Computers & Industrial Engineering 56 (2009) 417–432

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Zone design and control for vehicle collision prevention and load balancing in a zone control AGV system

Ying-Chin Ho *, Ta-Wei Liao

Institute of Industrial Management, National Central University, Chung-Li, Taoyuan 32001, Taiwan

ARTICLE INFO

Article history:
Received 12 August 2007
Received in revised form 24 January 2008
Accepted 6 July 2008
Available online 18 July 2008

Keywords:
Zone partition design
Zone control AGV
Vehicle collision prevention
Load balancing

ABSTRACT

Because of their routing flexibility, automated guided vehicles (AGVs) have been used in many manufacturing systems, especially those in which parts with diverse and complex processing routes are made. In recent years, there have been many studies on AGV-related problems. One of them is preventing the collision of vehicles. Some traditional vehicle collision prevention strategies are zone strategies that divide guide paths into several non-overlapping zones and restrict the presence of only one vehicle in any zone at any time. Traditional zone strategies are fixed zone strategies in which the area of a zone is fixed and vehicles are not allowed to help each other. Because of these restrictions, fixed zone strategies cannot always satisfy the system's transportation demand if there is a load imbalance between vehicles of different zones. In this paper, a new zone strategy is proposed for a zone control AGV system with a network guide path. The proposed strategy is a dynamic zone strategy, which is different from a fixed zone strategy. It relies on two methods – zone partition design and dynamic zone control – to prevent the collision of vehicles and to maintain the load balance between the vehicles of different zones. The zone partition design defines a relationship coefficient, which measures both the distance relationship and the flow relationship between workstations, and uses it to find an initial zone partition design. We then improve this initial design by an SA (Simulated Annealing) based improvement procedure to achieve a better load balance result. The dynamic zone control uses two methods – zone repartition and load sharing – to ensure that vehicle collision can be prevented and the system's load balance can be maintained when the system is in operation. Simulation experiments were conducted to understand the performance of the proposed strategy. The simulation results show that the proposed strategy outperform the fixed zone strategy in throughput, WIP inventory, and flow time. The results also show that the proposed strategy is able to adapt to any changes (in the system) that cause the load imbalance problem between zones.

1. Introduction

Automated guided vehicles (AGVs) are known for their routing flexibility advantage. Despite of this advantage, AGVs have the disadvantage of being more difficult to control. Many issues need to be resolved in AGV control, one of which is vehicle collision prevention. Many vehicle collision prevention methods have been developed in the past and can be classified into three categories. Methods in the first category rely on the design and layout of guide paths and material handling systems to avoid collisions. For example, spurs or side-tracks are often employed to let vehicles pass other vehicles without the risk of colliding (Koff, 1985). Sharp and Liu (1990) present an analytical method for configuring the network of a fixed-path, closed-loop material handling system. Its purpose is to make good initial decisions with respect to adding short cuts, adding off-line spurs, and their length. Aiello, Enea, and Galante (2002) solve the problem of simultaneously determining the location of departments in a facility and designing the material handling system. One can also find many studies, e.g. Gaskins and Tanchoco (1987); Kaspi and Tanchoco (1990) and Kim and Tanchoco (1993a, 1993b); Rajotia et al. (1998) and Kaspi et al. (2002), that focus on AGV guide path layout designs. Some of these studies are for unidirectional guide paths, some bi-directional, and some mixed uni-/bi-directional. The paper by Vis (2006) gives a detailed survey on studies focused on AGV guide path layout problems.

Methods in the second category can be further divided into two subcategories. Methods in the first subcategory assume that all transport requests are known ahead and predetermine the routes of vehicles to prevent their collisions. Some of these methods include those proposed by Maxwell and Muckstadt (1982); Kusiak and Cyrus (1985) and Chen et al. (1987). One limitation of these methods is that solutions found by them cannot be implemented in real time. Methods in the second subcategory are able to control vehicles in real time. For example, Egbelu and Tanchoco (1986)

* Corresponding author. Tel.: +886 3 422 7151; fax: +886 3 425 8197.
E-mail address: ho@cc.ncu.edu.tw (Y.-C. Ho).

0360-8352/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2008.07.007

propose a method to control AGVs in a bi-directional network. Taghaboni (1989) proposes a node-to-node routing method that moves vehicles one step at a time. Krishnamurthy (1990) combines the problem of assigning vehicles to requesting points and the problem of generating conflict-free routes for vehicles. Kim and Tanchoco (1991) present an efficient algorithm for finding conflict-free shortest-time routes for vehicles in a bi-directional AGV system. Another study by Kim and Tanchoco (1993) describes a strategy to coordinate the movement of vehicles in a bi-directional AGV system. Reveliotis (2000) uses zone control to avoid collisions and deadlocks in AGVs with bidirectional guide-paths. Jang, Suh, and Ferreira (2001) develop a look-ahead AGV control procedure for AGV and part routing in semiconductor and LCD production bays by using the information of the future state of clean bays. Liu and Hung (2001) propose a real-time deadlock-free control strategy for single multi-load automated guided vehicle in a job shop system. Qiu and Hsu (2001) propose an algorithm for scheduling and routing AGVs on a bi-directional path layout. Fanti (2002) develops a real-time control strategy to avoid deadlocks, collisions and situations known as *restricted deadlocks* in a zone-control AGV system. Singh and Tiwari (2002) propose an intelligent agent framework to find a conflict-free shortest-time path for AGVs in a bi- or unidirectional guide path network. Dotoli and Fanti (2004) propose colored Petri nets to model the dynamics of AGVs and implement the control strategy stemming from the knowledge of the system state. Yoo, Sim, Cao, and Park (2005) propose a simple and easily adaptable deadlock avoidance algorithm for AGV systems. Their algorithm uses the graph-theoretic approach. Sarker and Gurav (2005) present a bi-directional path flow layout and a routing algorithm that guarantee conflict-free, shortest-time routes for AGVs. Srivastava, Choudhary, Kumar, and Tiwari (2007) present an intelligent agent-based framework to overcome the inefficacies associated with issues (e.g. conflict-free shortest path, minimum time motion planning, and deadlock avoidance) in the design of AGV control. Wallace (2007) studies the utilization of heuristics to improve the performance of an agent-based AGV controller. These heuristics are used to ascertain whether an AGV could be redirected during negotiations.

Finally, methods in the third category divide the shop floor into several non-overlapping zones and restrict the presence of more than one vehicle in any zone at any time. These methods are referred to as *zone strategies*. In some zone strategies, vehicles are allowed to move into any zones provided they do not enter those that have been occupied by other vehicles (Groover, 1987). Since vehicles are free to visit any zones, there is no need to set up transfer stations between zones. On the other hand, in some zone strategies, e.g. the tandem guide path configuration (Bozer & Srinivasan, 1989, 1991, 1992) and the segmented guide path topology (Sinriech & Tanchoco, 1996), vehicles cannot leave the zones they have been assigned to. In these zone strategies, transfer stations must be set up between zones. In the tandem guide path configuration, each zone has one AGV and the guide path configuration of each zone is a loop. In the segmented guide path topology, every zone also has one vehicle, but its guide path is not confined to any specific configuration. Although these two zone strategies have their differences, they share one common trait – their zones are fixed. This fixed zone trait normally incurs one problem, which is the load balance between vehicles of different zones cannot be maintained if there are changes in the system. This load imbalance problem may cause some zones to bottleneck the entire system and thus hinder the system's performance. To solve the load imbalance problem of the fixed zone strategy, Ho (2000) propose a method in which a zone's area is not fixed. However, his method only works for AGV systems with a single-loop guide path. One cannot apply his method to an AGV system with a more complicated guide path configuration, such as a network configuration. Ho and Hsieh (2004) propose a design methodology for tandem AGV systems with multiple-load vehicles. Their goals include workload balance between vehicles, the minimization of inter-loop flow, and the minimization of flow distance. Shalaby, El Mekkawy, and Fahmy (2006) propose a partitioning algorithm for designing tandem AGV systems. The objectives of their proposed algorithm include minimizing the total handling cost, minimizing the maximum workload in the system, and minimizing the number of inter-zone trips. Tavakkoli-Moghaddam, Aryanezhad, Kazemipoor, and Salehipour (2007) present a non-linear integer mathematical model to minimize both inter-loop and intra-loop flow simultaneously based on balanced-loops strategy. Due to the complexity of the model, they propose a simulated annealing (SA) algorithm to solve the problem.

In this paper, we propose a new zone strategy, termed the dynamic zone strategy, that shares one similar trait with the fixed zone strategy (that being each zone is served by only one vehicle), but does not have the same limitation of a fixed zone area as is seen in the fixed zone strategy. Furthermore, the proposed dynamic zone strategy can be applied to any AGV system with a complicated guide path configuration, even those with a network configuration. The objective of the proposed zone strategy is twofold – preventing vehicle collisions and maintaining the load balance between vehicles. The proposed dynamic zone strategy contains two parts: zone partition design and dynamic zone control. The zone partition design deals with the problem of dividing an AGV system into different zones. It defines a relationship coefficient, which measures the distance relationship and the flow relationship between workstations, and uses it to find an initial zone partition design. This initial design is then improved by an SA-based optimization procedure that identifies a zone partition design with a better load balance. The dynamic zone control uses two methods – zone repartition and load sharing – to ensure that vehicle collisions can be prevented and load balance can be maintained once the system is in operation. The remainder of this paper is organized as follows. The proposed strategy is presented in three sections. First, a few important concepts and issues will be explained. This will help readers more fully understand the methodology behind the proposed strategy. These concepts and issues are presented in Section 2. As is shown later, the proposed zone strategy has two components – zone partition design method and dynamic zone control method. The zone partition design method is presented in Section 3, and the dynamic zone control method in Section 4. Experiments were conducted to test the proposed strategy's performance. These experiments and their results are presented in Section 5. Section 6 summarizes and concludes this research.

2. Related concepts and issues

In this section, we will explain some concepts and issues that are necessary to understand the proposed zone strategy.

2.1. Workstation design

In the proposed zone strategy, all zones are free to change their area to resolve load imbalances between them. However, there is one potential complication with this freedom, that being a transfer station that has been in charge of the load-transferring operations between two zones may not continue to do so if the area of the two zones has been changed. Apparently, some solution is necessary to ensure there will always be at least one transfer station for any two neighboring zones. The proposed solution is giving all workstations the ability to serve as transfer stations. Depending on the current zone partition design, some workstations will also serve as transfer stations. In order for this solution to work, some redesign work is necessary. Fig. 1 illustrates two designs. The first one is for workstations that are adjacent to only one guide path segment (Fig. 1a). We refer to these workstations as Type-I workstations. The second one is for workstations that are simultaneously adjacent to two guide paths (Fig. 1b). These workstations are referred to as Type-II workstations. In both designs, auxiliary flow paths are added to connect workstations with guide paths. In addition, every workstation has two pickup points and two delivery points. To ship a load from Zone A to Zone B in the layout of Fig. 1a, the AGV in Zone A drops the load at Delivery Point I first. The load then follows: a → b → c → d → e → f → g to arrive at Pickup Point II, where it waits for the vehicle in Zone B to pick it up. Furthermore, if the transfer station is a Type-I workstation, the midpoint between its two delivery points defines the boundary of the two neighboring zones (Fig. 1a). If the transfer station is a Type-II workstation, then the point where the two guide paths meet defines the boundary of the two neighboring zones (Fig. 1b).

2.2. Critical point and critical segment

A point is a critical point if one of the following two conditions is met. First, it is the point at which two guide paths meet. Second, it is the middle point of a Type-I workstation's two delivery points. For example, in Fig. 2, there are eighteen critical points. Points A, C, E, F, G, J, N, M, K, P, and R meet the first condition and Points B, D, H, I, L, O, and Q meet the second condition. A critical segment is a guide path segment between two neighboring critical points. For example, $\overline{AB}$, $\overline{BC}$, $\overline{CD}$, $\overline{DE}$, and $\overline{EJ}$ are some of the critical segments in Fig. 2.

2.3. Zone

A zone is made up of critical segments connected with each other and the workstations at these critical segments. For example, Fig. 2 has three zones – I, II, and III. Zone I has nine critical segments (i.e. $\overline{AF}$, $\overline{FG}$, $\overline{FO}$, $\overline{AB}$, $\overline{BC}$, $\overline{CG}$, $\overline{CD}$, $\overline{DE}$, and $\overline{EJ}$), Zone II six (i.e. $\overline{ON}$, $\overline{NM}$, $\overline{GM}$, $\overline{ML}$, $\overline{MP}$, and $\overline{PQ}$) and Zone III seven (i.e. $\overline{GH}$, $\overline{HI}$, $\overline{IJ}$, $\overline{JK}$, $\overline{KL}$, $\overline{KR}$, and $\overline{RQ}$). Zone I has four workstations (i.e. WS 1, WS 2, WS 3, and WS 5), Zone II five (i.e. WS 5, WS 6, WS 8, WS 9, and WS 10) and Zone III six (i.e. WS 3, WS 4, WS 6, WS 7, WS 8, and WS 10). Note that some workstations appear in more than one zone. These workstations are transfer stations.

2.4. Shortest feasible paths

A shortest feasible path is made up of critical segments. The shortest feasible path between two workstations exists if the following two conditions are both true. First, both workstations belong to the same zone or one of them belongs to one zone but not the other. Second, the shortest path between these two workstations does not cut through any zones other than the zone that either workstation belongs to. Fig. 3 gives two examples. It is assumed that in these examples WS 1 and WS 2 have been assigned to the same zone, and WS 3 has been designated as the seed workstation of a new zone (i.e. WS 3 is the first workstation of this new zone). WS 4 has not been assigned to any zones yet. In Fig. 3, the shortest feasible paths and the infeasible paths of both examples are illustrated.

2.5. Relationship coefficient (RC) and unrelated coefficient (UC)

The RC allows one to measure the degree of relationship between two workstations. An RC is made of two coefficients – the distance relationship coefficient (DR) and the flow relationship coefficient (FR). The first one is distance-related. The shorter the distance between two workstations, the greater their DR is. The second one considers the amount of flow between workstations. The greater the flow between two workstations, the greater their FR is. Eqs. (1)–(3) show how RC_{ij} – the relationship coefficient between two workstations, i and j – can be calculated. Table 1 gives the notation definition. Finally, the unrelated coefficient (UC) measures how unrelated two workstations are. UC_{ij} is used to denote the UC value between workstations i and j and is equal to $(1 - RC_{ij})$.

$$RC_{ij} = W_d DR_{ij} + W_f FR_{ij}, \tag{1}$$

$$DR_{ij} = \frac{MAXSD - sd_{ij}}{MAXSD - MINSD}, \tag{2}$$

$$FR_{ij} = \frac{f_{ij} - MINF}{MAXF - MINF}. \tag{3}$$

3. The proposed zone partition design method

In this section, the first part of the proposed zone strategy – the zone partition design method – is introduced. The goal is to come up with a zone design that can meet the following two objectives:

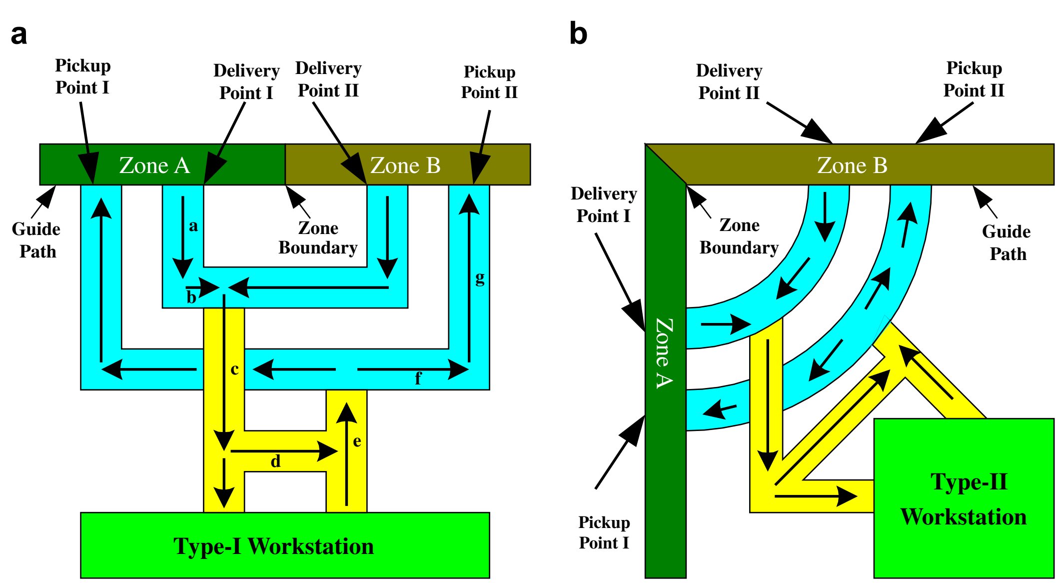

Fig. 1. Two proposed designs for different types of workstations.

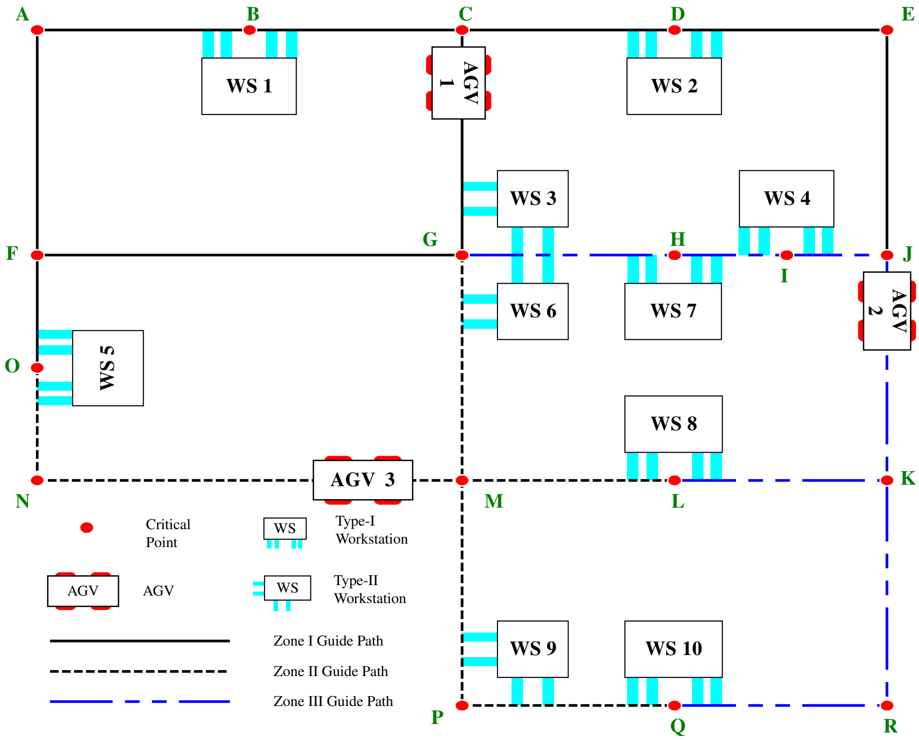

Fig. 2. A layout with identified zones.

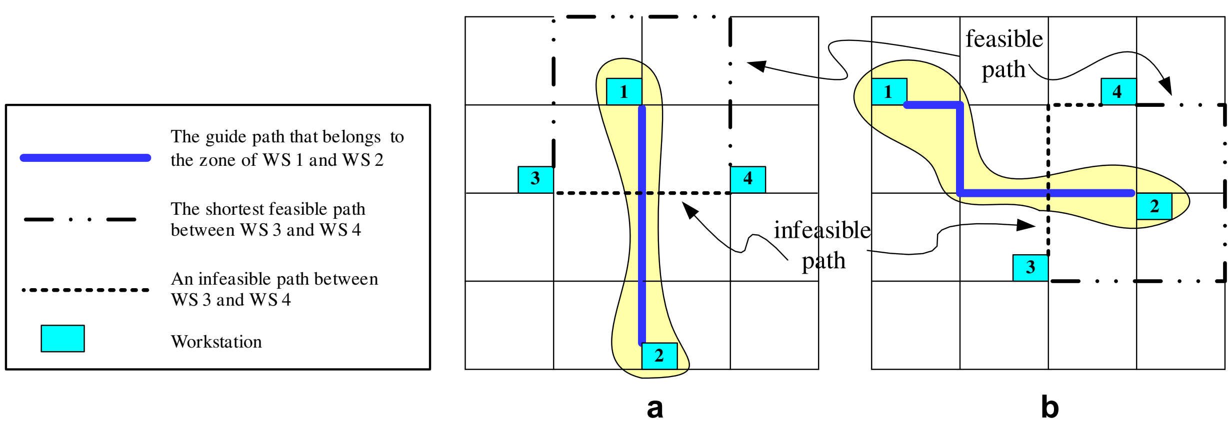

Fig. 3. Examples of the shortest feasible paths.

Table 1
Notation definition

DR_{ij}	The distance relationship coefficient between i and j
sd_{ij}	The length of the shortest feasible path between i and j
MAXSD	Of all the shortest feasible paths in the system, *MAXSD* is the length of the longest one
MINSD	Of all the shortest feasible paths in the system, *MINSD* is the length of the shortest one
FR_{ij}	The flow relationship coefficient between i and j
f_{ij}	The flow between i and j
MAXF	The largest flow between two workstations in the system
MINF	The smallest flow between two workstations in the system
W_d	the weight for the distance relationship coefficient, where $0 \leqslant W_f \leqslant 1$ and $W_d + W_f = 1$
W_f	The weight for the flow relationship coefficient, where $0 \leqslant W_f \leqslant 1$ and $W_d + W_f = 1$

load balance between vehicles of different zones and the minimization of the total flow distance in each zone. As is shown in Fig. 4, the proposed method has three stages: determine the number of zones, determine the area of each zone, and validate the zone partition design. These tasks and their solutions are explained as follows.

3.1. Stage 1 – Determine the number of zones

Since each zone has only one vehicle, determining the number of zones is equivalent to determining the number of vehicles. In this paper, a method proposed by Egbelu (1987) is adopted. This method uses Eqs. (4)–(8) to estimate the required number of vehicles. Table 2 gives the notation definition. Before these equations can be applied, the issue of calculating the distance between any two workstations needs to be resolved first. Since every workstation has two delivery points and two pickup points, it is impossible to predetermine which pickup point or delivery point will be used in a specific trip. This makes it impossible for us to have an accu-

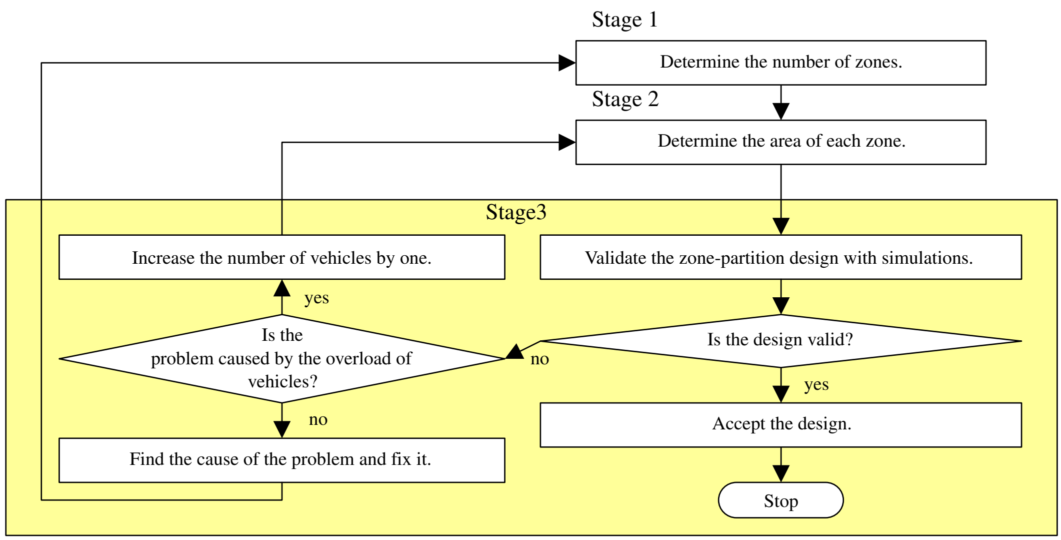

Fig. 4. The flow chart of the zone partition design method.

Table 2
Notation definition

nw	The number of workstations in the system
f_{ij}	The number of loads from workstation i to j during the period or shift. Since it is assumed here that all vehicles are single-load AGVs, f_{ij} is also equal to the number of loaded trips from workstation i to j during the period or shift
g_{ij}	The expected number of empty trips from workstation i to j during the period or shift
DA_{ij}	The expected total distance of empty trips from workstation i to$j$ (ft)
DB_{ij}	The expected total distance of loaded trips from workstation i to$j$ (ft)
d_{ij}	The distance between workstations i to j (ft)
V	The average vehicle travel speed (ft/min).
t_l	The mean time to load a vehicle (min)
t_u	The mean time to unload a vehicle (min)
T	The length of a period or shift (hour)
TT	The estimated total operational time for vehicles (min)
lt	The expected lost time by each vehicle during a time period or shift due to battery change (min)
nv	The number of required vehicles
nz	The number of required zones. Note: $nz = nv$

rate distance calculation. To resolve this problem, it is suggested that the midpoint between the two delivery points of a workstation be used as the reference point of the workstation. Reference points are used in calculating the distances between workstations.

$$g_{ij} = \frac{\left(\sum_{k=1}^{nw} f_{ki}\right)\left(\sum_{k=1}^{nw} f_{jk}\right)}{\sum_{m=1}^{nw}\sum_{n=1}^{nw} f_{mn}}, \tag{4}$$

$$DA_{ij} = g_{ij}d_{ij}, \tag{5}$$

$$DB_{ij} = f_{ij}d_{ij}, \tag{6}$$

$$TT = \frac{\sum_{i=1}^{nw}\sum_{j=1}^{nw}(DA_{ij} + DB_{ij})}{V} + \left(\sum_{i=1}^{nw}\sum_{j=1}^{nw} f_{ij}\right)(t_u + t_l), \tag{7}$$

$$nv = nz = TT/(60T - lt). \tag{8}$$

3.2. Stage 2 – Determine the area of each zone

Here, we propose a three-phase design method to determine the area of each zone. During Phase 1, an initial zone partition design is created. This initial design is then improved during Phase 2. Setting up transfer stations and assigning the remaining critical segments are performed during Phase 3.

3.2.1. Phase 1 – Initial zone partition design

The steps for generating an initial zone partition design are as follows:

Step 1. Let *RWS* stand for the set of workstations that have not been assigned to any zones yet. Initialize $W_d = 0.5$ and $W_f = 0.5$.

Step 2. Calculate the unrelated coefficient between every two workstations in RWS.

Step 3. From RWS, find nz workstations whose sum of unrelated coefficients is the greatest. These workstations are seed workstations. Each seed workstation is assigned to a zone. Let SW_z $(1 \leqslant z \leqslant nz)$ stand for the seed workstation assigned to *Zone z*. Delete these seed workstations from *RWS*.

Step 4. Expand the area of each zone as follows.

4.1 Let $GW(z)$ be the workstations of *Zone z* and $GS(z)$ the critical segments of *Zone z*. At the beginning, every zone has one workstation (i.e. its seed workstation) and no critical segment.

4.2 Initialize $K = 1$.

4.3 Calculate the relationship coefficient between every workstation in $GW(K)$ and every workstation in *RWS*. If no workstations in $GW(K)$ have any relationship coefficients with workstations in *RWS*, then it means that no feasible paths exist between workstations in $GW(K)$ and workstations in *RWS*. In this case, go to step 4.7.

4.4 From the relationship coefficients calculated above, select the greatest one. Let i and j stand for the workstations with the greatest coefficient, where i is from $GW(K)$ and j is from *RWS*.

4.5 Connect i and j with the shortest feasible flow path. Update $GW(K)$ by adding j to $GW(K)$. Also, update $GS(K)$ by adding those new critical segments that are in the shortest feasible path but not in $GS(K)$ to $GS(K)$. Delete j from *RWS*.

4.6 Are there any workstations left in *RWS*? If yes, go to step 4.7; otherwise stop.

4.7 Compare K and nz.
   4.7.1. If $K \geqslant nz$, set $K = 1$ and, based on the intermediate zone partition design, update W_d and W_f if necessary (see the explanation below) and return to step 4.3.
   4.7.2. If $K < nz$, set $K = K + 1$ and return to step 4.3.

Several issues are further explained as follows:

- After Phase I, workstations assigned to the same zone are connected by critical segments that together have a tree configuration. We will refer to these critical segments as "core critical segments" and the workstations at the tree's tips as tip workstations. Fig. 5 gives an example of what a typical zone partition design may look like after Phase I. In zone α, the critical segments are E, F, and G; the tip workstations are workstations 4, 5, and 6.
- Other issues include the initialization of W_d and W_f, and why they should be updated during the design process. W_d and W_f are weights for *DR* and *FR*. Since the pilot tests of the proposed method show that setting $W_d = 0.5$ and $W_f = 0.5$ can produce good designs, it is thus suggested that W_d and W_f be weighted equally with a value of 0.5 initially. However, human designers are encouraged to try different settings for W_d and W_f, identify different zone partition designs, and select the best one as the initial zone partition design. As the design procedure proceeds, it is likely that the addition of a workstation w to a zone z generates an awkward flow path connection. Usually w has to be some distance away from z in order for this to occur. Such situations are more likely to happen when the design procedure is near the end. This is because at the early stage it is easy for a zone to find a nearby workstation to include in. However, as the design procedure proceeds, the number of available workstations that can be assigned to zones will decrease. This decrease will make it more likely for a workstation to be included into a distant zone. It is thus suggested that W_d be increased gradually and W_f decreased accordingly as the design procedure proceeds. By doing so, occurrences of the problem mentioned here can be greatly reduced. In addition, as is shown at step 4.7.1, human designers can determine how much to increase W_d and to decrease W_f, based on the intermediate zone partition design.

3.2.2. Phase 2 – Zone partition design improvement

The zone partition design obtained during Phase 1 may not have a very good load balance result. During Phase 2, we propose a Simulated Annealing (SA) based improvement procedure to obtain another zone partition design with a better load balance result.

3.2.2.1. The SA-based improvement procedure

Step 1. Let c be the iteration counter, $c_{\max}$ the maximum number of iterations, T the temperature, $p(0)$ the initial zone partition design, $p(c)$ the zone partition design of the cth iteration, and $\hat{p}$ the optimal zone partition design. Initialize $c_{\max}$, T_i, and T_f. Set $c = 0$ and $\hat{p} = p(c)$.

Step 2. Randomly select two neighboring zones, α and β, from $p(c)$ and $L_{p(c)\alpha} \geqslant L_{p(c)\beta}$. From the tip workstations of α, identify those that are adjacent to the tip workstations of β. Let $WS(\alpha, \beta)$ be the set of these identified tip workstations, $CCS(\alpha)$ the set of core critical segments of α and $CCS(\beta)$ the set of core critical segments of β.

Step 3. Randomly select a tip workstation m from $WS(\alpha,\beta)$. Let m' be the nearest workstation that m is linked to with core critical segments in α. And, let m'' be the tip workstation in β that m is most adjacent to.

Step 4. Generate a new zone partition design p' by performing the following steps.
   4.1. Disconnect m from α by deleting the core critical segments connecting m with m' from $CCS(\alpha)$.
   4.2. Connect m and m'' with the shortest feasible path between them. Add the critical segments in this shortest path to $CCS(\beta)$.

Step 5. Calculate $vd = SV_{p(c)} - SV_{p'}$, where SV_p is the evaluation value of a design p. The calculation of SV_p will be explained later.
   5.1. If $vd \geqslant 0$, accept p'.
   5.2. If $vd < 0$, generate a random number r between $[0,1]$. If $r \leqslant e^{(vd/T)}$, accept p'; otherwise discard p' and go back to step 2.
   5.3. Increment $c \leftarrow c + 1$ and update $p(c) \leftarrow p'$.
   5.4. If $SV_{p'} < SV_{\hat{p}}$, update $\hat{p} \leftarrow p'$.
   5.5. Reduce temperature T if necessary.

Step 6. If $c \leqslant c_{\max}$, return to step 2; otherwise stop and $\hat{p}$ is the final improved zone partition design found by the SA-based improvement procedure.

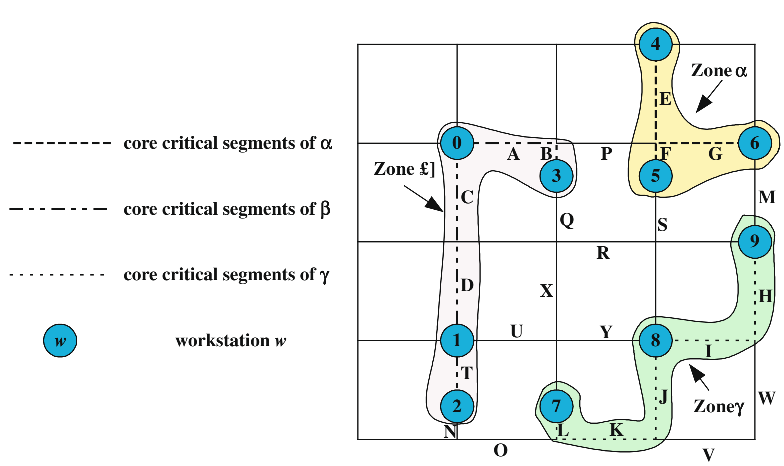

Fig. 5. An example of a zone partition design.

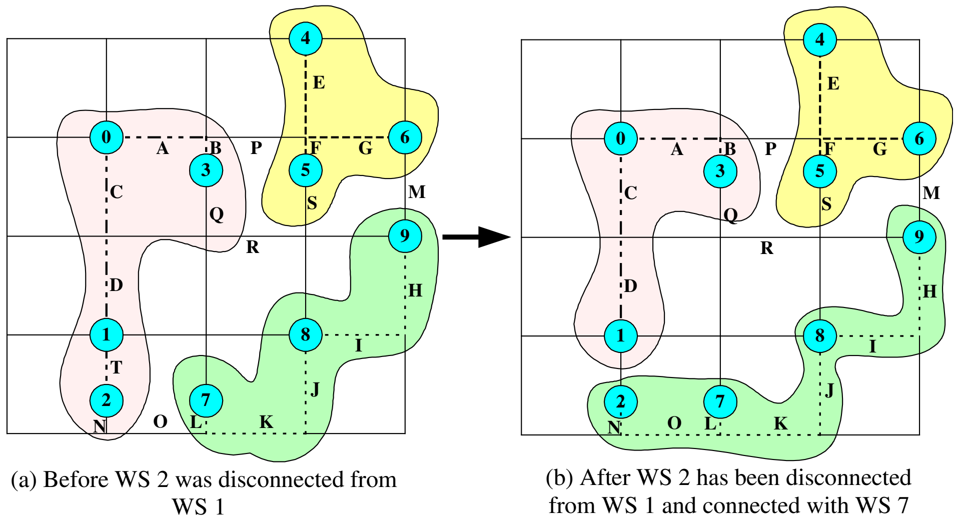

Fig. 6. An illustration of the zone partition design improvement operation.

3.2.2.2. Initial solution and generating neighboring solutions. As shown above, SA needs an initial solution to start with. In the proposed method, we use the zone partition design solution found during Phase 1 as the initial solution (see step 1). SA also needs to generate neighboring solutions from the current solution. This is done by disconnecting a workstation from one zone and connecting it to one of its neighboring zones (see steps 2–4). Fig. 6 illustrates an example in which a workstation WS 2 is disconnected from its original zone then connected to its neighboring zone.

3.2.2.3. Evaluation of a zone partition design. The equations for calculating SV_p – the evaluation value of a zone partition design p – are as follows. In the equations, z' and z'' are zone indexes and L_{pz} stands for the load of the vehicle in a zone z of a zone partition design p. The evaluation function produces values that are between zero and one.

$$TZLD_p = \sum_{z'=1}^{nz-1} \sum_{z''=z'+1}^{nz} |L_{pz'} - L_{pz''}|, \tag{9}$$

$$SV_p = \frac{TZLD_p}{\left(\sum_{z'=1}^{nz} L_{pz'}\right) \times (nz - 1)}. \tag{10}$$

The SV value of a zone partition design is between zero and one. A perfectly balanced zone partition design will have a zero SV. The smaller the SV value of a zone partition design, the better the design's load balance result. These equations also require us to calculate L_{pz}. Eqs. (11)–(14) are used for calculating L_{pz}. They are similar to Eqs. (4)–(7), with the difference being their notation. This is because Eqs. (4)–(7) are for the workload calculation of the entire system, while Eqs. (11)–(14) are for the workload calculation of an individual zone. Please note f_{pzij} of Eq. (11) is not necessary equal to f_{ij} of Eq. (4). If i is a tip workstation, then f_{pzij} includes not only f_{ij} (which is the flow originating from i and going to j), but also the flow that originates from workstations in other zones, passes i, and arrives at j, and the flow that originates from workstations in other zones, passes i, passes j, and arrives at workstations in other zones. Table 3 gives the notation definition.

Table 3
Notation definition

$ws(p,z)$	The set of workstations in a zone z of a zone partition design p
SV_p	The evaluation value of a zone partition design p
L_{pz}	The load of the vehicle in a zone z of a zone partition design p (min)
d_{pzij}	The distance from workstation i to j in a zone z of a design p (ft)
f_{pzij}	The number of loads from workstation i to j in a zone z of a design p during the period or shift
g_{pzij}:	the expected number of empty trips from workstation i to j in a zone z of a zone partition design p during the period or shift
DA_{pzij}	The expected total distance of empty trips from workstation i to j in a zone z of a design p (ft)
DB_{pzij}	The expected total distance of loaded trips from workstation i to j in a zone z of a design p (ft)

$$g_{pzij} = \frac{\sum_{k\in ws(p,z)} f_{pzki} \sum_{k\in ws(p,z)} f_{pzjk}}{\sum_{m\in ws(p,z)} \sum_{n\in ws(p,z)} f_{pzmn}}, \tag{11}$$

$$DA_{pzij} = g_{pzij} d_{pzij}, \tag{12}$$

$$DB_{pzij} = f_{pzij} d_{pzij}, \tag{13}$$

$$L_{pz} = \frac{\sum_{i\in ws(p,z)} \sum_{j\in ws(p,z)} (DA_{pzij} + DB_{pzij})}{V} + \left(\sum_{i\in ws(p,z)} \sum_{j\in ws(p,z)} f_{pzij}\right)(t_u + t_l). \tag{14}$$

3.2.2.4. Initial temperature and freezing temperature setting. Kouvelis, Chiang, and Fitzsimmons (1992) showed that SA is more likely to obtain good-quality solutions if the initial probability of accepting inferior solutions is around 0.8. Ho and Moodie (1998) took their advice and proposed a temperature range [3.5,5.5] for the initial temperature, providing the difference between the current solution value and the next solution value is in the [−1,1] range. This initial temperature range results in a [0.7515,0.8338] probability range (that is near 0.8) for accepting inferior solutions. Since the difference between the current solution value and the next solution value (i.e. $vd = SV_{p(c)} - SV_{p'}$) of our problem is also in the [−1,1] range, the initial-temperature range proposed by Ho and Moodie (1998) can be adopted by us. The freezing temperature T_f is the final temperature of SA. If T_f is set too low, SA may spend an inordinate amount of time completing the final iterations, which will harm its performance. Ho and Moodie (1998) conducted many experiments and proposed another range [0.2,0.5] for the freezing temperature, providing the difference between the current solution value and the next solution value is in the [−1,1] range. This freezing temperature range can also be adopted by us since the vd of our problem is between −1 and 1.

3.2.2.5. Temperature reduction strategy. Ho and Moodie (1998) proposed and compared four different temperature reduction strategies. They found that the third strategy, the decelerating-step decreasing strategy, has the best performance. This strategy is adopted in this paper. Under this strategy, the temperature is decreased M times and the number of iterations between two consecutive reductions is $c_{\max}/M$. The temperature after the nth reduction (i.e. T_n) is equal to $T_i \cdot \alpha^n$, where $\alpha = \sqrt[M]{T_f/T_i}$.

3.2.3. Phase 3 – Set up transfer stations and assign remaining critical segments to zones

There are two tasks to be accomplished during Phase 3. The first task is setting up transfer stations between neighboring zones. Since it is likely that after Phase 2 some critical segments might still have not been assigned to any zones yet, thus the second task is to assign these unassigned critical segments (if there are any) to zones.

3.2.3.1. Setting up transfer stations. In this section, we propose a method for determining which workstation should serve as a transfer station. The objective is to set up as many transfer stations between any two neighboring zones as possible. The greater the number of transfer stations set up between neighboring zones, the greater the number of alternative routes available for vehicles, which can be translated as "the higher the AGV system's routing flexibility". Setting up more transfer stations will not incur any additional cost in the system here, as every workstation has been designed to have the capability to serve as a transfer station. As one will see in the proposed method, different situations will be encountered when setting up transfer stations. Also, different situations may have different concerns. For example, in one situation, its concern is achieving the load balance between zones; while in another situation, providing feasible flow paths for vehicles is its concern. Detailed steps of the proposed method are as follows. An explanation will also be provided to help readers understand the proposed method.

Step 1. From the zone partition design p, select two neighboring zones, α and β, that have not gone through the following procedure yet. If every pair of neighboring zones has gone through this procedure, then stop.

Step 2. From α's tip workstations, identify those adjacent to β. Let $TW(\alpha)$ stand for these tip workstations.

Step 3. From β's tip workstations, identify those adjacent to α. Let $TW(\beta)$ stand for these tip workstations.

Step 4. Set $i = 1$.

Step 5. Collect all valid paths between the tip workstations in $TW(\alpha)$ and the tip workstations in $TW(\beta)$. Let $SPC_{TW(\alpha),TW(\beta)}$ stand for these paths. A path is valid if it is made up of critical segments that have been assigned to α or β, or have not been assigned to any zone yet. In other words, a path is invalid if it contains a critical segment that has been assigned to a zone other than α or β.

Step 6. From $SPC_{TW(\alpha),TW(\beta)}$, select the shortest path, say $SP(i)$. If $SPC_{TW(\alpha),\ TW(\beta)}$ is empty, go to step 1.

Step 7. Let $A(i)$ and $B(i)$ stand for the tip workstations connected by $SP(i)$, where $A(i)$ is from $TW(\alpha)$ and $B(i)$ is from $TW(\beta)$. Let $CSP(i)$ stand for the critical segments in $SP(i)$.

Step 8. If there are no critical segments in $CSP(i)$ belonging to α or β, then do the following; otherwise go to step 9.

8.1. Compare the workload of the vehicle in α, i.e. $L_{p\alpha}$, with the workload of the vehicle in β, $L_{p\beta}$.

8.2. If $L_{p\alpha} \geqslant L_{p\beta}$, designate $A(i)$ as the transfer station between α and β, and assign the critical segments in $CSP(i)$ to β.

8.3. If $L_{p\beta} > L_{p\alpha}$, designate $B(i)$ as the transfer station between α and β, and assign the critical segments in $CSP(i)$ to α.

8.4. Delete all the paths between $A(i)$ and $B(i)$ from $SPC_{TW(\alpha),TW(\beta)}$.

8.5. Increase i by one and return to step 6.

Step 9. If there are no critical segments in $CSP(i)$ belonging to β but there is a subset of critical segments in $CSP(i)$, say $SCSP(i)$, belonging to α, then do the following; otherwise go to step 10.

9.1. Designate $B(i)$ as the transfer station between α and β.

9.2. Let $RCSP(i)$ stand for the set of those critical segments that are in $CSP(i)$ but not in $SCSP(i)$. Assign the critical segments in $RCSP(i)$ to α.

9.3. Delete all the paths between $A(i)$ and $B(i)$ from $SPC_{TW(\alpha),TW(\beta)}$.

9.4. Increase i by one and return to step 6.

Step 10. If there are no critical segments in $CSP(i)$ belonging to α, but there is a subset of critical segments from $CSP(i)$, say $GCSP(i)$, belonging to β, then do the following; otherwise go to sep 11.

10.1. Designate $A(i)$ as the transfer station between α and β.

10.2. Let $FCSP(i)$ stand for the set of those critical segments that are in $CSP(i)$ but not in $GCSP(i)$. Assign the critical segments in $FCSP(i)$ to β.

10.3. Delete all the paths between $A(i)$ and $B(i)$ from $SPC_{TW(\alpha),TW(\beta)}$.

10.4. Increase i by one and return to step 6.

Step 11. If there is a subset of critical segments of $CSP(i)$ belonging to α and there is another subset of critical segments of $CSP(i)$ belonging to β, then do the following steps:

11.1. Delete $SP(i)$ from $SPC_{TW(\alpha),\ TW(\beta)}$.

11.2. Increase i by one and return to step 6.

As shown in the steps above, one may encounter four situations when setting up transfer stations between zones. These situations and their solutions are described in steps 8, 9, 10, and 11, respectively. To help readers understand these situations and their solutions, they are illustrated with the zone partition design p in Fig. 5. There are three zones (α, β, and γ) in p and we assume $L_{p\alpha} > L_{p\beta} > L_{p\gamma}$. The case in step 8 can be illustrated with an example of setting up transfer stations between α and γ. Let us assume that the selected shortest path is the path between WS 6 and WS 9. As one can see in the figure, this path is made up of only one critical segment M. Since M belongs to neither α nor γ, either WS 6 or WS 9 can serve as the transfer station between α and γ depending on whichever zone's vehicle has a heavier load. Since the vehicle in α has a heavier workload than the vehicle in γ, WS 6 will be the transfer station and the critical segment, M, in the shortest path between WS 6 and WS 9 will be assigned to γ. By making WS 6 the transfer station between α and γ, the vehicle with less workload (i.e. the vehicle in γ) will assume the responsibility to pick up loads coming from α to γ and deliver loads leaving γ for α at WS 6. By assigning this extra work to the vehicle with the lesser workload, the load balance quality of the zone partition design will be further improved. The situations in step 9 and step 10 are similar. Their situations can be illustrated with an example of setting up transfer stations between β and γ. Let us assume that the selected shortest path between the tip workstations of β and γ is the one connecting WS 2 and WS 7. This path is made up of three critical segments, N, O, and L. As one can see, one of these critical segments, L, is in γ. If WS 7 becomes the transfer station between β and γ, then the AGV in β will not be able to transfer any loads from β to γ, since it cannot follow this shortest path to reach WS 7. On the other hand, since none of these three critical segments belong to β, WS 2 will be designated as the transfer station between β and γ. Finally, the situation in step 11 can be explained with an example of setting up transfer stations between α and β. Let us assume that the selected shortest path is the one connecting

WS 3 and WS 5. This shortest path contains three critical segments – B, P, and F. Among them, B belongs to β and F belongs to α. It is apparent that with this path, WS 3 cannot be designated as the transfer station between α and β, because the vehicle in α cannot follow this shortest path to pick up transferred loads at WS 3. For a similar reason, WS 5 cannot be designated as the transfer station either. As a result, this shortest path will be deleted from further consideration.

3.2.3.2. Assigning the remaining critical segments to zones. After setting up the transfer stations, there may be some not-yet-assigned critical segments left in the system. To complete the zone partition design, they need to be assigned to zones. Examination of these critical segments will show that they belong to one of two types. Type-I critical segments have only one zone that they can be assigned to, making their assignment easily determinable. For example, E and F in Fig. 7 can only be assigned to Zone γ, as there are no other zones that they can be assigned to. For Type-II critical segments, their zone assignment is not immediately obvious, as there may be more than one zone that they can be assigned to. For example, in Fig. 7 segments R, S, T, U, V, W, and X could be assigned to more than one zone. To solve this problem, a simple yet efficient procedure that can produce feasible assignments is proposed. The proposed procedure is based on a simple rule – critical segments that are directly connected with the critical segments of any zones should be assigned first. A critical segment is directly linked with another critical segment if there are no other critical segments between them. For example, the critical segment U in Fig. 7 is directly linked with the critical segment Y. Detailed steps of the proposed assignment procedure are as follows:

Step 1. Let GS stand for the set of Type-II critical segments.
Step 2. From GS, randomly select a critical segment, *H*, that is directly linked to critical segments that have already been assigned to zones.
Step 3. Identify the set of critical segments that have been assigned to zones and are directly linked to H. Let *DLC*(*H*) stand for this set.
Step 4. From DLC(*H*), randomly select a critical segment, say *SB*. And, let *Z*(*SB*) stand for the zone that *SB* has been assigned to.
Step 5. Assign H to *Z*(*SB*) and delete *H* from *GS*.
Step 6. Repeat steps 2 to 5 until GS is empty.

3.3. Stage 3 – Validate the zone partition design

As shown in Fig. 4, after a zone partition design has been completed, the next thing that needs to be done is to confirm if the design can meet the production demand. Simulation experiments need to be conducted to validate the design. After experiments, it may be discovered that the number of required vehicles (zones) has been underestimated or there may be other unexpected factors that can hinder production. Please note that in these simulations the proposed dynamic zone control method (which will be presented in Section 4) can be adopted. After all, the main purpose of this validation stage is to ensure the design can work as well in real life as it does on paper.

4. The dynamic zone control method

The second part of the proposed zone strategy is the dynamic zone control method. After the zone partition design has been obtained, the system is ready to operate. The system should be operating well, provided the production environment remains unchanged. However, real systems rarely remain unchanged. Many things or events can destroy the load balance established earlier. Our goal here is to come up with a method that can maintain the system's load balance when it begins to operate. To accomplish this goal, the proposed control strategy relies on three modules – Monitoring Module, Control Module, and Solution Module (see Fig. 8). The function of the Monitoring Module is to monitor the status of the system and report it to the Control Module. The Control Module is the brain of the dynamic control strategy. It receives the system status report from the Monitoring Module. And, based on the report, it instructs the Monitoring Module or the Solution Module to take appropriate actions. The Solution Module contains the methods for solving load imbalance problems. It receives instructions from the Control Module and adopts the appropriate load-balancing method.

4.1. Monitoring module

The Monitoring Module has two monitoring schemes – Normal Monitoring (NM) scheme and Intensive Monitoring (IM) scheme.

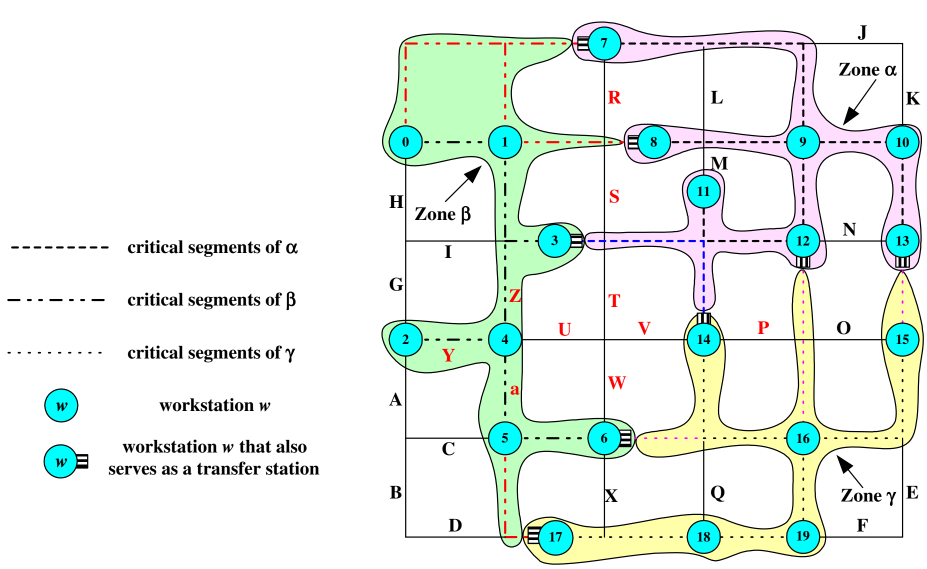

Fig. 7. A zone partition design example.

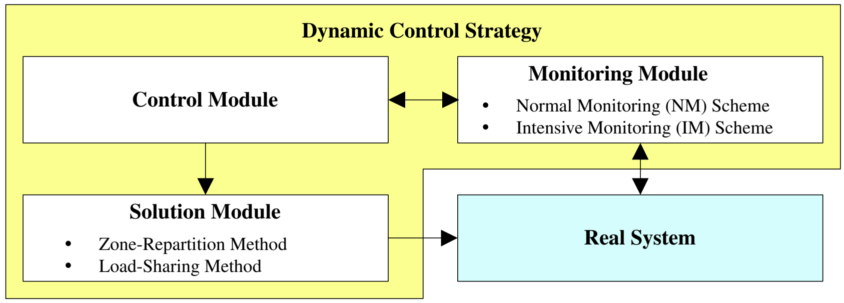

Fig. 8. Modules in the proposed dynamic zone control method.

Both schemes check the system's status at regular intervals, except the IM scheme's interval is shorter than that of the NM scheme. The NM scheme is applied only when the system has remained stable for a pre-specified period of time. By stable, it is meant that no load imbalance problems have occurred. On the other hand, the IM scheme is applied whenever there is a sign indicating the system is not stable. For example, let us assume that the Monitoring Module has detected a load imbalance problem and reported it to the Control Module. Let us also assume that the NM scheme is currently in use. At the moment the Control Module receives the message from the Monitoring Module; it directs the Monitoring Module to switch to the IM scheme. At the same time, it asks the Solution Module to activate one of its solution methods to solve the load imbalance problem. The Monitoring Module will continue applying the IM scheme until the system becomes balanced again. Determining which monitoring scheme to adopt is the job of the Control Module. Details of the Control Module will be explained in Section 4.2. Finally, please note a load imbalance problem occurs whenever the vehicles of two neighboring zones have a load difference greater than a tolerance value specified by the designer. In this paper, we calculate the workload of every zone's vehicle in the last t_0 minutes, where t_0 is specified by the designer. If the workload difference between vehicles of two neighboring zones is greater than a pre-specified tolerance value, then a load imbalance problem is detected.

4.2. Control module

The flow chart of the Control Module is shown in Fig. 9. As shown, a system is under the IM scheme whenever there is a sign indicating the system may be unstable. On the other hand, the system will be under the NM scheme only when it is determined that the system is stable. As is shown in Fig. 9, stability is achieved when the system has remained stable for t_2 minutes, where t_2 is specified by the human designer and $t_2 \geqslant t_0$. The load-sharing method is always the first method employed to solve the load imbalance problem. Furthermore, only when it has been confirmed that the load imbalance problem still has not been resolved by the load-sharing method after t_1 minutes, will the zone-repartition method be employed to solve the load imbalance problem. Please note t_1 is also specified by the designer and $t_1 \geqslant t_0$.

4.3. Solution module

The Solution Module has two methods – the zone-repartition method and load-sharing method. They are designed for different load imbalance problems, namely, transient and perpetual. A perpetual load imbalance problem is caused by changes that have long-term effects on the system. To solve a perpetual load imbalance problem, a remedy like the zone-repartition method that can solve the problem once for all is more appropriate. On the

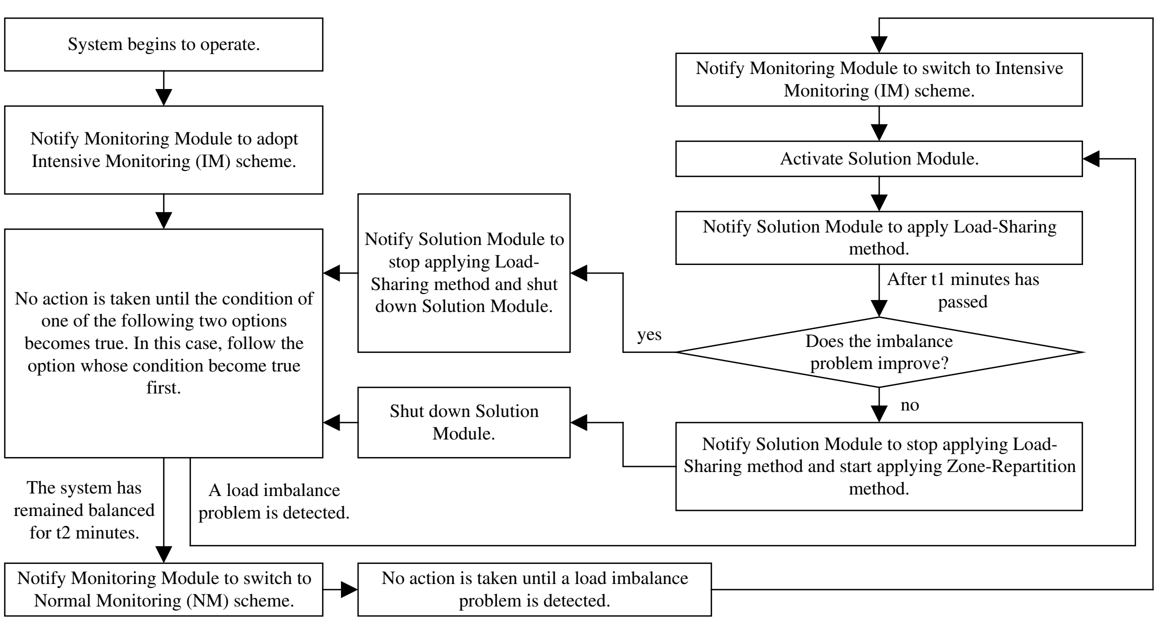

Fig. 9. The flow chart of the Control Module.

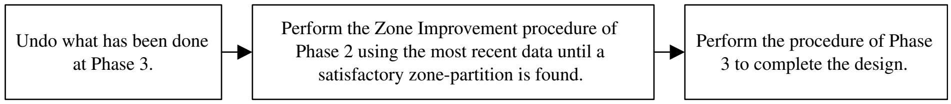

Fig. 10. Major steps of the zone-repartition method.

other hand, a transient load imbalance problem is caused by changes whose effects on the system are short-lived. For a transient load imbalance problem, a remedy like the load-sharing method that can alleviate the negative effects caused by the load imbalance problem is more appropriate. These two load-balancing methods will be explained as follows.

4.3.1. Zone-repartition method

Fig. 10 shows the three major steps of the zone-repartition method. During the first step, we undo what has been done during Phase 3. The idea is to make the zone partition design return to the state right before Phase 3. After that, the zone improvement procedure of Phase 2 can be performed using the most recent data to reflect any recent changes. This improvement procedure will not stop until a satisfactory design is found. After that, Phase 3 will be performed again to complete the design.

4.3.2. Load-sharing method

The load-sharing method is not designed to eradicate the cause behind the load imbalance problem, but to lessen or eliminate the negative effects brought about because of it. The idea is to have vehicles help their neighboring vehicles that are busier than they are so that the load imbalance problem can be improved. A vehicle in Zone A, say V(A), will perform load-sharing with *V*(B) – the vehicle in one of its adjacent zones, Zone B – when the following conditions are all true. First, there is a load imbalance problem between Zone A and Zone B. Second, V(A) is carrying a part, say *P*, which needs to visit a workstation, say WS_d, in Zone B. Third, V(A) has just arrived at a workstation, say WS_0, which is the transfer station between Zone A and Zone B. Let $WS_0 \to WS_1 \to WS_2 \to \cdots \to WS_d$ stand for the route that V(A) needs to follow in order to deliver *P* from WS_0 to WS_d. Initialize $j = 0$ to reflect the fact that V(A) is currently at WS_0. The load-sharing method has two routines – the moving-on routine and heading-back routine. The moving-on routine tries to move V(A) to WS_d. In this routine, V(A) move from its current workstation WS_j to the next workstation WS_{j+1} in its route (i.e. $WS_0 – WS_1 \to WS_2 \to \cdots \to WS_d$) to reach WS_d. In order for V(A) to move from WS_j to WS_{j+1}, the following three conditions all need to be true:

- First, the vehicle in Zone B, V(B), is currently busy.
- Second, there are no jobs waiting for the service of V(A) in Zone A.
- Third, V(A) will not collide with V(B) if it moves from WS_j to WS_{j+1}.

If these conditions are all true, V(A) will move fromWS_j to WS_{j+1}. Once V(A) has reached WS_{j+1}, V(A)'s position is updated by making $j = j + 1$. This moving-on routine will be repeated until V(A) has reached WS_d or any one of the above three conditions is no longer true. After this, the following two cases are checked to see which one is true.. In the first case, $j = 0$ (which is the same as saying the moving-on routine was never performed), thus V(A) did not enter Zone (B) and is still at WS_0. If this case is true, V(A) will drop P at drop *P* at WS_0 and resume its operations in its own zone, Zone (A). In the second case, $j > 0$, that is V(A) has already entered Zone (B) and is no longer at WS_0. If this case is true, V(A) will drop *P* at WS_j, then start heading back to Zone A by activating the heading-back routine.

In the heading-back routine, V(A) is heading back to its own zone, Zone A. On its way back to Zone (A), V(A) will also assist V(B) in delivering some parts if the following two conditions are true. First, these parts have to be at the workstations that are at V(A)'s route (i.e. $WS_j \to WS_{j-1} \to \cdots \to WS_0$) back to Zone A, where WS_j is V(A)'s current location. Second, the destinations of these parts are also at V(A)'s route back to WS_0. In other words, V(A) will only deliver those parts that do not require it to detour from its route back to Zone A. When V(A) starts going back to Zone A, it needs to check whether there are parts waiting to be picked up at WS_j. If yes, V(A) will collect those parts whose destinations are at V(A)'s route ($WS_j \to WS_{j-1} \to \cdots \to WS_0$) back to Zone A. If V(A) has successfully collected any parts meeting this condition, then

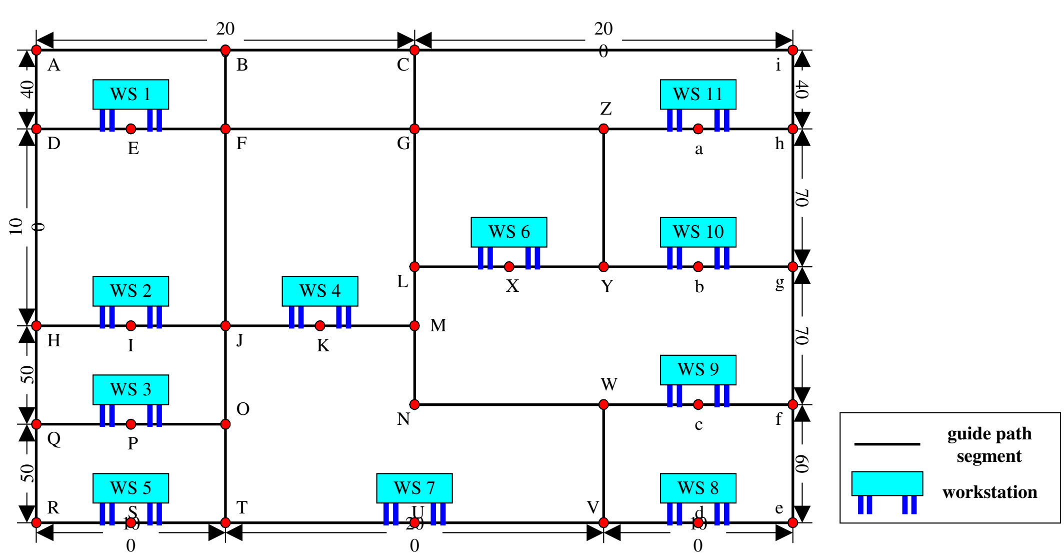

Fig. 11. Layout of workstations and AGV guide paths in the illustrated example.

from these collected parts it identifies those that it can pick up and delivery without the possibility of colliding with V(B). Again, if V(A) has successfully identified these parts, then from them V(A) will pick up one part and deliver it to its destination, say WS_k (where $0 \leqslant k < j$). Once V(A) has reached WS_k, it will drop the part at WS_k. V(A)'s location is then updated by setting $j = k$. On the other hand, if V(A) cannot find any parts that can meet the previous conditions, V(A) will move to WS_{j-1}. Once V(A) has arrived at WS_{j-1}, update V(A)'s position by making $j = j - 1$. A check is then required to see if V(A) has reached WS_0 (i.e. if $j = 0$ or not). If yes, V(A) has arrived at its own zone, Zone A; otherwise, V(A) will continue the heading-back routine.

5. An example

In this section, an example is given to illustrate the proposed design procedure. Simulations were conducted to test the performance of the proposed dynamic zone strategy.

5.1. A zone partition design example

The layout of our system is shown in Fig. 11. The guide paths are bi-directional and there are five part types in the system. Table 4 lists the processing routes of these parts and their production ratios. We assume that all vehicles move at a constant speed of 150 ft/min. Also, parts visiting the same workstations will go through the same operations. Table 5 lists the processing time distributions of these machines. The goal is to design an AGV system that can handle the transportation workload of 1000 parts in 11,000 min.

With the above information, we calculated the minimum required number of vehicles using the method presented in Section 3.1. Since the number obtained was three, we will divide the system into three zones. At the beginning, an initial zone partition design needs to be obtained. To come up with the initial design, three seed workstations were identified. Starting from each of these three seed workstations, the area of each zone was gradually expanded. Fig. 12 shows what the initial zone partition design looks like after the initial design stage. Using this initial zone partition design as the initial solution, the SA-based improvement procedure presented in Section 3.2.2 was then applied to look for an improved solution. Table 6 shows some of the intermediary results found during the improvement process. As is shown in Table 6, the best solution was found at Iteration 91. Compared with the initial solution, this improved solution has a better load balance result. Fig. 13 shows what this improved zone partition design looks like. Fig. 14 shows what the improved zone partition design looks like after the method introduced in Section 3.2.3 is applied to set up transfer stations between zones and assign the remaining critical segments to zones. The zone partition design shown in Fig. 14 was then used in the simulation experiments of Section 5.2.

Table 4
Processing routes and production ratios of parts in Part Set A

Part type	Processing route	Ratio
A	1–2–4–9–8–10–11	0.275
B	1–2–3–7–9–4–6–10–11	0.2
C	1–2–7–9–6–10–11	0.1
D	1–2–3–5–9–6–11	0.15
E	1–2–4–8–10–11	0.275

Table 5
Processing time distributions of machines

Workstation	Processing time (min)	Workstation	Processing time (min)
1	Normal(1,0.1)	7	Normal(2,0.2)
2	Normal(2,0.2)	8	Normal(3,0.1)
3	Normal(3,0.3)	9	Normal(3,0.3)
4	Normal(3,0.2)	10	Normal(3,0.1)
5	Normal(2,0.3)	11	Normal(2,0.1)
6	Normal(3,0.1)		

5.2. Simulation experiments

In this section, simulation experiments are conducted to test the performance of the proposed dynamic zone strategy. We would like to prove two things through these experiments. First, in Section 5.2.1, we would like to prove that the dynamic zone strategy can result in better performance than the fixed zone strategy. Second, in Section 5.2.2, we would like to prove that the system with the dynamic zone strategy is capable of adapting itself to changes occurring in the system.

5.2.1. Simulation experiment one

Two sets of simulation experiments are conducted. Table 7 summarizes the system attributes in each set. From Table 7, it is clear that each set has four experiments, each with a different mean inter-arrival time of parts. It is assumed that the inter-arrival times of parts follow exponential distributions. Four different mean inter-arrival times (5, 6, 7, and 8 min) are tested. Each experiment has thirty replications. The simulation time of each replication is 19,200 min. The warm-up time is 4800 min. The number of replications and the warm-up time are determined using the method described in Law and Kelton (1999). Products listed in Table 4 are made in these experiments. Three statistics are collected and calculated. They are average throughput, average Work-In-Process (WIP), and average flow time. Table 7 summarizes the average performance statistics obtained from the experiments. Table 8 summarizes the pair-t test results between the performances of the dynamic zone strategy and the fixed zone strategy in throughput, WIP, and flow time. From Tables 7 and 8, one can see that the proposed strategy has better throughput performance than the fixed zone strategy when the mean inter-arrival time is 5 or 6 min. Their throughput performance difference becomes less significant as the mean inter-arrival time increases. A greater mean inter-arrival time means a fewer number of parts will enter the system. Once the number of parts entering the system becomes so small that both systems are able to process all of them, the throughput performance between them will become less significant. The data shows that the dynamic zone strategy performs better than the fixed zone strategy in average WIP inventory and average flow time. From the results, it can be concluded that the dynamic zone strategy is indeed better than the fixed zone strategy in maximizing throughput, minimizing WIP inventory, and minimizing flow time.

5.2.2. Simulation experiment two

In this section, we would like to see if the dynamic zone strategy has the ability to adapt itself to changes occurring in the system. For this purpose, the following scenario is purposely created in the simulation experiment. In the experiment, two sets of parts will be made. The first set of parts, Part Set A, is shown in Table 4. The second set of parts, Part Set B, is shown in Table 9. As can seen, both Part Set A and Part Set B have the same part items. The only difference between them is the production ratios of parts. At the very beginning, parts in Part Set A are made in the system. Nothing changes in the system until 12,000 min has passed. At that moment, the system stops making parts in Part Set A and begins to make parts in Part Set B.

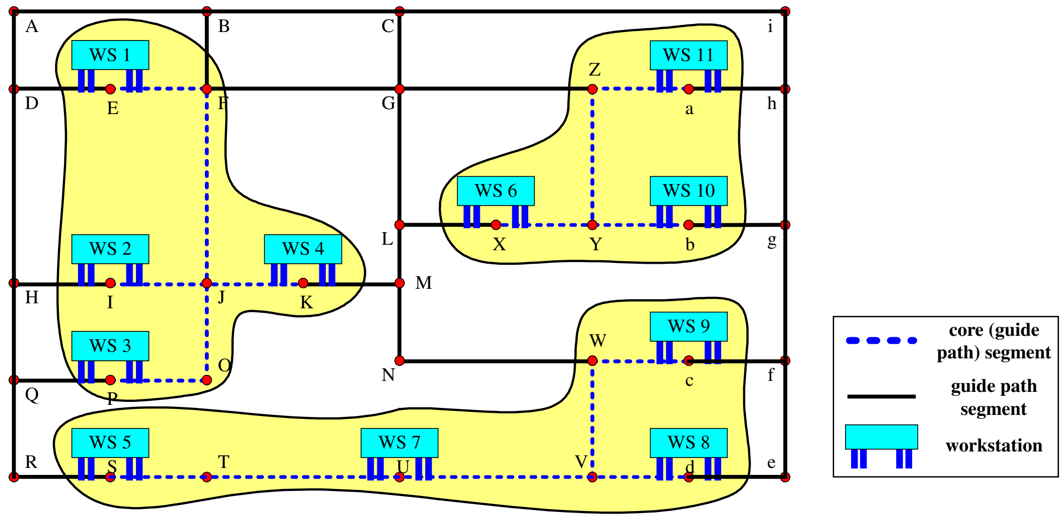

Fig. 12. Initial zone partition design.

Table 6
Intermediary solutions found during the improvement process

Iteration, (c)	Zone Partition Design found at the cth iteration, $p(c)$	$SV_{p(c)}$	(A)ccept or (R)eject	$SV_{\hat{p}}$
0	GW(1) = {1,2,3,4}, GW(2) = {5,7,8,9}, GW(3) = {6,10,11}	0.24319		0.24319
1	GW(1) = {2,3,4}, GW(2) = {5,7,8,9}, GW(3) = {1,6,10,11}	0.17775	A	0.17775
2	GW(1) = {2,3,4,7}, GW(2) = {5,8,9}, GW(3) = {1,6,10,11}	0.24170	R	0.17775
12	GW(1) = {1,2}, GW(2) = {5,7,8,9}, GW(3) = {3,4,6,10,11}	0.16028	A	0.16028
17	GW(1) = {1,3,6,7,11}, GW(2) = {5,8,9,10}, GW(3) = {2,4}	0.09969	A	0.09969
40	GW(1) = {6,10,11}, GW(2) = {3,5,7,8,9},GW(3) = {1,2,4}	0.07056	A	0.07056
49	GW(1) = {4,6,10,11}, GW(2) = {3,5,7,8,9} , GW(3) = {1,2}	0.05357	A	0.05357
91	GW(1) = {8,10,11}, GW(2) = {1,2 3,5}, GW(3) = {4,6,7,9}	0.03316	A	0.03316
99	GW(1)={3,7,8,11}, GW(2) = {1,2,5}, GW(3) = {4,6,9,10}	0.38840	R	0.03316
100	GW(1) = {3,7,8,10,11}, GW(2)={1,2,5}, GW(3)={4,6,9}	0.20714	R	0.03316

Table 7
Attributes and average performance values of each experiment set

Experiment set	Zone strategy	Mean inter-arrival time (min)	Average throughput		Average wip inventory		Average flow time (min)	
			Mean	SD	Mean	SD	Mean	SD
1	Dynamic zone	5	2388.13	33.38	502.41	210.34	2483.52	44.39068
		6	2391.93	23.42	103.27	211.76	606.62	36.67722
		7	2058.03	40.66	11.59	8.86	81.19	1.409994
		8	1807.30	39.89	7.67	2.46	61.09	0.446379
2	Fixed zone	5	2177.43	23.05	597.63	199.27	2979.01	47.32002
		6	2172.37	19.47	203.40	218.09	1216.47	38.60704
		7	2053.13	43.17	16.57	36.82	115.55	5.585247
		8	1806.70	40.38	8.16	4.18	64.92	0.668906

Table 8
The paired t-test between the dynamic zone strategy and the fixed zone strategy

Mean inter-arrival time (min)	p value		
	Average throughput	Average wip inventory	Average flow time
5	1.1E − 22***	5.24E − 22***	1.93E − 20***
6	1.49E − 28***	6.65E − 21***	6.21E − 21***
7	0.00968*	3.08E − 06***	2.13E − 06***
8	0.17135	1.93E − 08***	1.38E − 08***

Note: *, Significance at the 1% level; **, significance at the 0.1% level; and ***, significance at the 0.01% level or below. These notations will be used throughout this paper.

Four experiments – each with a different mean inter-arrival time of 5, 6, 7, and 8 min, respectively, are conducted. In each experiment, two simulations (one with the fixed zone strategy and the other with the dynamic zone strategy) are conducted. In all, eight simulations are conducted. The inter-arrival times of parts in these simulations all follow exponential distributions. Also, the warm-up time in each simulation is 4800 min. In each simulation, right after the warm-up time, the throughput is collected for every period of 720 min. In all, 20 periods are collected in each simulation. In other words, from Time 4800 to Time 12,000, the throughput performances of the first 10 periods (Period 1 – Period 10) are collected. In the first ten periods, parts in Part Set A enter the system. After that, 10 more throughput performances corresponding to Periods 11 through 20 are collected. In the second 10 periods, parts in Part Set B enter the system.

The throughput statistics collected from each simulation run are plotted in Fig. 15. There are four graphs in Fig. 15. These graphs show that the throughput performance of the fixed zone strategy or the dynamic zone strategy will eventually deteriorate once the

Table 9
Processing routes and production ratios of parts in Part Set B

Part type	Processing time	Ratio
A	1–2–4–9–8–10–11	0.05
B	1–2–3–7–9–4–6–10–11	0.3
C	1–2–7–9–6–10–11	0.3
D	1–2–3–5–9–6–11	0.3
E	1–2–4–8–10–11	0.05

system starts making parts in Part Set B. This is because the average processing time of parts in Part Set B is greater than the average processing time of parts in Part Set A. As a result, in the second 10 periods the system cannot produce as many parts as it did in the first 10 periods given the same amount of time. One also observes that in the first 10 periods the dynamic zone strategy performs better than the fixed zone strategy in all graphs. The dynamic zone strategy will eventually perform better than the fixed zone strategy in the second 10 periods. From Fig. 15a and Fig. 15b, one can see that in the first few periods of the second 10 periods the fixed zone strategy is better than the dynamic zone strategy. Further investigations show that many WIP parts have accumulated in these systems with the fixed zone strategy in Fig. 15a and Fig. 15b. As a result, for a while after systems have switched to making parts in Part Set B, the system with the fixed zone strategy in either Fig. 15a or Fig. 15b is still busy making parts in Part Set A, while the system with the dynamic zone strategy has already begun to make parts in Part Set B. Since on average parts of Part Set A require less processing time than parts of Part Set B, the system with the fixed zone strategy is capable of producing more parts than the system with the dynamic zone strategy in the first few periods of the second 10 periods.

The results in Fig. 15c and d indicate that the system with the dynamic zone strategy is more capable of adapting itself to changes in the system than the system with the fixed zone strategy. Fig. 15c and d show that both the dynamic zone strategy and the fixed zone strategy perform equally well in the first 10 periods. At the end of the first 10 periods, no significant amount of WIP parts have accumulated in either the system with the fixed zone strategy or the system with the dynamic zone strategy. In other words, both systems are able to complete parts entering them. This means that right before the moment when parts in Part Set B are about to be made, the shop floor situations in both sys-

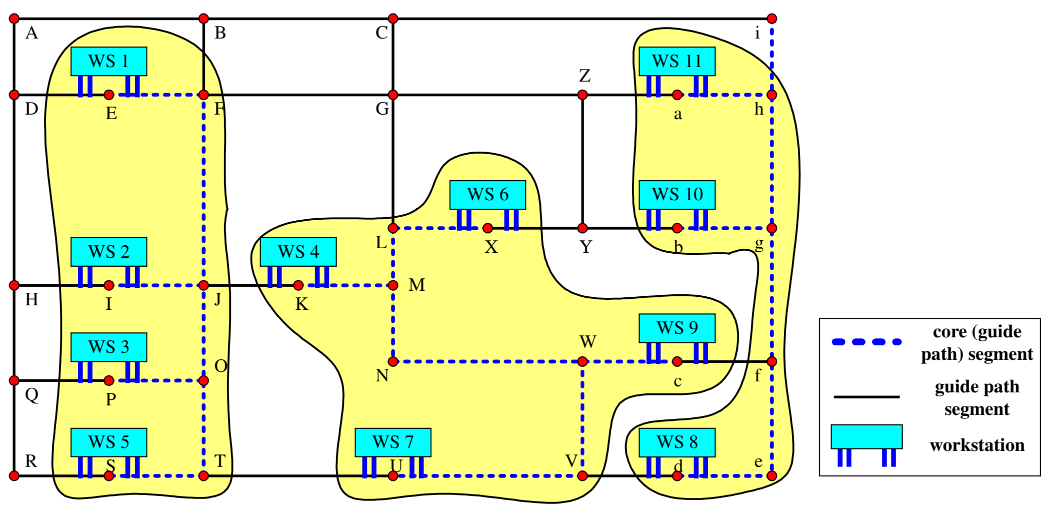

Fig. 13. Improved zone partition design.

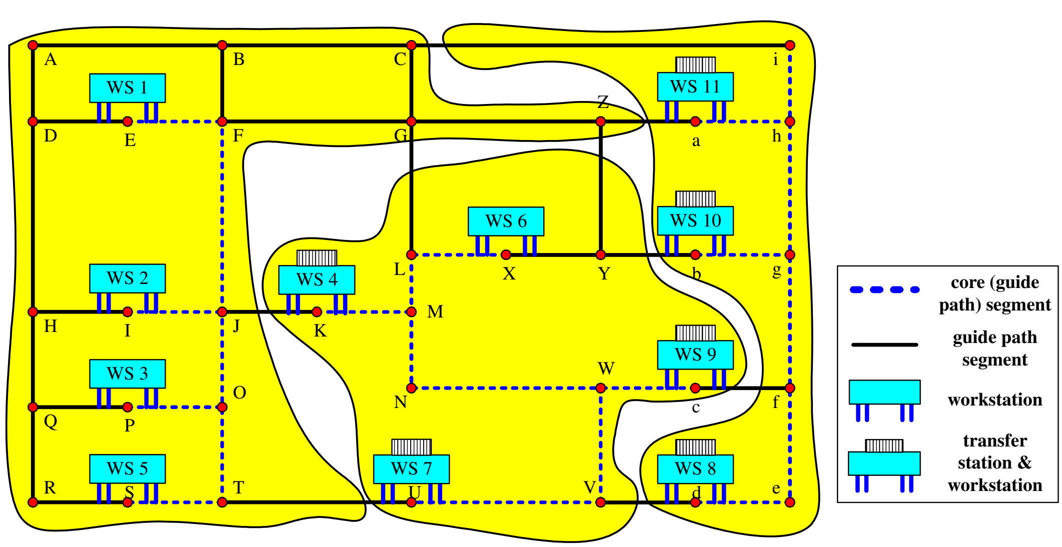

Fig. 14. Final zone partition design.

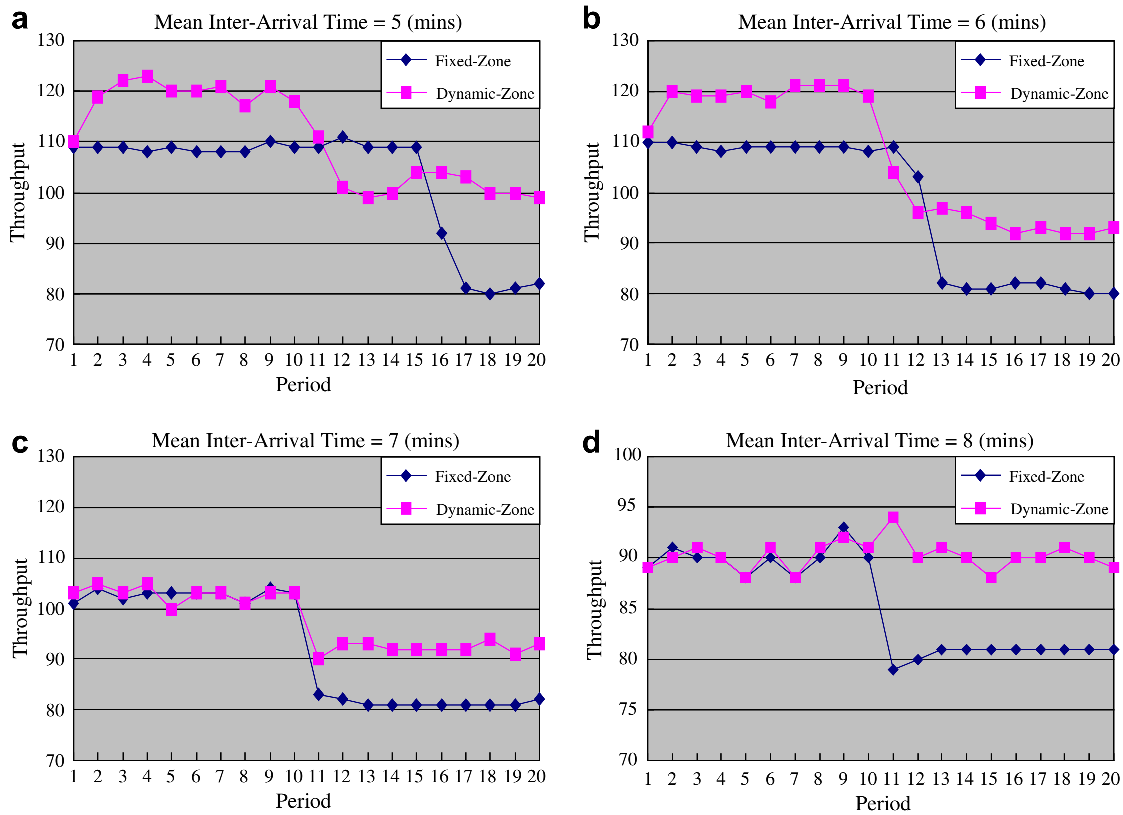

Fig. 15. Plot of throughput performances of fixed zone and dynamic zone strategies.

tems are similar. However, as both systems begin to produce parts in Part Set B, the throughput performance of the system with the fixed zone strategy deteriorates immediately while the system with the dynamic zone strategy can still maintain its performance. Since the shop floor situations of both systems are similar right before they start making parts in Part Set B, the only factor that can cause their performance difference in the second 10 periods is their zone control strategies. The simulation results show that the system with the dynamic zone strategy is indeed more capable of adapting itself to changes in the system than the system with the fixed zone strategy.

6. Summary and future research possibilities

In this paper, we propose a zone strategy for the purpose of vehicle collision prevention and load balance in an AGV system with a network guide path configuration. Unlike traditional fixed zone strategies, the proposed zone strategy allows every zone to change its area. The idea is to minimize the workload difference between vehicles of different zones. We discuss a unique problem incurred by the dynamic zone nature of our system. The problem is how to ensure that there will always be transfer stations between neighboring zones if they are allowed to change their area. We suggest two workstations designs to solve this problem. The proposed zone strategy contains a zone partition design method and a dynamic zone control method. The proposed zone partition design method contains three main stages. Its purpose is to come up with a zone partition design with a good load balance result. At the first stage, the number of zones is determined and an initial zone partition design is generated. At the second stage, the initial zone partition design is improved so that another zone partition design with a better load balance result can be obtained. Finally, at the last stage, transfer stations are set up and the remaining critical segments are assigned. There are three modules – Monitoring Module, Control Module, and Solution Module – in the proposed dynamic zone control method. The Solution Module contains two load-balancing methods – zone repartition and load sharing. These two methods are for different types of load imbalance problems. The Monitoring Module monitors the status of the system with two monitoring schemes – NM and IM. The system status is reported to the Control Module. The Control Module tells the Monitoring Module to adopt the correct monitoring scheme and directs the Solution Module to choose the appropriate methods for solving different load imbalance problems. The zone-repartition method is used if the load imbalance problem is a perpetual one, while the load-sharing method is used if the load imbalance problem is a short-lived one. Simulation experiments are conducted to show that the dynamic zone strategy is better than the fixed zone strategy. Two sets of experiments are conducted. Through the simulation experiment in Section 5.2.1, we have proved that the dynamic zone strategy performs better than the fixed zone strategy in various performance measures. Also, through the simulation experiment in Section 5.2.2, we can see that the system with the dynamic zone strategy is indeed more capable of adapting itself to changes in the systems than the system with the fixed zone strategy.

Finally, we would like to conclude this paper by presenting two future research possibilities. It is hoped that through the study of these problems one not only can improve the proposed dynamic zone strategy's performance, but also can increase its applicability in real world systems. First, in the moving-on routine of the load-sharing method, an AGV must follow a fixed route to assist another AGV in performing delivery operations. This moving-on routine will be discontinued if there is a chance of vehicle collision on this fixed route. Because the vehicle is not allowed to divert from the fixed route to avoid the collision problem, the vehicle may not be able to deliver the load as close to the load's destination as it can. In other words, the load-sharing effect is diminished due to the fixed route restriction. Similarly, in the heading-back routine of the load-sharing method, we also limit all pickup and delivery operations to a fixed route. The vehicle is not allowed to divert from the fixed route to look for other pickup and delivery operations to perform, even when there are no pickup or delivery operations on the fixed route. Again, the load-sharing effect is diminished due to this restriction. From the discussion above, it is likely a flexible route strategy may be better in enhancing the load-sharing effects. However, adopting a flexible route strategy will not be an easy task as it is in nature more difficult than the fixed route strategy. Developing a solution for implementing the flexible route strategy in the load-sharing method is the first future research problem suggested by us.

Second, since the focus of this study is to develop solutions that together can make the dynamic zone strategy work, the additional costs incurred by the proposed dynamic zone strategy method were not discussed. For example, additional costs can be incurred by the proposed method for giving each workstation the ability to serve as a transfer station. These costs include the cost of setting up auxiliary flow paths, the cost of adding one more pickup point and one more delivery point at each workstation, the cost of installing and operating control agents at workstations, etc. Before the proposed dynamic zone strategy can be implemented in a real world setting, a detailed cost analysis of the proposed dynamic zone strategy is necessary. This problem is the second future research problem suggested by us. The cost analysis result will allow us to understand the cost of the proposed dynamic zone strategy and help us discover potential problems.

References

Aiello, G., Enea, M., & Galante, G. (2002). An integrated approach to the facilities and material handling system design. *International Journal of Production Research, 40*(15), 4007–4017.

Bozer, Y. A., & Srinivasan, M. M. (1989). Tandem configurations for AGV systems offer simplicity and flexibility. *Material Handling System, 21*(2), 23–27.

Bozer, Y. A., & Srinivasan, M. M. (1991). Tandem configurations for automated guided vehicle systems and the analysis of single vehicle loops. *IIE Transactions, 23*(1), 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*(2), 173–191.

Chen, C. L., Lee, C. S. G., & McGillem, C. D. (1987). Task assignment and load balancing of autonomous vehicles in a flexible manufacturing system. *IEEE Journal of Robotics and Automation, 3*(6), 659–671.

Dotoli, M., & Fanti, M. P. (2004). Coloured timed Petri net model for real-time control of automated guided vehicle systems. *International Journal of Production Research, 42*(9), 1787–1814.

Egbelu, P. J. (1987). The use of non-simulation approaches in estimating vehicle requirements in an automated guided vehicle based transport system. *Material Flow, 4*(1), 17–32.

Egbelu, P. J., & Tanchoco, J. M. A. (1986). Potential for bi-directional guide-path for automated guided vehicles based systems. *International Journal of Production Research, 24*(5), 1075–1097.

Fanti, M. P. (2002). Event-based controller to avoid deadlock and collisions in zone-control AGVS. *International Journal of Production Research, 40*(6), 1453–1478.

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

Groover, M. P (1987). *Automation production systems and computer-integrated manufacturing*. Englewood Cliff, NJ: Prentice-Hall.

Ho, Y. C. (2000). A dynamic-zone strategy for vehicle-collision prevention and load balancing in an AGV system with a single-loop guide path. *Computers in Industry, 42*(2-3), 159–176.

Ho, Y. C., & Hsieh, P. F. (2004). A machine-to-loop assignment and layout design methodology for tandem AGV systems with multiple-load vehicles. *International Journal of Production Research, 42*(4), 801–832.

Ho, Y. C., & Moodie, C. L. (1998). Machine layout with a linear flow path in an automated manufacturing system. *Journal of Manufacturing Systems, 17*(1), 1–22.

Jang, J. J., Suh, J. D., & Ferreira, P. M. (2001). An AGV routing policy reflecting the current and future state of semiconductor and LCD production lines. *International Journal of Production Research, 39*(17), 3901–3921.

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

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

Kim, C. W., & Tanchoco, J. M. A. (1991). Conflict-free shortest-time bi-directional AGV routing. *International Journal of Production Research, 29*(12), 2377–2391.

Kim, C. W., & Tanchoco, J. M. A. (1993a). Operational control of a bi-directional automated guided vehicle system. *International Journal of Production Research, 31*(9), 2123–2138.

Kim, K. H., & Tanchoco, J. M. A. (1993b). Economical design of material flow paths. *International Journal of Production Research, 31*(6), 1387–1407.

Koff, G. A. (1985). The basics of AGVS. In Proceedings of Promat 85, Material Handling Institute, Feb. 1–18.

Kouvelis, P., Chiang, W. C., & Fitzsimmons, J. (1992). Simulated annealing for machine layout problems in the presence of zoning constraints. *European Journal of Operational Research, 57*(2), 203–223.

Krishnamurthy, N. N. (1990). Modeling blocking in automated guided vehicle systems. Ph.D. thesis, State University of New York at Buffalo, USA.

Kusiak, A., & Cyrus, J. P. (1985). Routing and scheduling of automated guided vehicles. In H.-J. Bullinger & H. J. Warnecke (Eds.), *Toward the factory of the future* (pp. 247–251). Berlin: Springer.

Law, K. M., & Kelton, W. D. (1999). *Simulation modeling and analysis*. Boston: McGraw-Hill.

Liu, F. H., & Hung, P. C. (2001). Real-time deadlock-free control strategy for single multi-load automated guided vehicle on a job shop manufacturing system. *International Journal of Production Research, 39*(7), 1323–1342.

Maxwell, W. L., & Muckstadt, J. A. (1982). Design of automatic guided vehicle systems. *IIE Transactions, 14*(2), 114–124.

Rajotia, S., Shanker, K., & Batra, J. L. (1998). An heuristic for configuring a mixed uni/bidirectional flow path for an AGV system. *International Journal of Production Research, 36*(7), 1779–1799.

Reveliotis, S. A. (2000). Conflict resolution in AGV systems. *IIE Transactions, 32*(7), 647–659.

Sarker, B. R., & Gurav, S. S. (2005). Route planning for automated guided vehicles in a manufacturing facility. *International Journal of Production Research, 43*(21), 4659–4683.

Shalaby, M. A., El Mekkawy, T. Y., & Fahmy, S. A. (2006). Zones formation algorithm in tandem AGV systems: a comparative study. *International Journal of Production Research, 44*(3), 505–521.

Sharp, G. P., & Liu, F. H. (1990). An analytical method for configuring fixed-path, closed-loop material handling systems. *International Journal of Production Research, 28*(4), 757–783.

Singh, S. P., & Tiwari, M. K. (2002). Intelligent agent framework to determine the optimal conflict-free path for an automated guided vehicle system. *International Journal of Production Research, 40*(16), 4195–4223.

Sinriech, D., & Tanchoco, J. M. A. (1996). The segmented bi-directional single-loop topology for material flow system. *IIE Transactions, 28*(1), 40–54.

Srivastava, S. C., Choudhary, A. K., Kumar, S., & Tiwari, M. K. (2007). Development of an intelligent agent-based AGV controller for a flexible manufacturing system. *International Journal of Advanced Manufacturing Technology*. doi:10.1007/s00170-006-0892-9.

Taghaboni, F. (1989). Scheduling and control of manufacturing systems with critical material handling. Ph.D. thesis, Purdue University, West Lafayette, IN, USA.

Tavakkoli-Moghaddam, R., Aryanezhad, M. B., Kazemipoor, H., & Salehipour, A. (2007). Partitioning machines in tandem AGV systems based on "balanced flow strategy" by simulated annealing. *International Journal of Advanced Manufacturing Technology*. doi:10.1007/s00170-007-1094-9.

Vis, I. F. A. (2006). Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research, 170*(3), 677–709.

Wallace, A. (2007). Multi-agent negotiation strategies utilizing heuristics for the flow of AGVs. *International Journal of Production Research, 45*(2), 309–322.

Yoo, J. W., ·Sim, E. S., ·Cao, C., & Park, J. W. (2005). An algorithm for deadlock avoidance in an AGV System. *International Journal of Advanced Manufacturing Technology, 26*(5–6), 659–668.