks. *International Journal of Production Research,* **27**, 1475–1486.

Kiran, A. S., Unal, A. T. and Kirabati, S., 1992, A location problem on a unicyclic network: balanced case. *European Journal of Operations Research,* **62**, 144–202.

Klein, C. M. and Kim, J., 1996, AGV dispatching. *International Journal of Production Research,* **34**, 95–110.

Kolen, A. W. J., Rinnooy-Kan, A. H. G. and Trienekens, H. W. J. M., 1987, Vehicle routing with time windows. *Operations Research,* **35**(2), 266–274.

Kouvelis, P. and Kim, M., 1992, Uni-directional loop network problem in automated manufacturing systems. *Operations Research,* **40**(3), 544–550.

Kouvelis, P., Gutierrez, G. J. and Chiang, W., 1992, Heuristic uni-directional flow path design approaches for automated guided vehicle systems. *International Journal of Production Research,* **30**, 1327–1351.

Langevin, A., Lauzon, D. and Riopel, D., 1996, Dispatching, routing and scheduling of two automated guided vehicles in a flexible manufacturing system. *International Journal of Flexible Manufacturing Systems,* **8**, 246–262.

Langevin, A., Montreuil, B. and Riopel, D., 1994, Spine layout design. *International Journal of Production Research,* **32**, 429–442.

Lee, J., Tangjarukij, M. and Zhu, Z., 1996, Load selection of automated guided vehicles in flexible manufacturing systems. *International Journal of Production Research,* **34**, 3383–3400.

Lim, K. K., 1988, Control algorithms for unit-load automated guided vehicles. PhD thesis, School of Industrial and System Engineering, Georgia Institute of Technology.

Lin, J. T., 1986, Development of a graphic simulation model for design of automated guided vehicles systems. PhD thesis, Department of Industrial Engineering, Lehigh University.

Lin, J. T., Chang, C. C. K. and Liu, W.-C., 1994, A load-routing problem in a tandem configuration AGVS. *International Journal of Production Research,* **32**, 411–427.

Lin, J. T. and Dgen, P.-K., 1994, An algorithm for routing control of a tandem automated guided vehicle system. *International Journal of Production Research,* **32**, 2735–2750.

Proceedings of the 4th Asian Conference on Robotics and its Application (ACRA 2001), National University of Singapore, Singapore, 6–8 June 2001.

Proceedings of the International Conference on Field and Service Robotics, Pittsburgh, PA, 29–31 August 1999.

Qiu, L. and Hsu, W.-J., 2000a, An algorithm for concurrent routing of AGVs in a mesh. In *Proceedings of the 7th Australasian Conference on Parallel and Real-Time Systems (PART 2000),* University of New South Wales, Sydney, Australia, 29–30 November, pp. 202–214.

Qiu, L. and Hsu, W.-J., 2000b, Routing AGVs by sorting. In *Proceedings of the 2000 International Conference on Parallel and Distributed Processing Techniques and*

Applications (PDPTA 2000), Las Vegas, NV, USA, 26–29 June 2000, Vol. 3, pp. 1465–1470.

QIU, L. and HSU, W.-J., 2000c, Routing AGVs on a mesh-like path topology. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (IV 2000),* Dearborn, MI, 3–5 October, pp. 392–397.

QIU, L. and HSU, W.-J., 20001a, A bi-directional path layout for conflict-free routing of AGVs. *International Journal of Production Research,* **39**, 2177–2195.

QIU, L. and HSU, W.-J., 20001b, Scheduling and routing algorithms for AGVs: a survey from a computer science perspective. In *Proceedings of the 5th International Conference on Mechatronics Technology (ICMT 2001)*, National University of Singapore, 6–8 June, pp. 112–117.

RAJOTIA, S., SHANKER, K. and BATRA, J. L., 1998a, A heuristic for configuring a mixed uni-/bi-directional flow path for an AGV system. *International Journal of Production Research,* **36**, 1779–1800.

RAJOTIA, S., SHANKER, K. and BATRA, J. L., 1998b, A semi-dynamic time window constrained routing strategy in an AGV system. *International Journal of Production Research,* **36**, 35–50.

SINRIECH, D. and TANCHOCO, J. M. A. 1991, Intersection graph method for AGV flow path design. *International Journal of Production Research,* **29**, 1725–1732.

SINRIECH, D. and TANCHOCO, J. M. A. 1992, An economic model for determining AGV fleet size. *International Journal of Production Research,* **30**, 1255–1268.

SINRIECH, D. and TANCHOCO, J. M. A, 1993, Solution methods for the mathematical models and single-loop AGV systems. *International Journal of Production Research,* **31**, 705–725.

SINRIECH, D. and TANCHOCO, J. M. A. 1994, SFT—segmented flow topology. In J. M. A. Tanchoco (ed.), *Material Flow Systems in Manufacturing* (London: Chapman & Hall), Chap. 8, pp. 200–235.

SINRIECH, D. and TANCHOCO, J. M. A., 1997, Design procedures and implementation of the segmented flow topology (SFT) for discrete material flow systems. *IIE Transactions,* **29**, 323–335.

TAGHABONI, F. and TANCHOCO, J. M. A., 1995, Comparison of dynamic routing techniques for automated guided vehicle systems. *International Journal of Production Research*, **33**, 2653–2669.

TANCHOCO, J. M. A. EGBELU, P. J. and TAGHABONI, F., 1987, Determination of the total number of vehicles in an AGV-based material transport system. *Material Flow*, **4**, 33–51.

TANCHOCO, J. M. A. and SINRIECH, D., 1992, OSL–optimal single-loop guided paths for AGVS. *International Journal of Production Research*, **30**, 665–681.

YE, R., VEE, V.-Y., HSU, W.-J. and SHAH, S. N., 2000, Parallel simulation of AGVs in container port operations. In *Proceedings of 4th International Conference/Exhibition on High Performance Computing in Asia-Pacific Region (HPC-ASIA 2000),* Beijing, 14–17 May, Vol. 1, pp. 1058–1063.