Journal of Industrial Engineering International *Islamic Azad University, South Tehran Branch*
January 2009, Vol. 5, No. 8, 37-51

Solving the tandem AGV network design problem using tabu search: Cases of maximum workload and workload balance with fixed and non-fixed number of loops

Elnaz Miandoabchi
Industrial Engineering Department, Amirkabir University of Technology, Tehran, Iran

Reza Zanjirani Farahani*
Associate Professor, Industrial Engineering Department, Amirkabir University of Technology, Tehran, Iran

Abstract

A tandem AGV configuration connects all cells of a manufacturing area by means of non-overlapping, single-vehicle closed loops. Each loop has at least one additional P/D station, provided as an interface between adjacent loops. This study describes the development of three tabu search algorithms for the design of tandem AGV systems. The first algorithm was developed based on the basic definition of a tandem network. The second and third algorithms, consider no preset number of loops and try to evenly distribute workload among loops by using workload balance as their objective functions. They generate different design scenarios for the tandem network, which can be evaluated and selected using a multi-attribute objective function. The first algorithm and the partitioning algorithm presented by Bozer and Srinivasan are compared for randomly generated problems. Results show that for large-scale problems, the partitioning algorithm often leads to infeasible configurations with crossed loops in spite of its shorter running time. However, the newly developed algorithm avoids infeasible configurations and often yields better objective function values.

Keywords: AGV; Tandem configuration; Tabu search

1. Introduction

One of the most important decisions in facility design is the design of handling systems. Material handling operations costs compose nearly 20 to 50 percent of the overall operational costs [44]. An automated guided vehicle (AGV) is a driverless vehicle used for the transportation of goods and materials within a production plant partitioned into cells (or departments), usually by following a wire guide-path. Among some basic issues in designing AGV systems is the guide path design. The problem of guide path design for AGV system is not new. A number of algorithms for AGV guide path design have been developed over the past 20 years [3]. The AGV guide path configurations discussed in literature include *Conventional/Traditional* [18,19,23,24,27,28,36,37,42,45,52,53], *Tandem* [10,11,30], *Single loop* [2,4,5,6,29,38,43,48,49,50,51], *Bi-directional shortest path* [13,26,32] and *Segmented flow* [7,39,40,41]. Vis [15] has reviewed the literature related to the design and control issues of AGV systems in manufacturing, distribution, transshipment and transportation systems.

The tandem configuration, which is the concern of this paper, was introduced by Bozer and Srinivasan [9, 10] and is based on the "divide-and-conquer" principle. A tandem configuration is obtained by partitioning all the workstations into single-vehicle, non-overlapping zones. Additional pick-up/delivery (P/D) points are provided between adjacent zones to serve as transfer points. This configuration offers some advantages such as eliminating blocking and congestion, simplicity of control, and flexibility due to system modularity. It also has some disadvantages including the need for handling a load by two or more vehicles, and thus longer load movement times, extra floor space and cost requirements, resulted by the use of additional P/D points and conveyors.

Tandem paths were initially proposed by Bozer and Srinivasan [9,10] who presented an analytical model

* *Corresponding author. E-mail: farahan@aut.ac.ir*

to compute the workload of a single loop. They developed a heuristic method for the partitioning the stations in loops [11]. As pointed out by Faraji and Batta [16], tandem AGV systems have another disadvantage: they yield a loss of efficiency due to the transfer stations - this efficiency loss is also the subject of another paper by Faraji and Batta [16].

Hsieh and Sha [21] proposed a design process for the concurrent design of machine layout and tandem routes. Liu and Chen [31] suggested a divided AGV system which is similar to a tandem AGV system in that there is one AGV in each divided zone. However, the difference is that the path of each zone is allowed to cross over the path of another. Aarab *et al.* [1] used hierarchical clustering and tabu search to determine tandem routes in a block layout. Yu and Egbelu [47] presented a heuristic partitioning algorithm for a tandem AGV system, based on the concept of variable path routing. Kim *et al.* [25] proposed an analytical model to design a tandem AGV system with multi-load AGVs. Later Bozer and Lee [12] tried to eliminate the conveyers by using an existing station as a transfer point. Ventura and Lee [46] studied tandem configurations with the possibility of using more than one AGV in the loops. Huang [22] proposed a new design concept of tandem AGV based on using a transportation center to link the transfer points in loops.

To test the viability of tandem configurations, Farling *et al.* [17] have performed a simulation study to evaluate the performance of three AGV configurations, namely traditional (parallel unidirectional flows), the tandem flow-path and the tandem loop, under various conditions. In a tandem loop flow-path there exists an express loop which connects all loops [34]. Ross *et al.* [34] and Choi *et al.* [14] conducted some experiments to compare a tandem AGV system with a conventional AGV system. Ho and Hsieh [20] proposed a design methodology for tandem AGV systems with multiple-load vehicles. Simulation and SA were applied to solve the problems.

The application of metaheuristic algorithms in designing tandem paths is limited to only two papers. Most of the previous studies have proposed heuristic algorithms for the design of tandem routes. Only in [1] and [20], tabu search and simulated annealing have been applied as design procedures. While in [1] hierarchical clustering was applied to partition departments into loops, tabu search was only used to find a single loop path for each zone. Thus, the application of metaheuristics as design procedures in tandem systems has not yet been fully considered.

The aim of this paper is to develop three Tabu search based algorithms for the design of a tandem configuration path network. The initial tandem configuration problem is the assignment of stations to single AGV loops, without allowing the paths of the loops to cross each other. The first algorithm was developed based on the problem definitions proposed by Bozer and Srinivasan [9,10], in which the number of loops is given as input, and the objective function is minimizing the maximum workload of the system. We have proposed another tandem configuration problem in which the number of loops is unknown and the objective is maximizing the balance between the workloads of the loops. The first proposed Tabu Search algorithm has been compared to the partitioning heuristic algorithm of Bozer and Srinivasan [11], referred to as the base algorithm, using randomly generated test problems. The latter two algorithms have been solved for the test problems and some analyses were done based on the test results. This paper is organized as follows: a brief description of the problem and its assumptions are presented in Section 1, the developed algorithms and their descriptions are discussed in Section 3, and the computational results are reported in Section 4, finally conclusions are presented in Section 5.

2. Problem Definition

Tandem AGV systems which were first introduced by Bozer and Srinivasan [9,10], was mainly defined and developed for the manufacturing environment. However, it can be used both in warehousing and manufacturing. Bozer and Srinivasan [11] defined the system on a grid layout where each workstation is presented as a single point and may represent a machine, or a group of machines, such as a cell or a department.

The problem of configuring a tandem AGV system consists of partitioning a set of N workstations into several independent single AGV loops (zones). Additional P/D stations called transfer points are introduced to provide an interface between adjacent loops. Transfer points are connected to each other by conveyors. Figure 1 illustrates a typical tandem configuration.

The workstations of a layout are partitioned in such a way that each station is assigned to only one loop, the workload of the AGVs associated with the material flow within and between the loops does not exceed the AGV capacity, and the workload is evenly distributed among all loops. The workload factor of each AGV, denoted by ω, is the proportion of time a vehicle is busy, either loaded or empty. This is the building block of the system and must be calculated for each loop.

According to the assumptions made by Bozer and Srinivasan [11], there are two types of workstations, the first type is input/output station and the second type is process station where the actual processing takes place. Transfer points also are considered as I/O stations.

Every station has an I/O queue. A bidirectional single load AGV is used in each loop. When loaded, the AGV follows the shortest path to the destination station, and when empty it uses the FEFS (first encountered first served) empty vehicle dispatching rule, which will never leave the AGV idle. When a tandem AGV system was proposed for the first time, the FEFS was the only dispatching rule suggested [8].

Additional limitations are as follows: intersections and overlaps are forbidden between loops; the number of loops must be at least 2; the number of loops can be provided as input, or they can be obtained through the design process.

3. The developed algorithms

In this section, the developed algorithms based on TS metaheuristic are described. All the three algorithms try to find the best configuration of the tandem loops, based on some criteria. Table 1 clarifies the differences between the three algorithms.

- The first algorithm (OTS): This algorithm tries to find a configuration that minimizes the maximum workload of the system (i.e. workload among loops), given desired number of loops by the decision maker. The definitions are exactly based on the work by [11].
- The second and third algorithms (MSETS and MADTS): The algorithms try to find a configuration that minimizes the mean standard square error of workloads (MSE) or mean absolute error of workloads (MAD) in order to balance the workload among loops. The desired number of loops is not known prior to the solution. The algorithms solve the problem for every feasible loop number and find the best configuration for each case. Thus for every problem, a number of scenarios are produced based on each loop number.

In this paper, the algorithms will be denoted by OTS, MSETS and MADTS. In Section 3.1 common characteristics of the algorithms are presented and the remaining of the section is dedicated to the detailed description of each algorithm.

3.1. General definitions

In this section, common definitions of the three algorithms are presented.

3.1.1. Feasibility conditions

A solution is called feasible if (1) no overlap or intersection exists between loops, (2) each station is assigned only to one loop, and (3) the workloads are less than 1 (or a reasonable value less than 1). The algorithm proposed by Bozer and Srinivasan [11], does not have any specific mechanism to check the overlaps among the loops. Our algorithms check the intersections of the loops in evaluating any move. For the sake of simplicity, the initial routes of loops are considered as the Euclidean traveling salesman route of the stations in the loop. Figure 2 shows a typical infeasibility in the presence of overlapping loops.

3.1.2. Neighborhood structure

The neighborhood of a solution is simply obtained by removing a station from one loop and adding it to another, provided that it does not create any intersections and the workloads of the loops do not exceed 1. A current solution corresponding to a partition of the set of workstations in L loops can be represented as follows:

$$S = \{P_1, ..., P_i, ..., P_L\} \qquad i = 1, ..., L \tag{1}$$

Consider the station s from the set of stations in loop P_i, where $st \in P_i$. Also consider solution S' in the neighborhood of solution S:

$$S' = \{P'_1, ..., P'_i, ... P'_L\} \qquad i = 1, ..., L \tag{2}$$

Solution S' is obtained by moving station s from loop P_i to loop P_j in solution S. In other words:

$$P'_i = P_i - \{s\} \tag{3}$$

$$P'_j = P_j \cup \{s\} \quad j \neq i \tag{4}$$

The feasible move m_{ij} is characterized by the transmission of station s from loop i to loop j, subject to the workload constraint. The neighborhood of S is the set of all feasible solutions that can be reached from S by applying moves m_{ij}. A typical neighborhood structure is shown in Figure 3. Figure 3(a) shows the possible moves for station 12, and Figure 3(b) shows the new solutions resulted by the possible moves of station 12 to 3 loops.

3.1.3. Generation of feasible solutions

The procedure of generating initial solutions consists of three phases:

a. Clustering the stations into L groups by means of k-means clustering method,

b. If necessary, removing workload infeasibilities,

c. Reducing the number of singleton stations.

In the first stage, a k-means clustering method is applied [35]. Applying the k-means clustering method to generate initial partitions (an initial partitioning) ensures that no intersections will occur among the created loops, since distance is used as the closeness measure.On the other hand, the workstations in a loop should be reasonably close to each other, so that unnecessary vehicle trips are avoided. The initial cluster centers are chosen randomly and the rectilinear distance is used as the closeness measure, due to the rectilinear shape of the final routes of the loops. The resulting clusters are then checked for workload feasibility. In case of infeasibility, a simple search method is used to reduce the workload of infeasible loops by moving some stations, following the defined neighborhood structure. The search method will stop as soon as the stop criterion is reached. In the last phase, the configuration is checked for the presence of singleton loops and if more than one is present, another procedure is employed to reduce the number of singletons by adding a station from adjacent loops to them.

3.1.4. Tabu restrictions and aspiration criterion

The algorithms use a fixed size tabu list. Due to the definition of a move, once a station is moved from one loop to another, it cannot be re-added to the original loop again in the next θ iterations. However, when a potential tabu move leads to a solution better than the best solution found so far, its tabu status is revoked.

3.1.5. Diversification

Diversification is a commonly used strategy in TS. It is used to prevent the search mechanism from falling into potential local minima. In our implementation, when no feasible move exists, the tabu list is emptied and the search restarts from the best-known solution.

3.1.6. Termination criterion

The termination criterion is set as the maximum number of iterations performed since the best solution was changed.

3.1.7. Allowing infeasible solutions

In some cases, all possible moves lead to increased workloads. In such situations, the search is allowed to proceed outside the feasible space, in the hope of reaching feasibility again at a later stage.

3.1.8. Controlling singleton stations

A control mechanism is applied in order to limit the number of singleton stations in system. This is done in two stages: in the generation of the initial solution, and in the main body of the algorithms. After generating an initial feasible solution, the number of singleton stations is checked and if there is more than one, another subroutine is applied to reduce this number to one.

Another control is applied as a condition for accepting the current solution as the best solution. The number of singleton stations in the current solution is checked and if it is more than one, the same procedure is applied to reduce it. If the number of stations in the best solution is more than one, and the current solution has fewer singleton stations, it is accepted as the best solution even if it has a worse objective function. In other words, a solution with fewer singleton stations is preferred when the number of singleton stations is more than one. Otherwise, if this number is less than or equal to one in the best solution found so far, the current solution is accepted as the best solution if its number of singleton stations is less than or equal to one, and it has a better objective function.

3.2. Development of the first TS algorithm (OTS)

In this section the Tabu Search algorithm, developed based on the minimax of workload as the objective function with fixed number of loops, will be described.

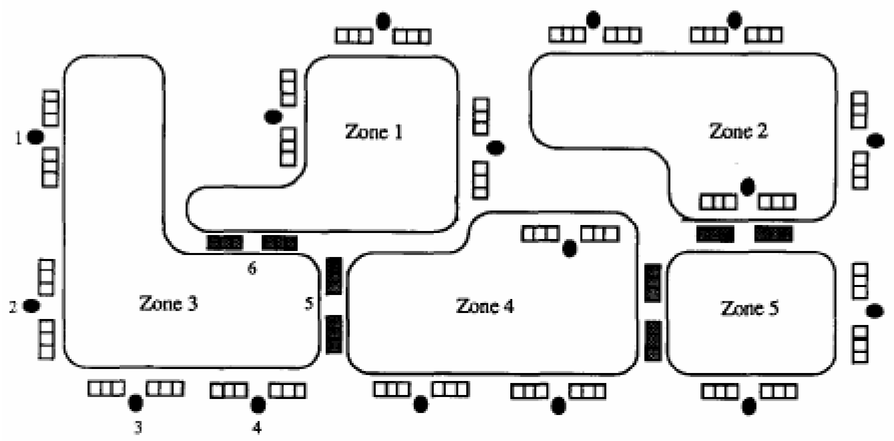


Figure 1. A typical tandem configuration (Bozer and Srinivasan, 1992).

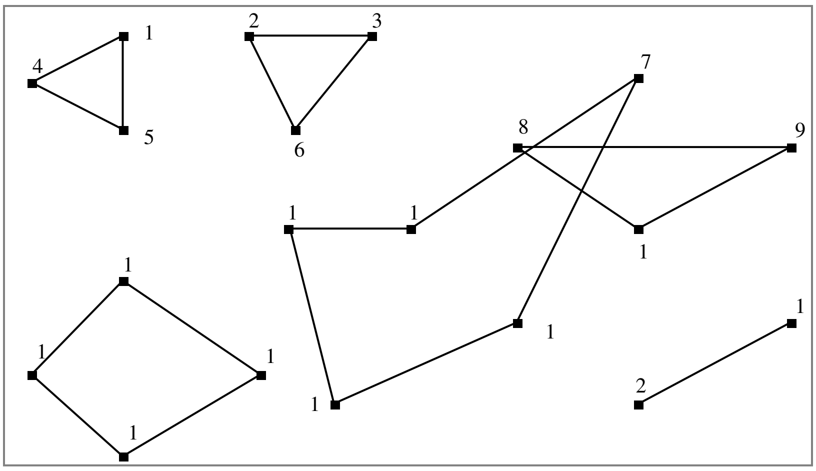
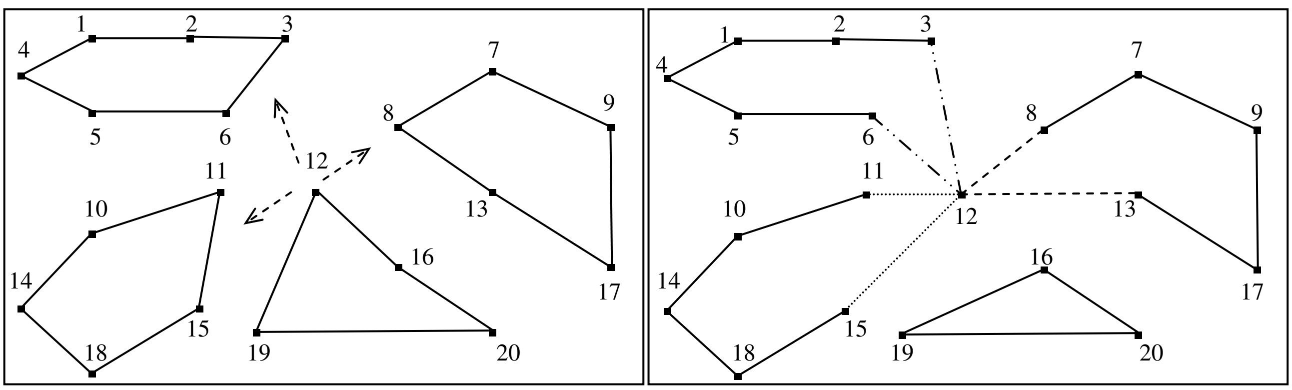


Figure 2. A typical infeasible solution.



(a) (b)

Figure 3. A typical neighborhood structure.

Table 1. Our TS based algorithms compared to each other.

Algorithm	Number of loops as input	Objective Function	Output
OTS	Known	Minimum of maximum workload	The sole optimum Configuration
MSETS	Unknown	Minimum of mean squared error of workloads	Optimum Configuration for each number of loops
MADTS	Unknown	Minimum of mean absolute error of workloads	Optimum Configuration for each number of loops

3.2.1. Objective function and evaluation of the neighborhood

We choose the objective function of the integer linear programming model [11] as the objective function, which is the minimization of maximum workload of all loops. Thus, the basis of the search procedure is to select a loop with maximum workload and try to reduce its workload. Reduction in workload is obtained by moving its stations to another loop, which causes an increase in the workload of the latter. The best move is the one leading to the largest decrease in the workload of the selected loop and the smallest increase in the workload of the second loop. Thus, a simple phrase can be used to summarize the evaluation criterion: the workload reduction of the selected loop minus the workload increase of the second loop.

3.2.2. Definition and selection of moves

According to the neighborhood definition, a move means moving a station from one loop to another one. Based on the minimax objective function, the evaluation of possible moves in each of the iterations is restricted to the loop with the maximum workload.

Investigation showed that it is not needed to check for moving all stations to all destination loops, since it lead to many overlapped loops and unnecessary calculations. As a rule of thumb and in order to improve efficiency, we only consider the closest adjacent six or seven loops, according to the distance of their geometrical centers to the one of the selected loop. Also, we found that it suffices to check only four stations with the largest workload decreases as candidates for being removed from the loop.

The inclusion of a station in another loop is allowed only if it does not lead to an infeasible solution. A destination loop workload threshold $\overline{\varpi}$ has been set to select the moves leading to the best possible solutions, by limiting the increase of the workload of the destination loop. When evaluating possible moves, the moves that satisfy the threshold conditions are first considered. In the absence of such moves, this restriction is not taken into account and all possible moves are evaluated.

3.2.3. Number of loops

Similar to the algorithm of Bozer and Srinivasan [11], the number of loops is given as an input.

3.2.4. Detailed description of the OTS algorithm

The detailed description of the OTS algorithm is as follows:

Phase 1.

1. Generate an initial solution S_0, and set $S := S^*$, $f(S) := f(S^*)$;
    - Cluster the stations into L loops.
    - If necessary, resolve workload infeasibilities.
    - Reduce the number of singleton stations.
2. Set $\tau := 0$ (τ is the (counter) number of the last iteration with an improvement in the objective function);
3. Set iteration counter to zero: $t := 0$.

Phase 2.

While $t - \tau < maxt$, repeat:

1. $t := t + 1$;
2. Choose the loop with maximum workload (P_O);

3. Define the possible candidate moves:
   - Choose some of the nearest loops to P_O, based on the defined rules.
   - Choose candidate stations to be moved from P_O, based on the defined rules.
   - Consider any move that doesn't lead to path overlap.
   - Exclude any move in Tabu list, except the one that satisfies the aspiration criteria.
4. If there are possible moves:
   - Compute evaluation criteria "workload reduction of loop P_O – workload increase of destination loop" for each candidate move.
   - If possible, select the move with minimum computed value, considering destination loop workload $\overline{\omega}$. Else, select the move without considering the threshold.
   - Update the tabu list (TL) by adding the moved station and the destination loop to the list.
5. If $\sigma(S) > 1$ ($\sigma(S)$ is the number of singleton stations in solution S), attempt to reduce it.
6. If $\sigma(S) > 1$ but $\sigma(S) < \sigma(S^*)$, then set $S := S^*$, $f(S) := f(S^*)$;
   Else if $\sigma(S^*) \leq 1$, $\sigma(S) \leq 1$ and $f(S) < f(S^*)$, then set $S := S^*$, $f(S) := f(S^*)$.
7. If no possible move exists, then use diversification strategy:
   - Set $S := S^*$, $f(S) := f(S^*)$.
   - Empty the tabu list ($TL = \emptyset$).

3.3. Development of the second and third TS algorithms

In this section the MSETS and MADTS algorithms, developed based on workload balance and variable number of loops, will be described.

3.3.1. Objective Function and Move Selection

As mentioned before, the object is to maximize the balance between the workloads of the loops. Here, the mean squared error (MSE) and the mean absolute error (MAD) of workloads have been selected as two objective functions. The basis of the search procedure is to select a loop with the maximum positive workload deviation from the average (i.e. the loop with maximum workload). This strategy ensures that while trying to reduce workload deviation, a reduction in the maximum workload of the system will occur. Similar to the OTS algorithm, only some of the moves are evaluated. Except that no workload threshold is used here. Here, the best move is the one resulting in the smallest objective function value.

3.3.2. Number of loops

In these two algorithms, the number of loops is considered to be variable and unknown. The algorithms solve the problem for all loop numbers beginning from the maximum possible number of loops ([N/2]) until a loop number of 2, or a maximum workload limit (i.e. 0.7 as in [11]) is reached. Although the main objective function is not the minimization of maximum workload, it is still a major indicator of the workload feasibility. The resulting solutions will be scenarios containing the best workload balance (MSE or MAD) obtained for each loop number, and their maximum workload values.

3.3.3. Detailed description of the MSE and MAD algorithms

The detailed description of the algorithms is as follows:

1. Set L:= [N/2], mw :=0 (the maximum workload of the system);
2. While $mw \leq 0.7$ or $L \geq 2$, repeat:

2.1. Generate an initial solution S_0, and set $S := S^*$, $f(S) := f(S^*)$:
   - Cluster the stations into L loops.
   - If necessary, resolve workload infeasibilities.
   - Reduce the number of singleton stations.

2.2. Set τ :=0 (τ is the number of the last iteration with an improvement in the objective function);

2.3. Set iteration counter to zero: t:=0;

2.4. While $t - \tau < maxt$, repeat :

2.4.1. $t := t + 1$;

2.4.2. Choose the loop with the maximum workload (P_O);

2.4.3. Define the possible candidate moves:

- choose some of the nearest loops to P_O, based on the defined rules.
- Choose candidate stations to be moved from P_O, based on the defined rules.
- Consider any move that doesn't lead to path overlap.
- Exclude any move in Tabu list, except the one that satisfies the Aspiration Criteria.

2.4.4. If there are possible moves:

- For each candidate move calculate the $f(S)$ (workload MSE or MAD).
- Choose the move with minimum $f(S)$ value and update $f(S)$, S.
- Update the tabu list (TL) by adding the moved station and the destination loop.

2.4.5. If $\sigma(S) > 1$ ($\sigma(S)$ is the number of singleton stations in solution S), attempt to reduce it;

2.4.6. If $\sigma(S) > 1$ but $\sigma(S) < \sigma(S^*)$, then set $S := S^*$, $f(S) := f(S^*)$;

Else if $\sigma(S^*) \leq 1$, $\sigma(S) \leq 1$ and $f(S) < f(S^*)$, then set $S := S^*$, $f(S) := f(S^*)$.

2.4.7. If no possible move exists, then use diversification strategy:

- Set $S := S^*$, $f(S) := f(S^*)$.
- Empty the tabu list ($TL = \emptyset$).

2.5. $L := L - 1$;

2.6. Save S^*, $f(S^*)$, L, mw of the obtained solution.

4. Computational results

Since no benchmark problem instances exist for the tandem configuration problem, some randomly generated problems were used to test the algorithms. In addition, the two example problems presented and solved in [11] were also solved by our OTS algorithm.

4.1. Test problems

The test problems were generated for five types of grid layouts including 10, 20, 30, 40 and 50 stations. For each size of layout, three types of From-To charts were randomly built for densities 0.2, 0.25 and 0.5. Flow values (units/hour) were chosen randomly between 0.05 and 0.3. The AGV specifications were obtained from Bozer and Srinivasan [11]. The speed of the AGV (empty or loaded) and the time required to pick up or deliver a load were set to 15 grid units/minute and 0.2 minutes, respectively. For each of the 15 problems, four random instances were generated. In total, 60 problems were solved.

4.2. Computational results for OTS algorithm

Each of the test problems was solved for three levels L (number of loops). The L values were initially derived from the base algorithm and were then applied in the solution procedure of the TS algorithm. These L values are:

L_{max} : the maximum possible value for L. Assuming that each station can form a feasible loop with at least one of its adjacent stations, this value will be equal to $[N/2]$;

L_{min} : the minimum value possible for L, at which the objective function doest not exceed 0.7 (the selected threshold for workload in the base algorithm);

$L_{average}$: the mean of L_{max} and L_{min} .

Based on the suggestions provided by Bozer and Srinivasan (1992), for reducing the run time of the integer linear programming model, an estimated threshold z_H was used to eliminate unnecessary loops. This threshold was set to 0.7 for $L_{min,}$ and was obtained from the average and maximum workloads of loops with two or three stations for L_{max}, and finally $L_{average}$ was set to a value between the z_H of L_{max} and L_{min}, or the average workload of loops with sizes equal to $[N/L]$. Due to the fact that at most one singleton station is assumed to be allowed in the final configuration obtained from the TS algorithm, the same assumption was made for the base algorithm as well. Tests were carried out on a 2.00 GHz Intel Pentium 4, with 256 MB RAM.

Table 2. Summary of the test results of OTS.

N No. of stations	d Density of FT chart	L No. of loops	No. of infeasible solutions of the base alg.	No. of better solutions		No. of worse solutions		No. of equal solutions		Average deviation of OTS objective function from the base alg.		Best improvement in the objective function of OTS	Average run time (seconds)	
				No.	%	No.	%	No.	%	Positive deviation	Negative deviation		Base alg.	OTS
10	0.2	max	0	0	0%	0	0%	11	100%	-	-	-	212.9	15.7
		average	0	4	44%	0	0%	5	56%	-	-6.1%	-9.4%	233.7	24.5
		min	1	4	67%	0	0%	2	33%	-	-0.4%	-0.5%	214.2	57
	0.25	max	0	2	20%	1	10%	7	70%	0.7%	-14.7%	-15.4%	281.1	20.1
		average	1	0	0%	0	10%	7	100%	-	-	-	298.8	23
		min	0	0	0%	0	0%	6	100%	-	-	-	282.3	65.9
	0.5	max	0	0	0%	0	0%	12	100%	-	-	-	75.7	17.4
		average	0	0	0%	2	20%	8	80%	6.8%	-	-	81	25.1
		min	0	2	29%	0	0%	5	71%	-	-2.8%	-2.8%	73	66.1
20	0.2	max	2	2	17%	1	8%	9	75%	3.1%	-0.6%	-6.0%	160.1	52.8
		average	0	11	92%	1	8%	0	0%	2.2%	-6.7%	-13.9%	162.3	144.8
		min	0	4	67%	0	0%	2	33%	-	-6.3%	-10.8%	171.2	363
	0.25	max	1	3	25%	1	8%	8	67%	2.2%	-1.0%	-1.0%	109.7	60.2
		average	0	2	17%	1	8%	9	75%	3.8%	-1.0%	-1.0%	109.1	107.1
		min	1	5	45%	1	9%	5	45%	6.7%	-5.1%	-6.5%	116.9	239.1
	0.5	max	1	6	50%	1	8%	5	42%	0.2%	-11.4%	-17.1%	36.4	61
		average	2	6	50%	0	0%	6	50%	-	-6.6%	-11.0%	37.3	91.6
		min	1	8	67%	3	25%	1	8%	5.7%	-8.9%	-17.4%	40.5	198.7
30	0.2	max	2	9	75%	0	0%	2	25%	-	-4.5%	-14.0%	166	92.7
		average	3	10	83%	1	8%	1	8%	3.1%	-4.9%	-12.5%	145.3	137.8
		min	2	11	92%	0	0%	1	8%	-	-8.0%	-19.0%	165.2	309.7
	0.25	max	1	3	25%	1	8%	8	67%	2.0%	-4.8%	-5.0%	150.5	78.7
		average	1	8	67%	3	25%	1	8%	4.5%	-5.3%	-12.8%	154.1	135.4
		min	0	12	100%	0	0%	0	0%	-	-2.7%	-11.7%	160.7	427.1
	0.5	max	1	7	58%	4	33%	1	8%	6.3%	-7.9%	-19.1%	50.4	87.6
		average	1	8	58%	3	25%	2	17%	2.3%	-7.5%	-11.1%	44.9	151.6
		min	2	10	83%	2	17%	0	0%	1.8%	-5.7%	-11.8%	50.1	244.6
40	0.2	max	3	5	42%	5	42%	2	17%	2.9%	-9.4%	-19.1%	186.4	98.8
		average	3	9	75%	3	25%	0	0%	4.8%	-4.3%	-15.9%	196.3	184.3
		min	0	12	100%	0	0%	0	0%	-	-11.4%	-22.5%	202.9	547.6
	0.25	max	2	9	75%	3	25%	0	0%	5.4%	-4.1%	-8.8%	131.3	96
		average	4	6	50%	2	17%	4	33%	7.0%	-5.1%	-9.6%	137.2	210.7
		min	0	11	92%	1	8%	0	0%	4.6%	-5.6%	-12.1%	155.4	358.1
	0.5	max	1	7	58%	5	42%	0	0%	6.8%	-5.2%	-14.0%	48.2	114.9
		average	2	7	58%	5	42%	0	0%	2.3%	-3.3%	-4.6%	50.3	159.1
		min	1	10	83%	2	12%	0	0%	6.1%	-5.5%	-8.8%	53.7	252.2

Table 2. Summary of the test results of OTS (continued).

N No. of stations	d Density of FT chart	L No. of loops	No. of infeasible solutions of the base alg.	No. of better solutions		No. of worse solutions		No. of equal solutions		Average deviation of OTS objective function from the base alg.		Best improvement in the objective function of OTS	Average run time (seconds)	
				No.	%	No.	%	No.	%	Positive deviation	Negative deviation		Base alg.	OTS
50	0.2	max	3	7	58%	5	42%	0	0%	2.4%	-9.5%	-14.8%	173.6	143
		average	4	11	92%	1	8%	0	0%	3.3%	-10.0%	-21.7%	191.2	253.9
		min	0	11	92%	1	8%	0	0%	1.5%	-7.1%	-14.7%	219.7	477
	0.25	max	3	12	100%	0	0%	0	0%	-	-8.4%	-21.4%	128.6	161.1
		average	4	11	92%	1	8%	0	0%	0.9%	-9.4%	-16.0%	151.2	256.5
		min	1	12	100%	0	0%	0	0%	-	-11.5%	-21.4%	160.5	405.8
	0.5	max	4	10	83%	2	12%	0	0%	7.8%	-6.8%	-16.9%	63.3	170
		average	3	6	50%	4	33%	2	17%	1.2%	-3.8%	-5.9%	56	219.3
		min	2	10	83%	2	12%	0	0%	0.7%	-7.80%	-12.6%	66.9	221.3

Table 3. Summary of solution cases for OTS.

Type of case	Number of cases	Percent of cases	Average deviation from the base algorithm	Maximum deviation from the base algorithm
Problems Instances with Better objective functions	303	60%	-6.3 %	-22.5 %
Problems Instances with Worse objective functions	68	13%	3.6 %	13.3 %
Problems Instances with Equal objective functions	137	27%	-	-
Total	508	100%	-	-

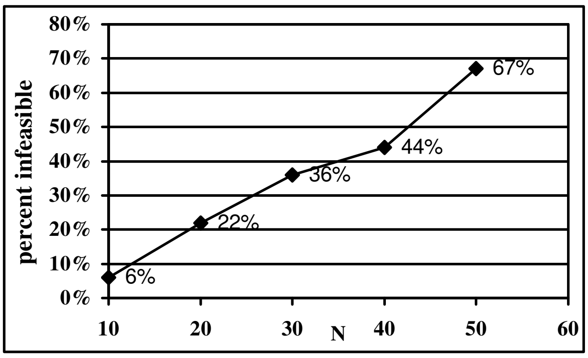


Figure 4. Percentage of infeasible solutions by base algorithm.

Table 4. Summary of test results for MSETS.

Problem No.	Run Time (seconds)	z^*_{MSE}	l^*_{MSE}	Best scenario α=0			Best scenario α=0.25			Best scenario α=0.5			Best scenario α=0.75			Best scenario α=1		
				z_{MSE}	l_{MSE}	*mw*	z_{MSE}	l_{MSE}	*mw*	z_{MSE}	l_{MSE}	*mw*	z_{MSE}	l_{MSE}	*mw*	z_{MSE}	l_{MSE}	*mw*
10-0.2-1	13.05	0.00002	2	0.00002	2	0.31	0.00002	2	0.31	0.00002	2	0.31	0.00002	2	0.31	0.00002	2	0.31
10-0.25-1	39.41	0.00001	2	0.00001	2	0.36	0.00001	2	0.36	0.00001	2	0.36	0.00001	2	0.36	0.00001	2	0.36
10-0.5-1	45.59	0.00006	2	0.00006	3	0.27	0.00006	3	0.27	0.00006	3	0.27	0.00006	3	0.27	0.00078	2	0.59
20-0.2-1	307.65	0.0008	3	0.0008	10	0.06	0.0011	4	0.35	0.0011	4	0.35	0.0011	4	0.35	0.0181	3	0.62
20-0.25-1	207.98	0.0022	4	0.0022	6	0.25	0.0022	6	0.25	0.0025	5	0.39	0.0025	5	0.39	0.0163	4	0.52
20-0.5-1	175.39	0.0016	5	0.0016	10	0.18	0.0016	10	0.18	0.0016	10	0.18	0.0039	7	0.32	0.0164	5	0.67
30-0.2-1	552.22	0.0021	5	0.0021	14	0.11	0.0021	14	0.11	0.0021	14	0.11	0.0072	7	0.38	0.0133	5	0.62
30-0.25-1	542.06	0.0047	6	0.0047	11	0.20	0.0047	11	0.20	0.0047	11	0.20	0.0070	9	0.29	0.0276	6	0.70
30-0.5-1	783.64	0.0125	8	0.0125	10	0.44	0.0125	10	0.44	0.0125	10	0.44	0.0203	8	0.68	0.0203	8	0.68
40-0.2-1	1196.42	0.0033	8	0.0033	11	0.34	0.0033	11	0.34	0.0033	11	0.34	0.0033	11	0.34	0.0240	8	0.60
40-0.25-1	1066.54	0.0081	8	0.0081	19	0.17	0.0081	19	0.17	0.0138	13	0.32	0.0275	8	0.66	0.0275	8	0.66
40-0.5-1	475.71	0.0429	13	0.0429	15	0.48	0.0429	15	0.48	0.0429	15	0.48	0.0429	15	0.48	0.0969	13	0.64
50-0.2-1	1529.5	0.0198	11	0.0198	23	0.18	0.0210	11	0.59	0.0210	11	0.59	0.0236	10	0.71	0.0236	10	0.71
50-0.25-1	1630.08	0.0083	12	0.0083	25	0.15	0.0083	25	0.15	0.0083	25	0.15	0.0209	12	0.58	0.0209	12	0.58
50-0.5-1	524.03	0.1194	18	0.1194	24	0.46	0.1205	23	0.44	0.1205	23	0.44	0.1614	18	0.63	0.1614	18	0.63

Table 5. Summary of test results for MADTS.

Problem No.	Run Time (sec-onds)	z^*_{MAD}	l^*_{MAD}	Best scenario *α=0*			Best scenario *α=0.25*			Best scenario *α=0.5*			Best scenario *α=0.75*			Best scenario *α=1*		
				z_{MAD}	l_{MAD}	*mw*	z_{MAD}	l_{MAD}	*mw*	z_{MAD}	l_{MAD}	*mw*	z_{MAD}	l_{MAD}	*mw*	z_{MAD}	l_{MAD}	*mw*
10-0.2-1	45.62	0.0059	2	0.0059	2	0.31	0.00589	2	0.31	0.00589	2	0.31	0.00589	2	0.31	0.00589	2	0.31
10-0.25-1	36.15	0.0145	2	0.0145	5	0.05	0.0145	5	0.05	0.0292	2	0.32	0.0292	2	0.32	0.0292	2	0.32
10-0.5-1	40.61	0.0126	2	0.0126	3	0.28	0.0126	3	0.28	0.0126	3	0.28	0.0126	3	0.28	0.0396	2	0.59
20-0.2-1	297.51	0.0401	3	0.0401	5	0.23	0.0401	5	0.23	0.0401	5	0.23	0.0401	5	0.23	0.1968	3	0.62
20-0.25-1	239.67	0.0244	4	0.0244	4	0.52	0.0244	4	0.52	0.0244	4	0.52	0.0244	4	0.52	0.0244	4	0.52
20-0.5-1	189.57	0.1052	5	0.1052	5	0.67	0.1052	5	0.67	0.1052	5	0.67	0.1052	5	0.67	0.1052	5	0.67
30-0.2-1	734.89	0.1394	5	0.1394	5	0.62	0.1394	5	0.62	0.1394	5	0.62	0.1394	5	0.62	0.1394	5	0.62
30-0.25-1	533.08	0.1344	6	0.1344	6	0.63	0.1344	6	0.63	0.1344	6	0.63	0.1344	6	0.63	0.1344	6	0.63
30-0.5-1	484.25	0.3215	8	0.3215	10	0.44	0.3215	10	0.44	0.3674	8	0.69	0.3674	8	0.69	0.3674	8	0.69
40-0.2-1	1355.33	0.0240	8	0.0240	8	0.60	0.0240	8	0.60	0.0240	8	0.60	0.0240	8	0.60	0.0240	8	0.60
40-0.25-1	1163.41	0.2741	8	0.2741	10	0.49	0.2741	10	0.49	0.3160	8	0.66	0.3160	8	0.66	0.3160	8	0.66
40-0.5-1	571.34	0.5114	13	0.5114	13	0.58	0.5114	13	0.58	0.5114	13	0.58	0.5114	13	0.58	0.5114	13	0.58
50-0.2-1	1523.28	0.3802	11	0.3802	12	0.52	0.3802	12	0.52	0.3802	12	0.52	0.3802	12	0.52	0.5765	11	0.63
50-0.25-1	1292.02	0.4272	12	0.4272	12	0.56	0.4272	12	0.56	0.4272	12	0.56	0.4272	12	0.56	0.4272	12	0.56
50-0.5-1	541.27	0.9951	18	0.9951	25	0.38	0.9951	25	0.38	0.9951	25	0.38	1.2295	18	0.61	1.2295	18	0.61

The first two phases of the base algorithm, which consist of generating subsets of stations and eliminating some of them, and the OTS algorithm were coded in Matlab 6.5. The IP model was coded with LINGO 8.0 (Roe, 1997). It was found that in most cases the effective size of the tabu list for the OTS is [N/5]±2. The maximum allowable number of non-improving iterations was set to 20. In solving OTS, three randomly selected initial solutions were used for each problem, except for some small problems where the number did not reach three.

The summary of the test results is shown in Table 2. The last two rows in the table correspond to the two examples of Bozer and Srinivasan [11]. The statistical information of Table 2 is summarized in Table 3. For small problems, the objective functions of both algorithms are often equal, which means they were the optimal solutions. The percentage of better solutions significantly increases with the instance size *N*.

The main advantage of OTS compared to the base algorithm is, in addition to improved solutions, the feasibility of solutions. OTS prevents intersections between the loops, but the base algorithm has no specific mechanism to ensure the independence of loops in the final configuration. As a result, a number of solutions generated by this algorithm are infeasible. The percentage of infeasible solutions as a function of the problem size is shown in Figure 4. The percentage of infeasible solutions increases with the problem size. For the largest problem size solved, more than 60% of the solutions are infeasible. It can be inferred that the base algorithm is not successful in solving problems with more than 20 stations. Nevertheless, the infeasibilities are present even in smaller problems.

Number of loops (*L*) has a significant effect on the running time of OTS. The average run time of the algorithm becomes higher as the number of loops increases. The explanation is that as the number of stations in the loops increases, more time is required to compute the workloads. In comparison with the base algorithm, OTS is often faster for lower flow densities and higher *L* values. In general, it seems that for larger problems the base algorithm tends to be faster than OTS, but the latter algorithm remains preferable because it always produces feasible solutions.

4.2.2. Computational results for MSETS and MADTS algorithms

As the number of loops is not fixed in these algorithms, they are solved for all feasible numbers of loops from [N/2] to 2. The resulting solutions were scenarios consisted of MAD or MSE and also maximum workload value obtained for each scenario. Here, we have suggested a weighted multi-attribute objective function to select the best scenario:

$$\text{Min}(w_1): \alpha.\left(\frac{z_{MSE} - z^*_{MSE}}{z^*_{MSE}}\right) + (1-\alpha).\left(\frac{l_{MSE} - l^*_{MSE}}{l^*_{MSE}}\right) \quad (5)$$

$$\text{Min}(w_2): \alpha.\left(\frac{z_{MAD} - z^*_{MAD}}{z^*_{MAD}}\right) + (1-\alpha).\left(\frac{l_{MAD} - l^*_{MAD}}{l^*_{MAD}}\right) \quad (6)$$

where in (5), z_{MSE} and l_{MSE} are MSE and *L* values for each scenario, z^*_{MSE} is the minimum of z_{MSE} values, and l^*_{MSE} is the minimum value of l_{MSE} values. Similar definitions apply to (6). Finally α and 1-α (0≤α≤1) are the weight coefficients assigned to normalized values of z and l.

Due to the structural similarities between the algorithms, and based on test results, the same formula was set for the MSETS and MADTS. The maximum allowable number of non-improving iterations was set to 40 for MSETS and MADTS. Tests were carried out on a 1.5 GHz Intel Pentium 4 Centrino, with 256 MB RAM.

Since there were no counterpart algorithms to be compared with the new algorithms, we solved them for only one instance of each problem category. The best scenarios were selected for 5 levels of α weight. Tables 4 and 5 indicate the best results of loop number and workload balance for each problem instance, and the best design scenarios considering each α level for MSETS and MADTS. In all cases, MSETS and MADTS were able to find the same minimum loop number as the partitioning heuristic of Bozer and Srinivasan [11] with acceptable maximum workloads.

5. Conclusion

We have proposed three new algorithms for designing AGV routes in a tandem configuration using tabu search metaheuristic. The first algorithm was designed based on the definitions proposed by Bozer and Srinivasan [11], which assumed fixed loop number and minimization of the maximum workload as the objective function. The two other algorithms were designed based on variable loop numbers and workload balance as the objective function. We used mean squared error and mean absolute error as indicators of workload balance in these two algorithms. Our first heuristic was compared to the base heuristic of Bozer and Srinivasan [11]. The two algorithms were run for/to solve 60 randomly generated problems at three levels of loop number. Results show that our algorithm is capable of producing better solutions and the amount of improvements in the objective function with respect to the base algorithm tends to be higher as the problem size increases. We solved our second and third algorithms for some test problems and presented different selectable design scenarios using two multi-attribute evaluation functions. The main advantage of our TS algorithms its avoidance of generating overlapping loops. Results show that as the problem size increases, the likelihood of generating infeasible solutions is very high for the base algorithm.

References

[1] Aarab, A. Chetto, H. and Radouance, L., 1999, Flow path design for AGV systems. *Studies in Informatics and Control*, 8(2), 97-106.

[2] Asef-Vaziri, A. Dessouky, M. and Sriskandarajah, C., 2001, A loop material flow system design for automated guided vehicles. *International Journal of flexible Manufacturing Systems*, 13, 33-48.

[3] Asef-Vaziri, A. and Laporte, G., 2005, Loop based facility planning and material handling. *European Journal of Operational Research*, 2005, 164, 1-11.

[4] Asef-Vaziri, A., Laporte, G. and Sriskandarajah, C., 2000, The block layout shortest loop design problem. *IIE Transactions*, 2000, 32, 724-734.

[5] Asef-Vaziri, A., Laporte, L. and Ortiz, R., 2007, Exact and heuristic procedures for the material handling circular flow path design problem. *European Journal of Operational Research*, 176(2), 707-716.

[6] Banerjee, P. and Zhou, Y., 1995, Facilities layout design optimization with single loop material flow path configuration. *International Journal of Production Research*, 33, 183-203.

[7] Barad, M. and Sinriech, D., 1998, 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.

[8] Bartholdi, J. J. and Platzman, L. K., 1989, Decentralized control of automated guided vehicles on a simple loop. *IIE Transactions*, 21(1), 76-81.

[9] Bozer, Y. A. and Srinivasan, M. M., 1989, Tandem configuration for automated guided vehicle systems offers simplicity and flexibility. *Industrial Engineering*, 21, 23-27.

[10] Bozer, Y. A. and Srinivasan, M. M., 1991, Tandem configuration for automated guided vehicle systems and the analysis of single vehicle loops. *IIE Transactions*, 23(1), 72-82.

[11] Bozer, Y. A. and Srinivasan, M. M., 1992, Tandem AGV systems: A partitioning algorithm and performance comparison with conventional AGV systems. *European Journal of Operational Research*, 63, 173-191.

[12] Bozer, Y. A. and Lee, C. G., 2004, Using Existing Workstations as Transfer Stations in Tandem AGV Systems. *Journal of Manufacturing Systems*, 23(3), 229-241.

[13] Chhajed, D., Montreuil, B. and Lowe, T., 1992, Flow network design for manufacturing systems layout. *European Journal of Operational Research*, 57, 145-161.

[14] Choi, H. G., Kwon, H. J. and Lee, J., 1994, Traditional and tandem AGV system layouts: A simulation study. *Simulation*, 63(2), 85-93.

[15] F.A.-Vis and I. F. A., 2006, Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, 170, 677-709.

[16] Faraji, M. and Batta, R., 1994, Forming cells to eliminate vehicle interference and system locking in an AGVS. *International Journal of Production Research*, 32(9), 2219-2241.

[17] Farling, B. E., Mosier, C. T. and Mahmoodi, F., 2001, Analysis of automated guided vehicle configuration in flexible manufacturing systems. *International Journal of Production Research*, 39, 4239-4260.

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

[19] Gaskin, R. J., Tanchoco, J. M. A. and Taghab-

oni, F., 1989, Virtual flow paths for free ranging automated guided vehicle systems. *International Journal of Production Research*, 27, 91-100.

[20] Ho, Y. C. and 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.

[21] Hsieh, L. F. and Sha, D. Y., 1996, A design process for tandem automated guided vehicle systems: the concurrent design of machine layout and guided vehicle routes in tandem automated guided vehicle systems. *Integrated Manufacturing Systems*, 7, 30-38.

[22] Huang, C., 1997, Design of material transportation system for tandem automated guided vehicle systems. *International Journal of Production Research*, 35, 943-953.

[23] Kaspi, M., Kesselman, U. and Tanchoco, J. M. A., 2002, Optimal solution for the flow path design problem of a balanced unidirectional AGV system. *International Journal of Production Research*, 40, 349-401.

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

[25] Kim, K. S., Chung, B. D. and Jae, M., 2003, A design for a tandem AGVS with multi-load AGVs. *International Journal of Advanced Manufacturing Technology*, 22, 744-752.

[26] 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.

[27] Ko, K. C. and Egbelu, P. J., 2003, Unidirectional AGV guide path network design: A heuristic algorithm. *International Journal of Production Research*, 41, 2325-2343.

[28] Kouvelis, P., Gutierrez, G. J. and Chiang, W. C., 1992, Heuristic unidirectional flow path design approach for automated guided vehicle systems. *International Journal of Production Research*, 30, 1327-1351.

[29] Laporte, G., Asef-Vaziri, A. and Sriskandarajah, C., 1996, Some application of the generalized traveling salesman problem. *International Journal of Operational Research Society*, 47, 1461-1467.

[30] Lin, J. T., Chang, C. C. K. and Liu, W. C., 1994, A load routing problem in a tandem-configuration automated guided vehicle system. *International Journal of Production Research*, 32, 411-427.

[31] Liu, F. H. and Chen, J. T., 1997, Analytical framework for designing the divided automated guided vehicles system. *International Journal of Industrial Engineering*, 4(2), 90-102.

[32] Rajagopalan, S., Heragu, S. S. and Taylor, G. D., 2004, A lagrangian relaxation approach to solving the integrated pick-up/drop-off point and AGV flow path design problem. *Applied Mathematics and Modeling*, 28, 735-750.

[33] Roe, A., 1997, *User's Guide for LINDO and LINGO*. Windows Versions, Duxbury Press, Belmont (Cal.).

[34] Ross, E. A., Mahmoodi, F. and Mosier, C. T., 1996, Tandem configuration automated guided vehicle systems: A comparative study. *Decision Sciences,* 27, 81-102.

[35] Seber, G. A. F., 1984, *Multivariate Observations*, Wiley, New York.

[36] Seo, Y. and Egbelu, P. J., 1995, Flexible guide path design for automated guided vehicle systems. *International Journal of Production Research*, 33, 1135-1156.

[37] Sinriech, D. and Tanchoco, J. M. A., 1991, Intersection graph method for AGV flow path design. *International Journal of Production Research*, 29, 1725-1732.

[38] 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(3), 705-725.

[39] Sinriech, D. and Tanchoco, J. M. A., 1994, *Material Flow Systems Manufacturing*. Chapman & Hall, London, 200-235.

[40] Sinriech, D. and Tanchoco, J. M. A., 1995, An introduction to the segmented flow approach to discrete material flow systems. *International Journal of Production Research*, 33, 3381-3410.

[41] 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.

[42] Sun, X. C. and Tchernev, N., 1996, Impact of empty vehicle flow on optimal flow path design for unidirectional AGV systems. *International Journal of Production Research*, 34, 2827-2852.

[43] Tanchoco, J. M. A. and Sinriech, D., 1992, OSL-optimal single-loop guide paths for AGVs. *International Journal of Production Research*, 30(3), 665-681.

[44] Tompkins, J. A., White, J. A., Bozer, Y. A. and Tanchoco, J. M. A., 2003, *Facilities Planning*. 3rd edition, John Wiley & Sons, New York.

[45] Venkataramanan, M. A. and Wilson, K. A., 1991, A branch and bound algorithm for flow path design of automated guided vehicle systems. *Naval Research Logistics Quarterly*, 38, 431-445.

[46] Ventura J. A. and Lee, C. U., 2001, Tandem Loop with Multiple Vehicles Configuration for Automated Guided Vehicle Systems. *Journal of Manufacturing Systems*, 20, 153-165.

[47] Yu, W. and Egbelu, P., 2001, Design of a variable path tandem layout for automated guided vehicle systems. *Journal of Manufacturing Systems*, 20, 305-319.

[48] Z.-Farahani, R., Karimi, B. and Tamaddon, S., 2006, Designing an efficient method for simultaneously determining the loop and the location of the P/D stations using genetic algorithm. *International Journal of Production Research*, In press.

[49] Z.-Farahani, R. and Laporte, G., 2004, *Two Formulations for Designing Optimal Single Loop and the Location of P/D Stations*. Proceedings of IIEC Conference, Tehran. Iran.

[50] Z.-Farahani, R., Laporte, G. and Sharifyazdi, M., 2005, A practical exact algorithm for the shortest loop design problem in a block layout. *International Journal of Production Research*, 43(9), 1879-1887.

[51] Z.-Farahani, R., Pourakbar, M. and Miandoabchi, E., 2006, Developing exact and tabu search algorithms for simultaneously determining AGV loop and P/D stations in single loop systems. *International Journal of Production Research*, In press.

[52] Z.-Farahani, R. and Tari, F. G., 2001, Optimal flow path designing of unidirectional AGV systems. *International Journal of Engineering Sciences*, 12, 31-44.

[53] Z.-Farahani, R. and Tari, F. G., 2002, A branch & bound method for finding flow-path designing of AGV systems. *IIE Transactions A: Basics*, 15, 81-90.