Journal of Optimization in Industrial Engineering
Vol.13, Issue 1, Winter & Spring 2020, 67-80
DOI: 10.22094/JOIE.2018.537.37

Research Article

A Mathematical Model for Multiple-load AGVs in Tandem Layout

Behnam Rahimikelarijani[a], Mohammad Saidi-Mehrabad[a,*], Farnaz Barzinpour[a]

[a] *Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran*
Received 11 October 2015; Revised 18 April 2018; Accepted 24 April 2018

Abstract

Reducing cost of material handling has been a big challenge for companies. Flexible manufacturing system employed automated guided vehicles (AGV) to maintain efficiency and flexibility. This paper presents a new non-linear mathematical programming model to group *n* machines into *N* loops, to make an efficient configuration for AGV system in Tandem layout. The model minimizes both inter-loop, intra-loop flow and use balanced-loops strategy to balance workload in system simultaneously. This paper significantly considers multiple-load AGVs, which has capability of reducing fleet size and waiting time of works. A modified variable neighborhood search method is applied for large size problems, which has good accuracy for small and medium size problems. The results indicate that using multiple load AGV instead of single load AGV will reduce system penalty cost up to 44%.

***Keywords*:** AGV, Tandem, Multiple-load, Machine-to-loop assignment, variable neighborhood search

1. Introduction

An AGV is an automated guided vehicle, using for horizontal transportation of goods and materials. AGVs traveling on a network of guide-paths. These vehicles are designed to perform repetitive, though not continual, tasks. However, their designs vary according to their usage, capacity and environment. They use in warehousing system, production line, container terminal, public transportation for material handling. The flexible manufacturing system (FMS) composed of different work cells, where different workstations are located in these cells. Categorizing workstations into different cells or loop and assign an AGV with appropriate workload capacity is one the initial decisions. To design such a system, first, the facilities should be located.

The systems can vary by the flow path configuration, which is known as a method of classification in AGV system. (Choobineh, Asef-Vaziri and Huang, 2012) enumerate these configurations to be conventional, single loop, tandem and segmented. The advantages of tandem configuration, as mentioned by (R.Tavakkoli-Moghaddam et al., 2008), include control over AVG, elimination of collision and deadlocks, and simplification in production planning. All these features encouraged several researchers to apply tandem configuration in their studies.

Tandem configuration distributed stations into non-overlapping loops, where each loop is served by one AGV, which is not allowed to move between loops. Hence, to perform inter-loop transportations, transfer points are considered (Fig 1)

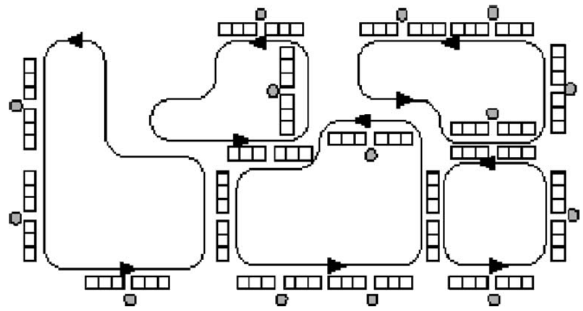

Fig. 1. tandem configuration as proposed by (Bozer and Srinivasan, 1989)

This paper addresses a new mathematical to group *n* machines into N loops and minimize both inter-loop and intra-loop flow simultaneously.

2. Literature Review

The problems in Tandem AGV system are categorized as follow (Salehipour, Kazemipoor and Moslemi Naeini, 2011);

- Determine number of loops (fleet size)
- Assigning workstations to loops
- Arranging stations in each loop
- Specifying optimum transfer point between loops

*Corresponding author Email address: mehrabad@iust.ac.ir

- Flow direction in loops (clockwise or counter clockwise)

After selecting AGV as the carrier, decisions should be made about AGV capacity. As claimed by (Ozden, 1988), increases in capacity from one to two, results in fleet reduction. (Meer, 2000) believes that an increase in AGV load leads into decrease in output mean time. As (Vis, 2006) puts it, most of researches focus on single load AGV and there are a few researches circling around multiple-load AGV. Henceforth, this specification is taken into account for the purpose of this research.

(Bozer and Srinivasan, 1992)solve a model to balance workload between loops by heuristic algorithm. (Aarab, Chetto and Radouane, 1999) Solved model to minimize inter loop flow. (Yu and Egbelu, 2001) Minimized and balanced workload in loops by a heuristic partitioning algorithm, but (Kim, Chung and Jae, 2003) (Ho & Hsieh 2004) and (Kim and Chung, 2007) present a new concept introduced for AGV system which considered multiple load AGV. (Ho & Hsieh 2004) solved this problem in two steps; in first step, they solve a traveling salesman problem to generate workstation subsets. Then, use a Markov chain model and its limiting probabilities for a feasibility test of the generated subsets. In the second step, they determine a final guide path set, but (Ho, Liu and Yih, 2012) first simulated AGV system and estimated AGVs performance coefficient, they mentioned that these coefficient change in nonlinear manner, when AGV load is one, it is one but when AGV load increase to two this coefficient is about 1.9. (Aarab, Chetto and Radouane, 1999) Proposed hierarchical partitioning method in which similarity coefficient figure out on flow amount and distance. (Yu and Egbelu, 2001) proposed configuring method in AGV system which minimize number of loops and transportation, they solved it by partitioning heuristic algorithm, in this research they proposed variable path instead of loops approach.(Shalaby, El Mekkawy and Fahmy, 2006) Solved designing single load tandem AGV system with two phase algorithm by three objective function, minimizing transportation cost, minimizing maximum system workload and minimizing inter loop transportation. (Zanjirani Farahani *et al.*, 2008)solved minimize maximum workload objective function by Meta heuristics algorithm, simulated annealing and genetic algorithm, and comparing these results. (R.Tavakkoli-Moghaddam et al., 2008) proposed non-linear programming model which assigned stations to loops by minimizing inter loops transportations objective function and balancing strategy, they introduced balancing coefficients in constraints to balance inter loops flow similar work was done in (Fan, He and Zhang, 2015). (ElMekkawy and Liu, 2009) Solved stations to loops assignment problem by minimum maximum workload objective function with dominating genetic algorithm and a local search algorithm. (Rezapour, Zanjirani-Farahani and Miandoabchi, 2010) Proposed multi objective function problem, which minimize inter loop flow and total flow and balancing workload between loops and solved it by simulated annealing algorithm. (Salehipour, Kazemipoor and Moslemi Naeini, 2011) Proposed integer-programming model to minimize total flow between stations and solved it by variable neighborhood search. They were concerned about reducing stations waiting time. (Liu *et al.*, 2018) Design to tandem AGV system using co-evolutionary algorithm to design station locations and all transfer points. As it is mentioned, few researches consider multiple load AGV systems. (Fazlollahtabar, Saidi-mehrabad and Balakrishnan, 2015) proposed a mathematical model to minimize the penalized material handling earliness and tardiness to satisfying the expected cycle for multiple load AGVS. In another research (Fazlollahtabar and Saidi—Mehrabad, 2015) proposed a risk based dynamic program to determine more reliable arcs for fortification purposes, considering multi-stage decision making process of the multiple AGVs on different arcs, they developed dynamic program being a useful tool for multi stage decision making. Discussion and cllasification of different type of AGV system problems are done in different papres, for example (Carlo, Vis and Roodbergen, 2014) , (Fazlollahtabar and Saidi-Mehrabad, 2015) , (Kumar *et al.*, 2015), (Gutta *et al.*, 2018) and (Das and Pasan, 2016).

In this research a mathematical programming model for multiple-load AGV in tandem configuration is presented, in which we try to consider the concerns about transportation and balancing workload using proposed balancing strategy in (R.Tavakkoli-Moghaddam et al., 2008). In solution methodology, different solution procedures will be offered and evaluated to find the best method to solve large scale problems.

The rest of this paper is arranged as follow, in section 2 notations are explained and programming model will be presented , Since the proposed programming model is NP-hard, in section 3 a modified variable neighborhood search (VNS) algorithm is proposed, in which we modify our programming model by Lagrangian relaxation (LR). In section 4, programming model will be tested, and results of exact methods and VNS algorithm will be evaluated. Finally, in section 5 conclusion of this study are presented.

3. Mathematical Model

The assumptions of the presented model are as follows:

- AGVs are bidirectional.
- There is only one AGV in each loop.
- There are at least two stations in each loop.
- Each station can only be assigned to one loop.
- The number of loops is considered as the input data to the algorithm.
- Only loaded transportations are considered.

The objective is to assign stations to non-overlapping loops in order to minimize inter-loop and intra-loop transportations. This goal is supposed to be achieved by assigning the AGV with a certain capacity to appropriate loops. Balance strategy, as proposed by (R.Tavakkoli-Moghaddam et al., 2008), will be implored to balance the workload in each loop according to the capacity of the assigned AGV. The indexes of proposed programming model are as follows:

i(k): defines i(k)th station
j(l): defines j(l)th station
p: defines AGV capacity
In our proposed programming model, f_{ik} refers to the number of transported loads between two stations.
The design variables are as follow:
x_{ij}: It equals to 1 if station i assigns to loop j; otherwise it equals to zero.
y_{jp}: It equals to 1 if AGV with P capacity assigns to loop j; otherwise it equals to zero.
Other parameters are:
w_p: Performance coefficient of AGV with P capacity
f_{ik}: Flow between station i and j
f_i: Flow from different station to station i;
T: The total time that AGV is available.
t_i: The mean time of loading/unloading and the process duration in the i^{th} station.
t'_j: The bottleneck time in j^{th} station.
N: The total number of predefined loops.
$\acute{\eta}, \eta$: Upper and lower bound of flow coefficient in each loop.
λ_p: Penalty of selecting AGV with P capacity
n: Number of stations
After the above whole introduction to programming model parameters, decision variables and indexes, we now turn to describe our model.

3.1. Proposed model

Model I is the proposed programming model, meant to assign stations to non-overlapping loops in multiple-load AGV system, while meeting the above-mentioned assumptions:

Model I

$$\text{Min } Z = \underbrace{\sum_{j=1}^{N}\sum_{l=1}^{N}\left(\frac{\sum_{i=1}^{n}\sum_{k=1}^{n} f_{ik}x_{ij}x_{kl}}{\sum_{p=1}^{m} y_{jp}w_p}\right)}_{i\neq k, j=l} + \underbrace{\sum_{i=1}^{n}\sum_{j=1}^{N}\sum_{k=1}^{n}\sum_{l=1}^{N} f_{ik}x_{ij}x_{kl}}_{i\neq k, j\neq l} + \sum_{j=1}^{N}\sum_{p=1}^{m} y_{jp}w_p\lambda_p \tag{1}$$

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad i = 1,2,3\ldots,n \tag{2}$$

$$\sum_{p=1}^{m} y_{jp} = 1 \qquad j = 1,2,3\ldots,N \tag{3}$$

$$\sum_{i=1}^{n} x_{ij} \geq 2 \qquad j = 1,2,3\ldots,N \tag{4}$$

$$\eta\left(\frac{\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{N}\sum_{k=1}^{n}\sum_{l=1}^{N} f_{ik}x_{ij}x_{kl}}_{i\neq k}\times\sum_{p=1}^{m} y_{jp}w_p}{\sum_{j=1}^{N}\sum_{p=1}^{m} y_{jp}w_p}\right) \leq \underbrace{\sum_{i=1}^{n}\sum_{k=1}^{n} f_{ik}x_{ij}x_{kj}}_{i\neq k} \qquad j=1,2,3\ldots,N \tag{5}$$

$$\acute{\eta}\left(\frac{\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{N}\sum_{k=1}^{n}\sum_{l=1}^{N} f_{ik}x_{ij}x_{kl}}_{i\neq k}\times\sum_{p=1}^{m} y_{jp}w_p}{\sum_{j=1}^{N}\sum_{p=1}^{m} y_{jp}w_p}\right) \geq \underbrace{\sum_{i=1}^{n}\sum_{k=1}^{n} f_{ik}x_{ij}x_{kj}}_{i\neq k} \qquad j=1,2,3\ldots,N \tag{6}$$

$$t'_j = \max_i\{f_i t_i x_{ij}\} \qquad j=1,2,3,..,N \tag{7}$$

$$\sum_{i=1}^{n} x_{ij} \leq \frac{T\sum_{p=1}^{m} y_{jp}w_p}{t'_j} \qquad j=1,2,3,..,N \tag{8}$$

$$x_{ij}\epsilon\{0,1\} \qquad \forall\, i,j \tag{9}$$

$$y_{jp}\epsilon\{0,1\} \qquad \forall\, i,j \tag{10}$$

Equation 0 is the objective function of the problem. The first part of Equation0 minimizes intra-loop transportations by considering the assigned AGV capacity. The second part of Equation 0 minimizes inter-loop transportations and the third part of Equation 0 penalizes AGV capacity, to prevent surplus increase in AGV capacity.
Constraint0 ensures that each station is assigned to only one loop. Constraint 0 ensures that only one AGV with specific capacity is assigned to each loop. Constraint 0 is to remove single station loops and ensures that at least there are two stations in each loop. Constraints 0 and are aimed at equal balance strategy between loops. $\acute{\eta}, \eta$ are defined to make these constraints feasible. Since obtaining an exactly equal workload is not feasible, these coefficients are defined to indicate the possible range of flow in loops.

$$\eta = \frac{1}{N} - \Delta \qquad \acute{\eta} = \frac{1}{N} + \Delta \tag{11}$$

$$\text{Where, } 0 \leq \Delta \leq \frac{1}{N} \tag{12}$$

Equations 0 and 0 are indicative of the process through which this coefficient is determined. If there are 2 (N)

loops, $\frac{1}{2(N)}$ workload must be assigned to each loop. However, in real-world situations, it is not feasible to assign an exact amount of work. Thus, in order to avoid infeasibility and make small range variation possible, Δ is defined, which is called balancing flow coefficient in this study. Definitions of this concept are elaborated prominently in (R.Tavakkoli-Moghaddam et al., 2008).
Constraint demonstrates bottleneck station and its cycle time in each loop. Constraint 0 is meant to assess the feasibility of providing service to those stations which are assigned to a specific loop, while taking the AGV capacity into consideration. This constraint ensures that this type of AGV is capable of performing intra-loop transportations. Constraints 0 and 0 state that x_{ij} and y_{jp} are binary variables.
To determine the value of w_p, loading and unloading policy must be established precisely. Accordingly, different policies result in quite different outcomes. As discussed by (Ozden, 1988) and (Ho * and Hsieh, 2004) an increase in the capacity of AGV leads into a decrease in fleet size, and consequently, an increase in performance coefficient. However, it should be pointed out that these increases do not follow a linear relationship. Importantly, determining the appropriate loading and unloading policy raises argumentative discussions and needs to be elaborated with more scrutiny. As a result, random but logical values was used for w_p, based on the capacity of AGV in this study.

3.2. Simplifying the model to produce feasible solutions

A nonlinear programming model was introduced in the previous session. This model can be simplified by application of a general method in a quadratic assignment problem. To this end, the following decision variables are defined, i.e. $z_{ijkl}=x_{ij}\times x_{kl}$. In order to add the decision variable z_{ijkl}, some extra constraints are required, namely 0 , 0 and 0, this model is still non-linear, due to the presence of some decision variables in denominator of constraints and objective function. In order to solve this problem, we use parameters Ω as follow:

$$\Omega_j=\sum_{p=1}^{m} y_{jp}w_p \qquad \forall j \tag{13}$$

$$\Omega_{j\geq 1} \qquad \forall j \tag{14}$$

Applying the necessary changes to the previous model results in

Model II:

Model II

$$\text{Min } Z=\underbrace{\sum_{j=1}^{N}\sum_{l=1}^{N}\left(\frac{\sum_{i=1}^{n}\sum_{k=1}^{n} f_{ik}z_{ijkl}}{\Omega_j}\right)}_{i\neq k,\, j=l}+\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{N}\sum_{k=1}^{n}\sum_{l=1}^{N} f_{ik}z_{ijkl}}_{i\neq k,\, j\neq l}+\sum_{j=1}^{N}\sum_{p=1}^{m} y_{jp}w_p\lambda_p \tag{15}$$

$$\sum_{j=1}^{N} x_{ij}=1 \qquad i=1,2,3\ldots,n$$

$$\sum_{p=1}^{m} y_{jp}=1 \qquad j=1,2,3\ldots,N$$

$$\sum_{i=1}^{n} x_{ij}\geq 2 \qquad j=1,2,3\ldots,N$$

$$\eta\left(\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{N}\sum_{k=1}^{n}\sum_{l=1}^{N} f_{ik}z_{ijkl}}_{i\neq k}\times\Omega_j\right)\leq\sum_{j=1}^{N}\Omega_j\times\underbrace{\sum_{i=1}^{n}\sum_{k=1}^{n} f_{ik}z_{ijkj}}_{i\neq k} \qquad j=1,2,3\ldots,N \tag{16}$$

$$\acute{\eta}\left(\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{N}\sum_{k=1}^{n}\sum_{l=1}^{N} f_{ik}z_{ijkl}}_{i\neq k}\times\Omega_j\right)\geq\sum_{j=1}^{N}\Omega_j\times\underbrace{\sum_{i=1}^{n}\sum_{k=1}^{n} f_{ik}z_{ijkj}}_{i\neq k} \tag{17}$$

$j=1,2,3\ldots,N$

$$\acute{t}_j\geq f_i t_i x_{ij} \qquad j=1,2,3,\ldots,N$$

$$\acute{t}_j\sum_{i=1}^{n} x_{ij}\leq T\Omega_j \qquad j=1,2,3,..,N$$

$$x_{ij}+x_{kl}-2z_{ijkl}\geq 0 \qquad \forall\, i,j,k,l \tag{18}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{N}\sum_{k=1}^{n}\sum_{l=1}^{N} z_{ijkl}=\frac{n(n-1)}{2} \qquad k>i \tag{19}$$

$\Omega_j = \sum_{p=1}^{m} y_{jp} w_p \qquad \forall\, j$

$\Omega_{j \geq 1} \qquad \forall\, j$

$x_{ij} \in \{0,1\} \qquad \forall\, i, j$

$y_{jp} \in \{0,1\} \qquad \forall\, i, j$

$z_{ijkl} \in \{0,1\} \qquad \forall\, i, j, k, l$ (20)

As it is pointed out in (Christofides, Mingozzi and Toth, 1980) and (Ho * and Hsieh, 2004), constraint 0 is necessary.

4. Solving Model

In this study, the proposed programming model is that of quadratic assignment. Furthermore, as regards (Gilmore, 1962), (Christofides, Mingozzi and Toth, 1980) and (Ho * and Hsieh, 2004) and also with concern to the extra constraint 0, the problem can be classified as an NP-hard combinatorial optimization problem. Thus, metaheuristic algorithms should be used, when solving large-sized problem or medium-sized ones. Our proposed solving algorithm is a Modified Variable Neighborhood Search (VNS), which is a single solution algorithm, introduced by (Hansen and Mladenovi, 2001). The main idea in VNS focuses on exploring consecutive neighborhood to obtain the best solution. This algorithm finds local optimum by searching different levels of neighborhood, and finally achieves the best result (Fig. 2). This algorithm does not require parameter tuning and exploring neighborhood is systematic or random.

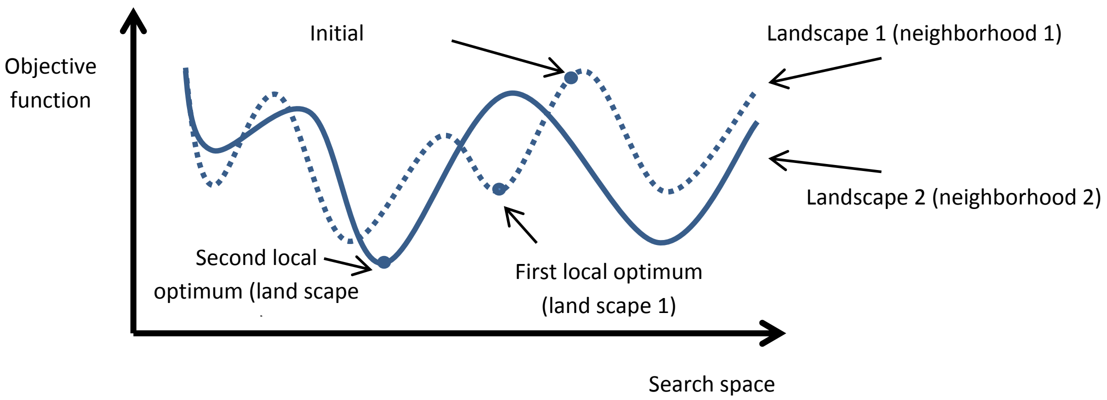

Fig. 2. Variable neighborhood search using two neighborhoods. The first local optimum is obtained according to the neighborhood 1. According to the neighborhood 2, the second local optimum is obtained from the first local optimum (Talbi, 2009).

VNS finds local optimum and, through a change in neighborhood level, escapes from these local optimums in order to find global optimum. In fact, it can produce different local optimum by exploring different levels of neighborhood. Hence, it is possible that one of the local optimums be the global one. The pseudo-code of the proposed algorithm is presented in.

Input: *a set of neighborhood structures* N_k *for* $k = 1, k_{max}$ *for shaking.*
$X = X_0$ *; /* Generate the initial solution */*
Repeat
    $k = 1$;
    Repeat
        Shaking: pick a random solution X *from the* k^{th} *neighborhood* $N_k(X)$ *of* X ;
            $X'' =$ *local search*(X') ;
            Assigning AGV;
            If $f(X'') \leq f(X)$ **Then**
                $X = X''$;
                Continue to search with N_1 ; $k = 1$;
            Otherwise $k=k+1$;
        Until $k = k_{max}$

Fig. 3. Template of the basic variable neighborhood search algorithm (Talbi, 2009)

In what follows, in order to improve the searching process we add an elitism procedure at initiation of algorithm, this procedure select the best answer among the firsts and continue the rest of algorithm using this answer, also to adopt the proposed algorithm to mathematical model, the process of generating initial solution, producing neighborhood and revising objective function will be presented. To demonstrate the performance of the proposed algorithm, first the programming model will be tested by small-sized problems, as discussed in the literature review. After obtaining the desired result, it will be applied to large-sized problems. Moreover, the accuracy of results will be verified, using third solving method (numeration algorithm).

4.1. Initial solution

The first step towards solving the problem is making initial solution, which can vary according to the problem structure. A three-row matrix is defined to elaborate on a solution. In this matrix, the first row shows the station name, the second row shows the loop number to which the station in upper row is assigned, and in the last row, the capacity of AGV in that loop is presented. Through an increase in the number of stations, the columns of this matrix will increase, as well. For instance, suppose that we need to assign 6 stations to 2 loops by selecting appropriate type of AGV. Here, the AGVs by 1 to 3 loads are available. Given this problem, our result matrix will be as presented in Fig 4.

Station	1	2	3	4	5	6
Loop	1	2	2	1	1	2
AGV capacity	1	3	3	1	1	3

Fig. 4 solution structure

In the above-mentioned example, station 1, 4, and 5 are in loop 1 and an AGV with load capacity 1 is assigned. On the other hand, station 2, 3, and 6 are assigned to loop 2, in which an AGV with load capacity 3 is active.

4.2. Objective function

In the second step of our problem, in order to design a more efficient solving procedure, the objective function is modified to meet all constraints. As a consequence, Lagrangian relaxation approach is applied (MS.Bazaraa, 1993). In this approach, the constraints, which are difficult to meet, will be rewritten. For example, suppose a programming model with Y objective function and $g_i(x)$ constraints:

$$\begin{aligned} &\text{Min } Y = f(x) \\ \text{S.t:}\quad &g_i(x) \le b_i, \forall i \end{aligned} \tag{21}$$

To remove less–than-or-equal-to constraints and rewrite them in objective function, the following change was made:

$$\min y = \{f(x) + rk(x)\} \tag{22}$$

In the above-mentioned equation, r is penalty coefficient and k(x) is penalty function;

$$k(x) = \sum_{i=1}^{m} \max\{(g_i(x) - b_i), 0\}^2 \tag{23}$$

Now, the problems 0 and 0 are equivalent. The same modification will be made to the constraints 0, 0 and 0. The final objective function is (Model III)

Model III

$$\begin{aligned} \text{Min } Z = &\underbrace{\sum_{j=1}^{N}\sum_{l=1}^{N}\left(\frac{\sum_{i=1}^{n}\sum_{k=1}^{n} f_{ik} z_{ijkl}}{\Omega_j}\right)}_{i\neq k, j=l} + \underbrace{\sum_{i=1}^{n}\sum_{j=1}^{N}\sum_{k=1}^{n}\sum_{l=1}^{N} f_{ik} z_{ijkl}}_{i\neq k, j\neq l} + \sum_{j=1}^{N}\sum_{p=1}^{m} y_{jp} w_p \lambda_p \\ &+ r\sum_{j=1}^{N}\left(\text{Max}\left[\left\{\eta\left(\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{N}\sum_{k=1}^{n}\sum_{l=1}^{N} f_{ik} z_{ijkl}}_{i\neq k} \times \Omega_j\right) - \sum_{j=1}^{N}\Omega_j \times \underbrace{\sum_{i=1}^{n}\sum_{k=1}^{n} f_{ik} z_{ijkj}}_{i\neq k}\right\}, 0\right]^2\right. \\ &\left.+ \text{Max}\left[\left\{\eta\left(\sum_{j=1}^{N}\Omega_j \times \underbrace{\sum_{i=1}^{n}\sum_{k=1}^{n} f_{ik} z_{ijkj}}_{i\neq k} - \underbrace{\sum_{i=1}^{n}\sum_{j=1}^{N}\sum_{k=1}^{n}\sum_{l=1}^{N} f_{ik} z_{ijkl}}_{i\neq k} \times \Omega_j\right)\right\}, 0\right]^2\right) \end{aligned} \tag{24}$$

Constraints 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 and 0 are the same as it was mentioned in is our proposed programming model, meant to assign stations to non-overlapping loops in multiple-load AGV system, while meeting the above-mentioned assumptions:

4.3. Generating neighborhood solution

To generate neighborhood solution, neighborhood level needs to be considered in the first place. For example, in order to generate second level neighborhood, two stations should be selected. Then, two loops should be selected based on two criteria; first, the produced solution should be different, and second, there should be two stations in each loop. By considering these two criteria, a new solution will be generated.

Station	1	2	3	4	5	6
Loop	1	2	2	1	1	2
AGV capacity	1	3	3	1	1	3

Station	1	2	3	4	5	6
Loop	1	1	2	2	1	2
AGV capacity	1	3	3	1	1	3

Fig. 5. First step of generating neighborhood

As it is illustrated in **Error! Reference source not found.**, station 2 is moved from loop 2 to loop 1, and station 4 is moved from loop 1 to loop 2. Nevertheless, a mistake is apparently observed in the type of AGV in each loop. To solve this problem and having the best assignments, we reassign AGV to each loop by checking all the possible solutions and finally generating the answer, as presented in Fig 6.

Station	1	2	3	4	5	6
Loop	1	2	2	1	1	2
AGV capacity	1	3	3	1	1	3

Station	1	2	3	4	5	6
Loop	1	1	2	2	1	2
AGV capacity	1	1	3	3	1	3

Fig. 6. Second first step of generating neighborhood

4.4. Stopping criteria

Different criteria can be specified to this end, including stopping condition, maximum iteration, solving time, minimum improvement between two iteration and so on. In the proposed algorithm of this study, maximum iteration is used, as the requirement for stopping small-sized problems and limited time condition is mentioned to be effective for medium and large-sized problems.

5. Numerical Example

To test the proposed programming model and performance of VNS algorithm, the data obtained from the related literature was used, such as (R.Tavakkoli-Moghaddam et al., 2008). The problems were divided to three scenarios, p1-p5 as small-sized problems, p6-p9 as medium-sized problems and p10-p11 as large-sized problems (see). Furthermore, numeration algorithm is used as the third solving method to check the results and deviation. This algorithm is time-consuming; however, since it counts all possible results, it is a good technique for checking other algorithm solutions.

Table 1
Parameters' value for instances (PC: 6.7 GHz, 4 GB RAM)

instances	parameters					
	n	N	η	η'	t	T
P1	6	2	0.25	0.75	{1,1,1,1,1,1}	2000
P2	7	2	0.2	0.8	{2,1,5,1,3,1,1}	1000
P3	8	2	0.25	0.75	{2,1,5,1,3,1,1,3}	4000
P4	9	2	0.2	0.8	{2,1,5,1,3,1,1,3,4}	4000
P5	10	2	0.15	0.85	$1\ \forall\ i$	5000
P6	12	3	0.09	0.57	$1\ \forall\ i$	5000
P7	15	3	0.09	0.57	$1\ \forall\ i$	5000
P8	17	3	0.09	0.57	$1\ \forall\ i$	5000
P9	20	3	0.09	0.57	$1\ \forall\ i$	5000
P10	24	4	0.05	0.45	$1\ \forall\ i$	5000
P11	30	5	0.02	0.38	$1\ \forall\ i$	5000

This study needs to define w_p, r and λ. w_p is the performance coefficient of AGV type P. This coefficient does not have linear relationship with AGV capacity and can differ according to variations, such as loading and unloading policies. Thus, to determine this parameter, uniform distribution was used to define AGV capacity. This value defined as 1 for AGV with one load capacity; but for AGV with 2 load capacity, it was a random number between [1.5, 2]; and for AGV with 3 load capacity, it was a random number between [2.3, 3]. These values considered as 1.8 and 2.65, respectively. λ is a penalty that applies to the AGV with further capacity. In the solving procedure, different values of λ will be analyzed and the effect of these changes on subsequent results will be assessed. By determining the value of the constraints penalty to be $r = 1$, the first 5 problems are solved.

Table 2
Small-sized problem for λ=1

	GAMS	Time(sec)	Best VNS	Max deviation	Time(sec)	Enumeration	Time(sec)
P1	901.82	3	901.82	0	less than 1	901.82	less than 1
P2	126.8	5	126.8	0	less than 1	126.8	1.2
P3	181.77	25	178.866	3.11	less than 1	178.39	2.7
P4	2337.65	50	2254.6	26.6	less than 1	2254.6	6.2
P5	2948.77	67	2769.7	159.1	less than 1	2769.7	15.37

Table 2
Small-sized problem for λ=1

	GAMS	Time(sec)	Best VNS	Max deviation	Time(sec)	Enumeration	Time(sec)
P1	901.82	3	901.82	0	less than 1	901.82	less than 1
P2	126.8	5	126.8	0	less than 1	126.8	1.2
P3	181.77	25	178.866	3.11	less than 1	178.39	2.7
P4	2337.65	50	2254.6	26.6	less than 1	2254.6	6.2
P5	2948.77	67	2769.7	159.1	less than 1	2769.7	15.37

As it is seen in, along with an increase in the dimensions of the problems, the solving time increases, and the exact solvers get stuck in a local optimum. Meanwhile, regarding the constraints, the proposed numeration algorithm generates better solutions in comparison to LINDO solver in GAMS software. Thus, we use this algorithm to check deviations from metaheuristic and LINDO solver answers. All the generated results, for the problems p1 and p2, are the same and optimum. However, by an increase in dimensions of problems, GAMS solver gets stuck in local optimum, and it is not optimum anymore. By checking GAMS solver results, it can be inferred that in line with an increase in dimension of problems, the solving time in this method has increased in a nonlinear manner and it is correspondingly stuck in local optimum.

Numeration algorithm is a comprehensive method to cover all solutions. Nonetheless, along with an increase in problem dimensions, this algorithm will be time-consuming, and its usage will not be rational anymore. For instance, all possible answers to the problem p5 is $3^{10} \times 2^{10}$. However, some answers are impossible and will be ignored in the solving process, but the overall solving time will increase considerably in medium and large-sized problems.

Comparatively, VNS algorithm has good performance for the first 5 problems, as long as it is dealing with fixed number of iterations and limiting the solving time under 1 second. This algorithm has been used 10 times with a fixed number of 100 iteration for each of the (p1-p5) problems. It generated optimum answer in the best iterations. Therefore, it may be evidently stated that this algorithm will generate solutions, when there is an increase in iterations, along with an increase in problems dimensions.

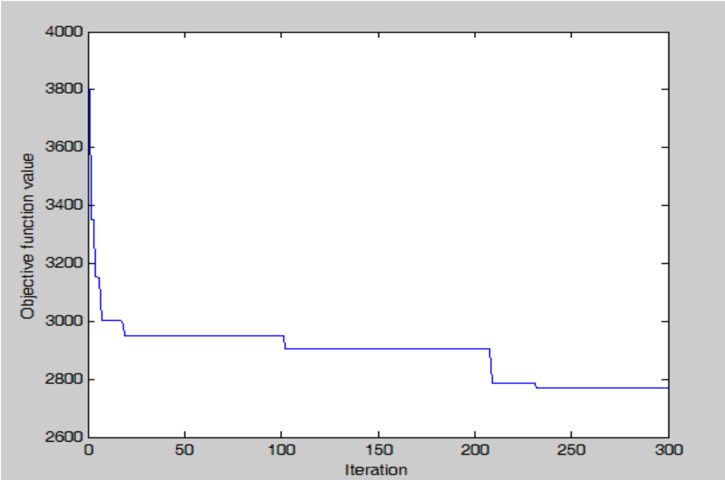

Fig. 7. Convergence in the solving algorithm

Analysis of the VNS solving process for problem p5 (see Fig 7) shows that the solving process is convergent, and as time elapses, the result improves. Furthermore, analyses of Fig. **8** shows that after the 100th iteration, considerable variations can be observed. These high variations are resulted from changes in neighborhood level, which causes better search in different areas to find global optimum. It might be worth pointing out that, these changes in neighborhood level and results are also seen in iteration 200 to 300.

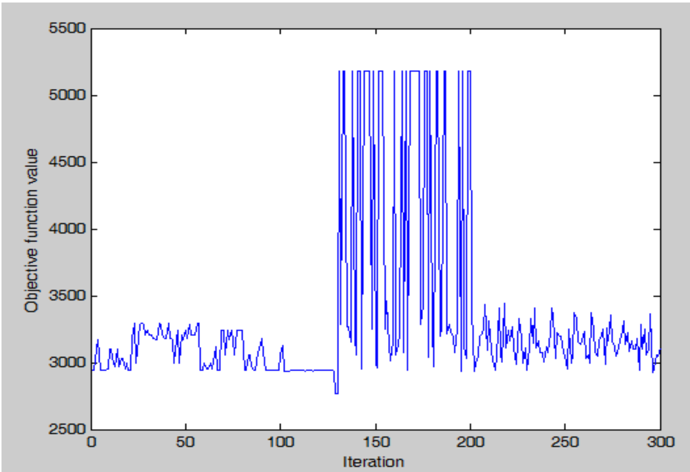

Fig. 8. Changing neighborhood level to escape from local optimums

Before solving the next two scenarios, changes in λ were carefully analyzed. As shown in p6, when $\lambda = 1$, the answer will be as shown in Fig 9:

Station	1	2	3	4	5	6
Loop	2	2	2	1	1	2
AGV capacity	3	3	3	3	3	3

Fig.9. Optimum answer of p6 for λ=1

The case of small-sized problems may differ, for instance in problem p1, the two stations 4 and 5 are in one loop and the others are in loop two. However, a three load AGV is assigned to both loops, which can be revised by different management approaches. For example if we consider $\lambda = 1000$, the answer will be as presented in Fig 10.

Station	1	2	3	4	5	6
Loop	1	1	1	2	2	2
AGV capacity	1	1	1	1	1	1

Fig. 10. Optimum answer of p6 for λ=1000

By an increase in penalty cost, as presented in Fig 10. , the objective function tends to reduce the related penalty cost, thus it assigns one load AGV to each loop. It is necessary to tune λ in proportion to the respective increase in cost of load or management. In this way, we will deal with a suitable assignment and at the same time, there will be a cost reduction proportional to the AGV capacity. In another example, we consider $= 200$, and the answer will be as presented in Fig 11. in which the assignment seems logical.

Station	1	2	3	4	5	6
Loop	1	1	1	2	2	2
AGV capacity	1	1	1	1	1	1

Fig. 11. Optimum answer of p6 for λ=200

In the next three examples, which are medium-sized problems, there is an increase in loops number and the solving time and answers are analyzed. As it was observed in small-sized problems, along with an increase in dimension, there was an increase in solving time accordingly, and solver was stuck in local optimum. In medium-sized problems, we act in a similar manner, but termination criteria were changed. Therefore, VNS algorithm terminated in 180 seconds and GAMS solver were stopped after 2400 seconds.

The increase in loops number results in high increase in problem dimension, for example in p6, all possible answers are $3^{12} \times 3^{12}$. Along with these changes, extreme increase is observed in solving time and disability of GAMS solver in a limited duration of time.

Table 3
Medium-sized problem for λ=1

	GAMS	Time(sec)	Best VNS	Max deviation	Time(sec)	Enumeration	Time(sec)
P6	-	*	4709.7	10.1	180	4709.7	5687.5
P7	-	*	2689.1	-	180	-	*
P8	-	*	3556.8	-	180	-	*
P9	-	*	4127.1	-	180	-	*

In the next three examples, which are medium-sized problems, there is an increase in loops number and the solving time and answers are analyzed. As it was observed in small-sized problems, along with an increase in dimension, there was an increase in solving time accordingly, and solver was stuck in local optimum. In medium-sized problems, we act in a similar manner, but termination criteria were changed. Therefore, VNS algorithm terminated in 180 seconds and GAMS solver were stopped after 2400 seconds.

The increase in loops number results in high increase in problem dimension, for example in p6, all possible answers are $3^{12} \times 3^{12}$. Along with these changes, extreme increase is observed in solving time and disability of GAMS solver in a limited duration of time.

Table 3
Medium-sized problem for λ=1

	GAMS	Time(sec)	Best VNS	Max deviation	Time(sec)	Enumeration	Time(sec)
P6	-	*	4709.7	10.1	180	4709.7	5687.5
P7	-	*	2689.1	-	180	-	*
P8	-	*	3556.8	-	180	-	*
P9	-	*	4127.1	-	180	-	*

Table 4 shows that GAMS solver after 2400 seconds is not able to generate solution for medium-sized problems. In addition, enumeration algorithm for p6 obtains the best solution after about 5700 seconds but our proposed VNS algorithm obtains the best solution in 60 percent of times when there is a limitation of 180 seconds.

In large-sized problems, neighborhood level is increased to 4, and in the same vein, the VNS algorithm run time is increased to 300 seconds. In these problems, GAMS solver and enumeration algorithm are unable to generate answers in a limited time. However, through application of our effective proposed VNS algorithm, we can be hopeful to obtain the desired results in a limited time (see).

Table 4
Large-sized problem for λ=1

	GAMS	Time(sec)	VNS	Time(sec)	Enumeration	Time(sec)
P10	-	*	7158.2	180	-	*
P11	-	*	7156.5	180	-	*

Throughout this research, the presented programming model was tested, using three different solving method. It was shown that along with an increase in problem dimension, the related solving time was increased. For small-sized problems, our proposed VNS algorithm appeared to have a good performance in comparison to enumeration method and GAMS solver. Standard deviation was applied to compare the generated answers for small-sized problems (see Fig 12)

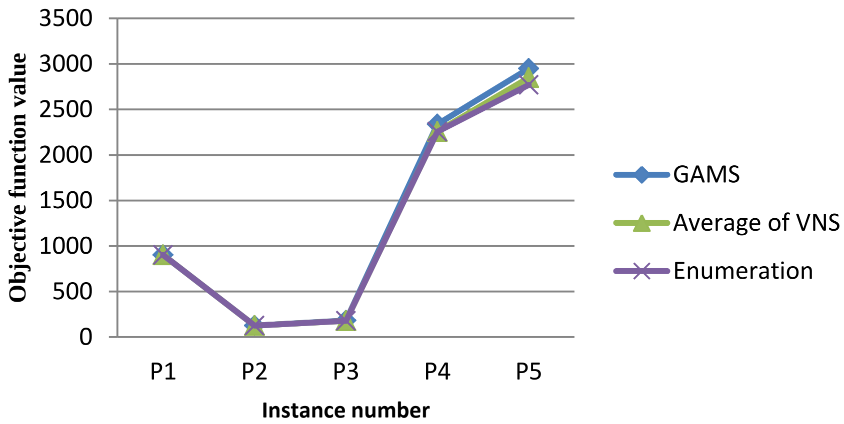

Fig. 12. Analyze p1-p5 results

Furthermore, a comparison of solving times was made in these methods (see Fig. **13**). Evidently, iterations were constant in VNS algorithm in small-sized problems and only a change in the dimension was observed. Hence, there is a small variation of VNS solving times.

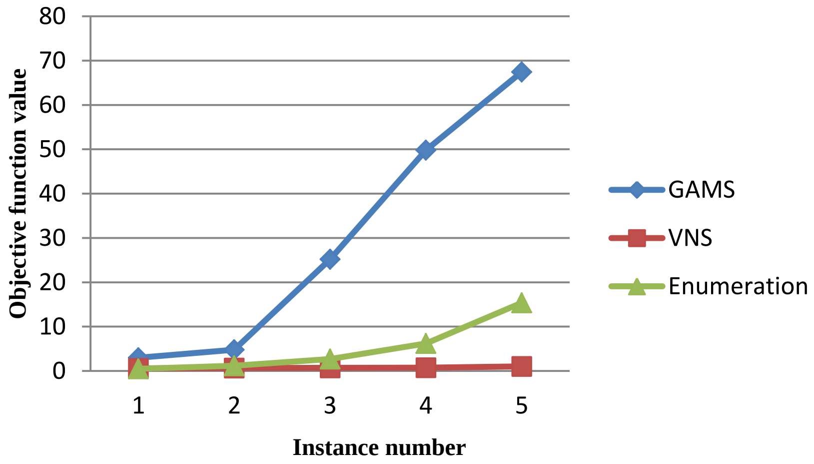

Fig. 13. Analyze p1-p5 solving time

Finally, comparing single load AGV's best solutions with multiple-load AGV's for the same problems (Table 5**Error! Reference source not found.**

Table 5
Comparing single load AGV's best solutions with multiple-load AGV's

	Single load AGV	Multiple-load AGV	Improvement (%)
P1	1467	901.82	0.39
P2	192	126.8	0.34
P3	266	178.866	0.33
P4	3449.8	2254.6	0.35
P5	4456	2769.7	0.38
P6	5976	4709.7	0.21
P7	3560	2689.1	0.24
P8	4644	3556.8	0.23
P9	5565	4127.1	0.26
P10	8851	7158.2	0.19
P11	12688	7156.5	0.44

6. Conclusion

This paper proposed a nonlinear mathematical programming model for multiple-load AGV system in tandem configuration. The proposed model had three goals: minimizing intra-loop transportation, minimizing inter-loops transportation, balancing workload between different loops and assigning AGV with appropriate load capacity to each loop. Since this problem is NP-hard, a modified Variable Neighborhood Search (VNS) algorithm was proposed. In addition, two other methods, i.e. GAMS solver and enumeration algorithm, were applied to check the performance of VNS. Analyses showed that GAMS and enumeration algorithm are good for small-sized problems. Nonetheless, when the dimension of the algorithm increased, they become time-consuming, and got stuck in local optimums, where they were not effective anymore. When a comparison was made between small-sized problems and other methods, VNS algorithm results confirmed the performance of this algorithm. Hence, it is rational to apply this algorithm to medium- and large- sized problems. For further research, it is suggested to investigate the simulation of different loading and unloading policy in comparison to each other. Finally, this paper proposed an effective programming model. This model can be revised concerning multi-period production to make it more realistic and effective.

References

Aarab, A., Chetto, H. and Radouane, L. (1999) 'Flow path design for AGV systems', *Studies in Informatics and Control*, 8, pp. 97–106.

Bozer, Y. A. and Srinivasan, M. M. (1989) 'Tandem configurations for AGV systems offer simplicity and flexibility', *Industrial Engineering*. American Institute of Industrial Engineers, 21(2), pp. 23–27.

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*. Elsevier, 63(2), pp. 173–191.

Carlo, H. J., Vis, I. F. A. and Roodbergen, K. J. (2014) 'Transport operations in container terminals: Literature overview, trends, research directions and classification scheme', *European Journal of Operational Research*, 236(1), pp. 1–13. doi: 10.1016/j.ejor.2013.11.023.

Choobineh, F. F., Asef-Vaziri, A. and Huang, X. (2012) 'Fleet sizing of automated guided vehicles: a linear programming approach based on closed queuing networks', *International Journal of Production Research*, 50(12), pp. 3222–3235.

Christofides, N., Mingozzi, A. and Toth, P. (1980) 'Contributions to the quadratic assignment problem', *European Journal of Operational Research*, 4(4), pp. 243–247. doi: http://dx.doi.org/10.1016/0377-2217(80)90108-3.

Das, S. K. and Pasan, M. K. (2016) 'Design and Methodology of Automated Guided Vehicle-A Review', *IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)*, pp. 29–35.

ElMekkawy, T. Y. and Liu, S. (2009) 'A new memetic algorithm for optimizing the partitioning problem of tandem AGV systems', *International Journal of Production Economics*, 118(2), pp. 508–520.

Fan, X., He, Q. and Zhang, Y. (2015) 'Zone Design of Tandem Loop AGVs Path with Hybrid Algorithm', *IFAC-PapersOnLine*. Elsevier, 48(3), pp. 869–874.

Fazlollahtabar, H. and Saidi-Mehrabad, M. (2015) 'Methodologies to Optimize Automated Guided Vehicle Scheduling and Routing Problems: A Review Study', *Journal of Intelligent & Robotic Systems*, 77(3–4), pp. 525–545. doi: 10.1007/s10846-013-0003-8.

Fazlollahtabar, H., Saidi-mehrabad, M. and Balakrishnan, J. (2015) 'Mathematical optimization for earliness / tardiness minimization in a multiple automated guided vehicle manufacturing system via integrated heuristic algorithms', *Robotics and Autonomous Systems*, 72, pp. 131–138. doi: 10.1016/j.robot.2015.05.002.

Fazlollahtabar, H. and Saidi—Mehrabad, M. (2015) 'Risk assessment for multiple automated guided vehicle manufacturing network', *Robotics and Autonomous Systems*. Elsevier B.V., 74(December 2016), pp. 175–183. doi: 10.1016/j.robot.2015.07.013.

Gilmore, P. C. (1962) 'Optimal and suboptimal algorithms for the quadratic assignment problem', *Journal of the Society for Industrial & Applied Mathematics*. SIAM, 10(2), pp. 305–313.

Gutta, P. R. *et al.* (2018) 'A Review On Facility Layout Design Of An Automated Guided Vehicle In Flexible Manufacturing System', *Materials Today: Proceedings*. Elsevier, 5(2), pp. 3981–3986.

Hansen, P. and Mladenovi, N. (2001) 'Variable neighborhood search : Principles and applications', 130.

Ho, Y.-C., Liu, H.-C. and Yih, Y. (2012) 'A multiple-attribute method for concurrently solving the pickup-dispatching problem and the load-selection problem of multiple-load AGVs', *Journal of manufacturing systems*, 31(3), pp. 288–300. doi: http://dx.doi.org/10.1016/j.jmsy.2012.03.002.

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*. Taylor & Francis, 42(4), pp. 801–832. doi: 10.1080/00207540310001602874.

Kim, K., Chung, B. and Jae, M. (2003) 'A design for a tandem AGVS with multi-load AGVs', *The International Journal of Advanced Manufacturing Technology*, 22(9), pp. 744–752.

Kim, K. S. and Chung, B. Do (2007) 'Design for a tandem AGV system with two-load AGVs', *Computers & Industrial Engineering*, 53(2), pp. 247–251.

Kumar, R. *et al.* (2015) 'Automated guided vehicle configurations in flexible manufacturing systems: a comparative study', *International Journal of Industrial and Systems Engineering*. Inderscience Publishers (IEL), 21(2), pp. 207–226.

Liu, Z. *et al.* (2018) 'A co-evolutionary design methodology for complex AGV system', *Neural Computing and Applications*. Springer, 29(4), pp. 959–974.

Meer, J. R. van der (2000) *Operational control of internal transport*. Erasmus University Rotterdam.

MS.Bazaraa, H. S. (1993) *Nonlinear programming*. Second edn. Wiley, NY.

Ozden, M. (1988) 'A simulation study of multiple-load-carrying automated guided vehicles in a flexible manufacturing system', *The International Journal Of Production Research*. Taylor & Francis, 26(8), pp. 1353–1366.

R.Tavakkoli-Moghaddam H.Kazemipoor, A.Salehipour, M. B. A. (2008) 'Partitioning machines in tandem AGV systems based on "balanced flow strategy" by simulated annealing', *The International Journal of Advanced Manufacturing Technology*. Springer-Verlag, 38(3–4), pp. 355–366. doi: 10.1007/s00170-007-1094-9.

Rezapour, S., Zanjirani-Farahani, R. and Miandoabchi, E. (2010) 'A machine-to-loop assignment and layout design methodology for tandem AGV systems with single-load vehicles', *International Journal of Production Research*. Taylor & Francis, 49(12), pp. 3605–3633. doi: 10.1080/00207543.2010.489056.

Salehipour, A., Kazemipoor, H. and Moslemi Naeini, L. (2011) 'Locating workstations in tandem automated guided vehicle systems', *The International Journal of Advanced Manufacturing Technology*, 52(1), pp. 321–328.

Shalaby, M. A., El Mekkawy, T. Y. and Fahmy, S. A. (2006) 'Zones formation algorithm in tandem AGV systems: a comparative study', *International Journal of Production Research*, 44(3), pp. 505–521.

Talbi, E.-G. (2009) *Metaheuristics: from design to implementation*. John Wiley & Sons.

Vis, I. F. A. (2006) 'Survey of research in the design and control of automated guided vehicle systems', 170, pp. 677–709. doi: 10.1016/j.ejor.2004.09.020.

Yu, W. and Egbelu, P. J. (2001) 'Design of a variable path tandem layout for automated guided vehicle systems', *Journal of manufacturing systems*, 20(5), pp. 305–319.

Zanjirani Farahani, R. *et al.* (2008) 'Designing efficient methods for the tandem AGV network design problem using tabu search and genetic algorithm', *The International Journal of Advanced Manufacturing Technology*, 36(9), pp. 996–1009.

This article can be cited: Rahimikelarijani, B., Saidi-Mehrabad, M. & Barzinpour, F. (2020). A Mathematical Model for Multiple-load AGVs in Tandem Layout. *Journal of Optimization in Industrial Engineering*. 13 (1), 67-80.

http://www.qjie.ir/article_539592.html
DOI: 10.22094/JOIE.2018.537.37