

## Research Article

# A Multiobjective Optimization Algorithm to Solve the Part Feeding Problem in Mixed-Model Assembly Lines

Masood Fathi,<sup>1</sup> Maria Jesus Alvarez,<sup>1</sup> Farhad Hassani Mehraban,<sup>2</sup> and Victoria Rodríguez<sup>3</sup>

<sup>1</sup> Department of Industrial Organization, School of Engineering (TECNUN), University of Navarra, P° Manuel Lardizabal 13, 20018 San Sebastian, Spain

<sup>2</sup> Department of Management, King's College London, 150 Stamford Street, London SE1 9NH, UK

<sup>3</sup> Economics and Management School, University of Navarra, Campus Universitario, 31080 Pamplona, Spain

Correspondence should be addressed to Masood Fathi; [fathi.masood@gmail.com](mailto:fathi.masood@gmail.com)

Received 27 November 2013; Accepted 6 February 2014; Published 20 May 2014

Academic Editor: Andrzej Swierniak

Copyright © 2014 Masood Fathi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Different aspects of assembly line optimization have been extensively studied. Part feeding at assembly lines, however, is quite an undeveloped area of research. This study focuses on the optimization of part feeding at mixed-model assembly lines with respect to the Just-In-Time principle motivated by a real situation encountered at one of the major automobile assembly plants in Spain. The study presents a mixed integer linear programming model and a novel simulated annealing algorithm-based heuristic to pave the way for the minimization of the number of tours as well as inventory level. In order to evaluate the performance of the algorithm proposed and validate the mathematical model, a set of generated test problems and two real-life instances are solved. The solutions found by both the mathematical model and proposed algorithm are compared in terms of minimizing the number of tours and inventory levels, as well as a performance measure called workload variation. The results show that although the exact mathematical model had computational difficulty solving the problems, the proposed algorithm provides good solutions in a short computational time.

## 1. Introduction

In the contemporary business environment, assembly line designs have been following mixed-model assembly to respond to a variety of products. In general, in a mixed-model assembly line, for different versions, there are likely variations associated with base products.

In this era, and for automotive manufacturers in particular, mixed-model assembly lines are employed to produce a variety of submodels of the same automobile. Despite the many advantages of mixed-model assembly lines and its widespread use across the manufacturing plants, supplying these high-variant mixed-model lines has become a critical issue for managers as a huge number of parts/materials must be transferred to a location near the line (at stations) [1]. Moreover, a strong desire to provide an efficient Just-In-Time (JIT) parts supply, which aims to synchronize the supply of parts with their demand while avoiding shortages, has

become another difficulty as any shortage of parts might result in line stoppage and presumably an interruption of the production process. To deal with the challenges of this type as well as increase the reliability and flexibility of the part feeding process, a new concept—the so-called “supermarket”—was introduced, and it is utilized by many world-class manufacturers.

The supermarket is a decentralized logistics area near the assembly line where all parts/materials are sorted. This decentralized in-house logistics area enables the manufacturers (especially those who are dealing with high-volume production) to ensure accessibility for a reliable small-lot JIT part delivery in assembly lines [2]. In particular, based on predefined production timelines, parts in the supermarket are transported to the shop floor in small bins and by means of tow trains (consisting of a small towing vehicle and a few wagons).

Although applying the supermarket concept has several advantages, planning the in-house logistics concept is a complex operation that could be implemented in four interrelated decision problems, as follows [3–5].

- (1) Decisions regarding the number and location of decentralized in-house logistics areas (supermarket).
- (2) Determining the number of tow trains in each supermarket as well as assigning the line segments to them, which can be considered as a tow train vehicle routing problem.
- (3) Determining a Delivery Schedule (DS) for each tow train.
- (4) Decisions regarding the bins and the quantity to be loaded on a tow train for each tour, which is known as the Tow Train Loading (TTL) problem.

The last two problems, which address optimal loading of tow trains and optimal delivery schedule, are most likely to be considered the main optimization criteria in the context of the Assembly Line Part Feeding Problem (ALPFP), which is the focus of this paper.

The rest of the paper is structured as follows. In Section 2, a brief literature review is provided and an explanation of the empirical case study in an automobile assembly plant is given in the third section. In Section 4, a new mathematical model is introduced to provide context for the novel simulated annealing algorithm-based heuristic presented in the fifth section. The computation results for both the mathematical model and the algorithm are reported in Section 6. Finally, conclusions and future research direction are discussed in Section 7.

## 2. Literature Review

According to Emde and Boysen [3], there are less than a handful of studies addressing the Delivery Schedule (DS) and Tow Train Loading (TTL) problems in the literature. While some similarities can be found in previous studies, such as fleet sizing [6] and inventory routing problems [7], none of them cover the DS and TTL problems. Those few remarkable and recent studies related to the ALPFP are reported below.

Emde et al. [5] studied the tow train loading (TTL) problem by introducing an exact polynomial-time solution while the routes and schedule were given and the tugger capacity was assumed to be limited. The aim of this study was to minimize the inventory level near the line by optimally loading the tow trains while shortage was not allowed. Rao et al. [8] presented an optimization model for scheduling the single vehicle (comparable to tugger train) in order to deliver parts from the storage centre to the workstations in a mixed-model assembly line. The aim of the study was to minimize the total inventory holding and travelling costs by suggesting and applying a backward-backtracking approach and a hybrid genetic algorithm and simulated annealing (GASA). Emde and Boysen [4] presented an exact solution procedure (which was a nested dynamic programming algorithm) with a polynomial run time to cope with the routing and scheduling

problem jointly, and the aims of the study were considered to be the minimization of the stock level at the assembly line as well as the number of tow trains. Golz et al. [1] introduced a heuristic procedure to deal with routing, scheduling, and the tow train loading problems simultaneously. This study took the supermarket concept into consideration, and the main aim was to minimize the number of drivers (those involved in a part feeding process) as a consequence of minimizing the number of vehicles while shortage was not allowed. Choi and Lee [9] treated the routing, scheduling, and tow train loading problems jointly by presenting a local search procedure. The presented procedure was tested in a real-world case in the automobile industry with the aim of minimizing the deviation of optimal delivery times per bin delivery.

A review of the literature revealed that while there are few studies in the area of ALPFP with the supermarket concept and increasing attention has been paid to this topic by the scientific community in the past year, there is no reported study treating the DL and TTL problems simultaneously while considering the number of tours and inventory level as objectives. Therefore, this study is aimed at dealing with the part feeding problem at mixed-model assembly lines through solving both decision problems (TTL and DS) simultaneously based on the JIT-supermarket concept and dealing with two conflicting objectives (number of tours and inventory level).

The ALPFP studied here can be described as a set of bins in which all bins are packed into a number of wagons with respect to possible objectives. In this sense, the ALPFP can be considered as a special case of the bin packing problem, where a set of items should be packed into a minimum number of bins. In addition, there are also some similarities between the ALPFP and the dynamic lot sizing problem. In the dynamic lot sizing problem, the main aim is to find the best order amount of products while the demand is known in each period. Furthermore, the order amount of products in this problem varies, and there is a setup cost for each order and inventory holding cost per item. However, in ALPFP, the aim is to find the best order amount of different parts in each tour assuming that the total demand of each part is known in the planning horizon. In fact, in ALPFP, the order amounts vary in different tours and depend on the decisions made in previous tours.

According to Eilon and Christofides [10] and Parreño et al. [11], the bin packing problem is considered to be NP-hard in the strong sense for the one-dimensional bin packing problem. Leaving the multidimensional analysis to one side, the dynamic lot sizing problem, which is a generalization of the economic order quantity model, has also been proven to be NP-hard in most cases, including general objective functions [12]. Due to the complexity of the ALPFP, which is a combination of the bin packing and the dynamic lot sizing problems, the key focus of the present research is on designing an efficient algorithm to resolve this challenge. Thus, a simulated annealing algorithm, which combines with some new priority heuristics, is presented here. Moreover, a new mathematical model is also provided to clarify the ALPFP as well as support the algorithm presented.

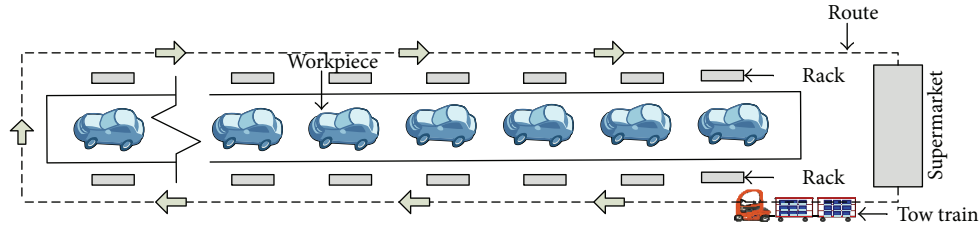


FIGURE 1: A straight assembly line view with the supermarket-concept.

### 3. Case Study Explanation and Problem Description

This study is motivated by a situation on the shop floor at the Volkswagen (VW) in Navarra assembly plant and is based on two mixed-model assembly lines at VW-Navarra. An overview of the part supply process is as follows.

In VW-Navarra, parts are categorized as high demand and low demand. There is a main warehouse and one supermarket, which is close to each assembly line. According to the VW-Navarra part feeding policy, three strategies are available to supply the assembly lines. The first is transporting the parts from the warehouse directly to the assembly line in big containers; the second is transporting the parts in small bins directly from the warehouse to the assembly line; the third is transporting the parts to the supermarket (in the case where excessive operation is required, e.g., repackaging the big containers into small bins to have frequent small-lot deliveries by means of two trains) and supply the assembly stations through the supermarket. Since a big portion of the parts used in the assembly operation is supplied from the supermarket in small bins, the focus of this study is on the third supplying process.

Small bins are fed to the assembly line by means of tow trains, and wagons are loaded with bins in the supermarket. A tour begins at the supermarket, goes to the assembly plant according to a predefined schedule in order to deliver the full bins to the corresponding assembly stations, and then collects the empty bins. The empty tow train returns to the supermarket to be reloaded for the next tour. The tours frequently take place during each working shift with the aim of reducing the inventory level in the assembly line.

The production plan is usually known a few days before production, and thus the exact consumption rate of each part between two consecutive visits at each station can be calculated for the planning horizon. Consequently, bins can be loaded on tow trains in the supermarket with a fixed and predetermined consumption rate for each part, which consequently enables reliable scheduling in the defined planning horizon. The planning horizon is daily and the number of tours each day is 15 (5 tours per shift). The storage capacity restriction is known for each station and all the parts are delivered in small, standardized bins with a specific capacity for each part.

The capacity limit for each wagon of a tow train is almost 50 bins, and no more than two wagons are allowed per tow train. Furthermore, each assembly line is supported

by a single supermarket, which makes it easier and more practical by avoiding any interruption since all the requests and deliveries are managed at a single place near the assembly line.

In addition, the assembly lines are two-sided where both sides of the line (left and right) are used in parallel. A schematic view of the assembly line with a single supermarket is represented in Figure 1.

### 4. Description of the Model

In this section, an optimization model, which is a mixed integer linear programming (MILP) model, is presented based on the VW-Navarra feeding process. The MILP model is a model for optimally loading tow trains and scheduling the deliveries. The improvement criteria in this model are the number of tours and the inventory level. It is worth noting that the improvement criteria and their priority are selected according to our experience at VW-Navarra. The indices, parameters, and decision variables for the MILP model are presented in Table 1.

*The MILP Model.* Consider the following:

$$\text{Min } Z = \alpha \sum_t Y_t + \beta \sum_i \sum_t IL_{it}. \quad (1)$$

Subject to

$$\sum_t^{NT} d_{it} \times \tau_{it} + IL_{i0} \geq \sum_t^{NT} d_{it} \quad \forall i = 1, \dots, n \quad (2)$$

$$d_{it} \times \tau_{it} + IL_{i,t-1} - d_{it} = IL_{it} \quad \forall i = 1, \dots, n, t = 1, \dots, NT \quad (3)$$

$$\sum_{i=1}^n d_{it} \times \tau_{it} \leq A \quad \forall t = 1, \dots, NT \quad (4)$$

$$d_{it} \times \tau_{it} + IL_{i,t-1} \leq C_i \quad \forall i = 1, \dots, n, t = 1, \dots, NT \quad (5)$$

$$\tau_{it} \leq \bar{\tau} \times Y_t \quad \forall i = 1, \dots, n, t = 1, \dots, NT \quad (6)$$

$$\tau_{it} \in [0, \bar{\tau}] \quad \forall i = 1, \dots, n, t = 1, \dots, NT \quad (7)$$

$$P_{it} \geq 0 \text{ and integer } \quad \forall i = 1, \dots, n, t = 1, \dots, NT \quad (8)$$

$$IL_{it} \geq 0 \quad \forall i = 1, \dots, n, t = 1, \dots, NT \quad (9)$$

$$Y_t \in \{0, 1\} \quad \forall t = 1, \dots, NT. \quad (10)$$

The first term of the objective function denotes the number of tours and the second term deals with the inventory level. It should be noted that shortage is not allowed.

The first constraint (2) ensures that the total quantity of reference  $i$  that is transported to the assembly line meets the total demand for reference  $i$ . Equation (3) shows the inventory level of reference  $i$  on each tour and ensures that no stock-outs occur on any tour. Equation (4) ensures that the number of bins assigned to each tow train on each tour does not exceed the capacity of the tow train. Equation (5) ensures that the number of bins delivered on each tour in addition to the available inventory is not more than the capacity of the respective station for reference  $i$  on each tour. Equation (6) ensures that any bin can be delivered on a tour if the tour does not exist. Constraints (7)–(10) define the domain of the variables.

## 5. Solution Methodology

Due to the complexity of ALPFP, it is almost impossible to solve the large-scale problems by using exact optimal approaches. Consequently, the use of heuristic and metaheuristic approaches is one of the most applied solutions for this dilemma. Moreover, according to Fathi et al. [13] and Gourgand et al. [14], most of the constructive procedures are based on priority heuristic rules, and heuristics are the foundation of metaheuristic approaches. Therefore, a number of studies can be found in the literature in which priority heuristic rules are used as the foundation of metaheuristics such as Ant Colony Optimization (ACO), Genetic Algorithm (GA), and Simulated Annealing (SA) in order to cope with different combinatorial optimization problems. For instance, Baykasoğlu [15] proposed an SA algorithm that combines several priority heuristics while Baykasoğlu and Dereli [16] employed priority heuristics as the base of ACO algorithm. Furthermore, Kazemi et al. [17], Baykasoğlu and Özbakir [18], and Haq et al. [19] combined the priority heuristics with GA.

In this study, in order to tackle the ALPFP, we propose a simulated annealing algorithm that is combined with some new priority heuristics. The SA-based heuristic proposed here is presented in two parts. In the first part, some new priority heuristics are presented, and in the second part, these priority heuristics are used as the main core of the SA algorithm.

**5.1. The Priority Heuristics.** In this study, 10 priority heuristics are introduced, and they can all be defined as a single-pass heuristic because only one feasible solution is generated by each of them. Using these heuristics, references are prioritized so they can be assigned to the tow train on each tour based on their particular specifications. Most of the priority heuristics proposed are straightforward; however, the one that tries to prioritize the references based on their criticality requires some further explanation.

In this priority heuristic, calculating the criticality of each reference is mainly done based on the key concept for the

TABLE 1: Notations for the MILP model.

$I$	Set of references
$i$	Index of references ( $i \in I$ )
$T$	Set of tours
$t$	Index of tours ( $t \in T$ )
$IL_{i0}$	Inventory level of reference $i$ at the beginning of the first tour
$A$	Capacity of tow train (volume)
$C_i$	Capacity of correspondence station to reference $i$ (volume)
$d_{it}$	Demand of reference $i$ on tour $t$
$n$	Number of references
$NT$	Number of tours
$\alpha$	Importance coefficient for the first objective
$\beta$	Importance coefficient for the second objective
$P_{it}$	Amount of reference $i$ transported on tour $t$
$IL_{it}$	Inventory level of reference $i$ on tour $t$
$Y_t$	Equals one if tour $t$ is carried out; otherwise, equals 0
$\tau_{it}$	Replenishment rate of reference $i$ on tour $t$ , ( $P_{it}/d_{it}$ )
$\bar{\tau}$	Maximum replenishment rate (to be fixed, e.g., 2)

well-known Critical Path Method (CPM) used in project management studies. The motivation for applying the CPM concept in ALPFP can be explained in this way: in project management, all the tasks that are on the critical path (longest path through the network) have a high priority to perform, and each delay in their performance results in postponing the entire project. Similarly, this criticality concept could have the same importance in the ALPFPs. This would mean that if suitable references are not assigned to the tow trains on each tour, there might be a greater number of tours, which will lead to increases in expenses and the use of human resources.

In such circumstances, the primary aim of the proposed heuristic is to find and assign the critical references to the tow train(s) on each tour. In order to know the criticality of the references, a weight is calculated for each reference, and the higher weight shows the importance of a reference, and it should be assigned prior to the others.

The weight of each reference is calculated according to (11). The rationale behind the formula proposed in (11) is that a reference that has a higher portion of unassigned bins should get a higher weight as compared to the others, and it should be assigned earlier. Otherwise, it may create a bottleneck, which can cause a greater number of tours as follows:

$$w(i) = \frac{TB(i) - TDB(i)}{TB(i) + 1} \quad \forall i = 1, \dots, n, \quad (11)$$

where  $w(i)$  is the weight of reference  $i$ ,  $TB(i)$  is the total number of bins for reference  $i$  that should be delivered within the planning horizon, and  $TDB(i)$  is the total number of delivered bins of reference  $i$ .

A list of proposed priority heuristics and their corresponding details is presented in Table 2.

**5.2. Simulated Annealing-Based Heuristics.** Although there are a variety of metaheuristics in the literature that could be used to address the ALPFP, simulated annealing (SA) was

TABLE 2: Proposed priority heuristics.

Number	Name of the heuristic	Symbol	Definition
1	Most critical reference	MCR	Max.: $w(i)$ $\forall i = 1, \dots, n$
2	Least critical reference	LCR	Min.: $w(i)$ $\forall i = 1, \dots, n$
3	Smallest reference number	SRN	Smallest $i$ $\forall i = 1, \dots, n$
4	Greatest reference number	GRN	Greatest $i$ $\forall i = 1, \dots, n$
5	Maximum storage capacity	MaSC	Max.: $C_i$ $\forall i = 1, \dots, n$
6	Minimum storage capacity	MiSC	Min.: $C_i$ $\forall i = 1, \dots, n$
7	Maximum demand	MaD	Max.: $\sum_{t=1}^{NT} d_{it}$ $\forall i = 1, \dots, n$
8	Minimum demand	MiD	Min.: $\sum_{t=1}^{NT} d_{it}$ $\forall i = 1, \dots, n$
9	Maximum slack	MaS	Max.: $NT - \lceil TB(i)/C_i \rceil^+$ $\forall i = 1, \dots, n$
10	Minimum slack	MiS	Min.: $NT - \lceil TB(i)/C_i \rceil^+$ $\forall i = 1, \dots, n$

selected in this study because of its simplicity and effectiveness in dealing with complicated optimization problems [20–22].

Simulated annealing (SA) is a family of stochastic neighbourhood search methods that are a useful tool for solving large-scale combinatorial optimization problems [23]. The main characteristic of SA is to avoid becoming trapped at a local optimum as it uses a random search that not only accepts a neighbour with a better objective function, but also it accepts a neighbouring solution with a worse objective function by using an acceptance probability [24]. This acceptance probability ( $P_s$ ) is calculated according to (12), which is from Wang et al. [25] as follows:

$$P_s = \exp\left(\frac{\partial Z}{T}\right), \quad (12)$$

where  $T$  is temperature and  $\partial Z$  is the change in the value of the objective function between the two solutions.

In each iteration, if the neighbouring solution is better than the current solution, the neighbouring solution is directly accepted. However, if the neighbouring solution is worse than the current solution, a random number between  $[0, 1]$  is uniformly generated. If the generated number is smaller than or equal to  $P_s$ , the worse solution will be accepted [26]. Moreover, the value of  $T$  in each iteration is computed according to the cooling schedule presented as

$$T_{k+1} = \alpha T_k, \quad (13)$$

where  $\alpha$  is a constant value between zero and one, though it is usually close to one; in this study it is set to 0.95.

Additionally, the initial temperature ( $T_0$ ) is calculated according to the following:

$$T_0 = -\frac{\Delta Z}{\ln P_c}, \quad (14)$$

where  $\Delta Z$  is the difference between the worst and best results for the objective function of a given problem ( $Z_{\text{worst}} - Z_{\text{best}}$ ).  $P_c$  is the initial worst acceptance probability that is normally adjusted between  $[0, 1]$  but very close to one; in this study it is set to 0.98.

There are two other parameters to be set, which are the number of iterations at each temperature (inner loop) and the

final temperature ( $T_f$ ), which is known as the termination criterion of the outer loop. In this study, the number of iterations at each temperature is defined as being equal to the number of part references in each problem, and the final temperature is set to be one. At the same time, both the inner loop and outer loop can also be terminated earlier if the value of the objective function for five consecutive solutions remains unchanged.

*5.2.1. Hybridization of Priority Heuristics and SA.* In this study, the SA algorithm is combined with the proposed priority heuristics so that a string of heuristics is created and the part references are selected for assignment to the tow train on each tour according to the respective heuristic on the string. In fact, the algorithm tries to find the best amount of each reference that should be assigned to the tow train on each tour, and this aim can be achieved through finding the best combination of different heuristic rules in the created solution string. The length of the string is equal to the total number of bins that should be delivered and each element of the string is a number that refers to a priority heuristic with the same number in Table 2. A schematic view of the solution string, for an example, with 10 total bins is presented in Figure 2, where each element of the string includes a number that refers to a priority heuristic with the same number given in Table 2. For instance, the first element in the solution string in Figure 2 is 3, which refers to the smallest reference number (SRN).

After each iteration, the objective function is calculated for the current solution string according to (15), in which both objectives are combined into a single objective function where the lexicographic method is used to prioritize the objectives, and thus a much higher importance is assigned to the first objective (minimizing the number of tours) than the second objective (minimizing the inventory level) due to the real need in the case under study (VW-Navarra), which we believe to be the same in a number of plants as follows:

$$\text{Obj.} = \alpha f'_m(x) + \beta f'_m(x), \quad (15)$$

where  $f'_m(x)$  is the normalized  $m$ th objective value calculated by (16), while  $f_{\text{min}}$  and  $f_{\text{max}}$  are the least desirable value and

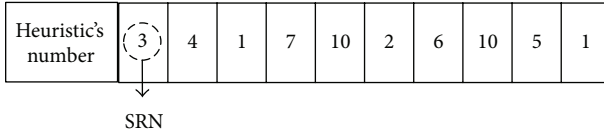


FIGURE 2: Solution string.

the largest value for the corresponding objective, respectively. Moreover,  $f(x)$  is the current value of the following objective:

$$f'_m(x) = \frac{f(x) - f_{\min}}{f_{\max} - f_{\min}}. \quad (16)$$

The solution string is modified iteration by iteration by comparing the calculated objective function for the current solution string and the neighbouring solution string. Moreover, to generate a neighbouring solution, two mutation operators, namely, INSERT and SWAP, are employed with a probability of 0.1. According to Homayouni and Tang [27], SWAP is a commonly used operator in the area of scheduling that helps to generate neighbouring solutions by interchanging the elements of the solution string. Moreover, the INSERT operator is also applied in the algorithm presented here to replace the elements of the solution string with a new value. Examples of both the INSERT and SWAP operators are shown in Figure 3.

To better understand the SA-based heuristics proposed here, an algorithmic description of the solving procedure is presented in the form of pseudocode in two parts. In the first part, there has been an attempt to minimize the number of tours while the initial number of tours is given, and bins are assigned to the tow trains on each tour according to the created solution string (see Algorithm 1). The second part starts while the minimum number of tours found in the first part is an input, and the algorithm tries to minimize the inventory level through assigning the bins to the tow trains as Just-In-Time as possible with the use of the given solution string (see Algorithm 2).

Notations used in the proposed heuristic and their explanations are given in Table 3.

In the second stage of the algorithm, the heuristic seeks to minimize the inventory level in the line through implementing the JIT part supply system. In this regard, all unnecessary bins on each tour are transferred to the latest possible available tour (the tour which is not already removed in the first stage).

For a better understanding of the proposed algorithm, a simple illustrative example is solved in the next subsection.

**5.3. Illustrative Example.** To show the solving process of the proposed algorithm, a simple example is presented and solved as follows.

The number of tours and capacity of the tow train are assumed to be six and ten, respectively. It is also assumed that each reference corresponds to one station, and all the demands and the capacity of the stations are counted based on the bins. The information for the illustrative example is given

TABLE 3: Notations used in the pseudocode.

$h$	Nearest available tour before the $t$ th tour
$TB(i)$	Total number of bins for reference $i$
$AB_{it}$	Number of assigned bins of reference $i$ on tour $t$
$TDB(i)$	Total number of delivered bins of reference $i$
$[B]^+$	The nearest integers greater than or equal to $B$
$[B]^-$	The nearest integers smaller than or equal to $B$
AT	Set of available tours
NA	Number of available tours
$FCL(t)$	First candidate list on the $t$ th tour
$SCL(t)$	Second candidate list on the $t$ th tour
$FICL(t)$	Final candidate list on the $t$ th tour
$RT(t)$	Set of consecutive removed tour(s) after the $t$ th tour
$OB_N$	Objective function of the new solution
$OB_C$	Objective function of the current solution
$C_s$	Current solution
$N_s$	New solution
$B_s$	Best solution
$r$	A uniformly generated random number in the range of $[0, 1]$
NI	Number of iterations

TABLE 4: Illustrative example information.

Reference number	Total demand	Capacity of station	Demand/tour
1	2	2	0.333
2	5	2	0.833
3	3	2	0.5
4	4	3	0.666
5	8	4	1.333

in Table 4. Moreover, a supposed solution string for solving the current example is presented in Figure 4.

### 5.3.1. First Stage

- (1) Start the first tour and create the first candidate list,  $FCL = \{1, 2, 3, 4, 5\}$ . According to the demand per tour in Table 4, one bin of reference 1 to 4 and two bins of reference 5 should be assigned to the tow train without any condition. The total assigned bins are six and there are still four free places in the tow train. Therefore, the second candidate list (SCL) should be created and a reference will be selected for assignment according to the solution string.
- (2)  $SCL = \{1, 2, 3, 4, 5\}$  and the first element in the solution string is 4, which is the “greatest reference number”. Therefore, one bin of reference 5 is assigned to the tow train. There are still three more spaces on the tow train and the second element in the string is 1, which is “most critical reference.” The criticality weight of the references is calculated according to (11)

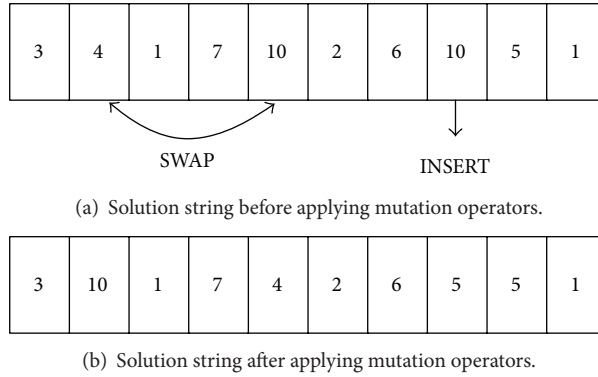


FIGURE 3: INSERT and SWAP operators.

4	1	1	5	7	1	3	4	8	10	7	6	1	9	5	2	5	7	1	10	9	2
---	---	---	---	---	---	---	---	---	----	---	---	---	---	---	---	---	---	---	----	---	---

FIGURE 4: Supposed solution string.

and the weights are  $w(1) \cong 0.33, w(2) \cong 0.66, w(3) = 0.5, w(4) = 0.6, w(5) \cong 0.55$ . Therefore, one bin of reference 2 is assigned to the tow train as it has the highest weight. There are still two more spaces on the tow train, and the updated SCL includes references number 1, 3, 4, and 5. The third element of the string is 1, so therefore the criticality weight of the references is calculated again and the highest weight belongs to reference 4. Thus, one bin of reference 4 is assigned to the tow train. The fourth element of the string is 5, which is “maximum capacity of station.” Therefore, one bin of reference 5 is assigned to the tow train. There is no free space on the tow train and a new tour should be started.

- (3) Start the second tour, updating the station capacity and stock level for each reference according to the consumption rate of each reference (demand per tour). As there are enough inventories for all the references, the FCL is empty. This means that there is no candidate to be assigned to the second tour, and therefore the second tour can be removed.

The assigning process is continued in the same way until all bins are assigned. A summary of the assigning process by the supposed solution string is given in Table 5.

5.3.2. Second Stage

In the first stage, the minimum number of tours is found to be four, and in this stage the aim is to minimize the inventory.

- (1) The second tour is removed, so the assigned bins plus the previous inventory in the first tour should be enough for two tours, and the extra bins should be transferred to the next available tour. In such circumstance, one bin of reference 5 is transferred to the third tour (next available tour).

TABLE 5: The results of assignment process in the first stage of the proposed heuristic.

Tour	Reference number					Sum
	1	2	3	4	5	
1	1	1 + 1	1	1 + 1	2 + 2	10
2	—	—	—	—	—	—
3	0 + 1	1	1 + 1	0 + 2	0 + 2	8
4	—	1	—	—	0 + 2	3
5	—	1	—	—	—	1
6	—	—	—	—	—	—

TABLE 6: Final delivery schedule.

Tour	Reference number					Sum
	1	2	3	4	5	
1	1	2	1	2	3	9
2	—	—	—	—	—	—
3	—	1	1	—	1	3
4	1	1	—	2	2	6
5	—	1	1	—	2	4
6	—	—	—	—	—	—

- (2) As the fourth tour already exists, only the demand for one tour should be covered in the third tour. Therefore one bin of references 1 and 3 and two bins of references 4 and 5 should be transferred to the fourth tour.
- (3) For the same reason, as the fifth tour already exists, the inventory in the fourth tour only needs to be enough for one tour. Thus, one bin of reference 3 and two bins of reference 5 should be transferred to the fifth tour.
- (4) As all bins are assigned and the number of bins assigned to the tow train does not exceed the tow train capacity on any of the tours, the algorithm is terminated.

The final delivery schedule, including the number of bins of each reference to be loaded in the tow trains, is given in Table 6.

```

While  $T_0 > 1$ 
  For  $k = 1$  to NI
    While (i) is not satisfied
      
$$\sum_{i=1}^{NT} AB_{it} \geq \sum_{i=1}^{NT} d_{it} \quad \forall i = 1, \dots, n \quad (i)$$

      Start a new tour,  $m = 1$ , and create the FCL according to (ii).
      
$$i \in \text{FCL}(t) \iff IL_{it} < d_{it} \quad (ii)$$

      If FCL is not empty
        While FCL is not empty
          Select a part reference which is in the FCL and assign as many bins as necessary to
          satisfy (iii). Remove the selected reference from FCL.
          
$$AB_{it} + IL_{it} \geq d_{it} \quad (iii)$$

        End while
        While Equation (iv) is satisfied
          
$$\sum_{i=1}^n AB_{it} < A \quad (iv)$$

          Create the SCL according to (v).
          
$$i \in \text{SCL}(t) \iff [AB_{it} + IL_{it}]^+ < C_i \cap \sum_{p=1}^t AB_{ip} < \sum_{t=1}^{NT} d_{it} \quad (v)$$

          If SCL is not empty
            Use the solution string, select a reference according to the given heuristic on the  $m$ th
            element of the string and assign one bin of the selected reference to the tow train,  $m = m + 1$ .
          Else
            Update the stock level and capacity in the stations according to (vi) and (vii), respectively.
            Terminate the assigning process in the current tour.
            
$$IL_{it+1} = IL_{it} + AB_{it} - d_{it} \quad \forall i = 1, \dots, n \quad (vi)$$

            
$$C_i = [C_i - IL_{it+1}]^- \quad \forall i = 1, \dots, n \quad (vii)$$

          End if
        End while
      Else
        Leave the current tour without assigning any bins and update the stock level and capacity
        in the stations according to (vi) and (vii), respectively.
      End if
    End while
  Continue the solution process in the second part of the algorithm.
  Calculate the objective function
  If  $OB_N \leq OB_C$ 
    
$$C_s = N_s$$

  Else
    Calculate the  $P_s$  according to (12) and generate the  $r$ 
    If  $r \leq P_s$ 
      
$$C_s = N_s$$

    End
  End
  Generate a neighbouring solution using the defined mutation operators.
  Terminate the process in the current inner loop if  $C_s$  is the same for 5 consecutive iterations
  End if
  End for
  If  $C_s \leq B_s$ 
    
$$B_s = C_s$$

  End if
  Reduce the temperature using the decrement function presented as (13).
  Terminate the algorithm if  $B_s$  is the same for 5 consecutive iteration
  End while

```

ALGORITHM 1: First stage: minimizing the number of tours.



Identify the AT and NA according to (viii) and (ix), respectively.

$$t \in AT \iff \sum_i^n AB_{it} \neq 0 \quad \forall t = 1, \dots, NT \quad (\text{viii})$$

$$NA = \sum_{t \in AT} t \quad (\text{ix})$$

**While** AT is not empty  
 Select the first possible tour (which is in the AT) and find the number of removed consecutive tour(s) after the selected tour ( $NR_t$ ) according to (x).

$$t \in RT(t) \iff \sum_i^n AB_{it} = 0 \quad \forall t = t + 1, \dots, NT \quad (\text{x})$$

Keep as many bins as necessary to satisfy (xi) and transfer the extra bins to the next available tour. Update all the information related to the current tour and the tour that received the extra bins. Remove the selected tour from AT.

$$IL_{it} + AB_{it} \geq (NR_t + 1) \times d_{it} \quad (\text{xi})$$

**End while**  
**For**  $t = NA$  to 2  
**If**(xii) is not satisfied

$$\sum_{i=1}^n NA_{it} \leq A \quad (\text{xii})$$

**While**(xii) is not satisfied  
 Identify the nearest available tour before the current tour ( $h$ ) that satisfies (iv) and (xiii).

$$\sum_i^n AB_{it} \neq 0 \quad (\text{xiii})$$

Create the FICL according to (xiv).

$$i \in FICL(t) \iff AB_{it} + [IL_{it}]^- > d_{it} \cap [AB_{ih} + IL_{ih}]^+ < C_i \quad (\text{xiv})$$

Use the solution string and select a reference that is in the FICL and transfer one bin of the selected reference to tour  $h$ .  
 Update the information related to the current tour and the tour that received the selected reference

**End while**  
**End if**  
**End for**

ALGORITHM 2: Second stage: minimizing the inventory level.

## 6. Computational Results

In order to evaluate the efficiency and effectiveness of the proposed heuristic, a set of instances, including nine different problems with different sizes (small, medium, and large), is generated. Moreover, two real cases taken from the Volkswagen Navarra plant in Spain are also presented. All nine generated instances are solved with three different numbers of tours, meaning that 29 instances are solved in total. The set of generated instances and the two real cases studied here can be found at <http://www.tecnun.es/departamentos/doi/investigacion/optimizacion.html>.

The selected criteria for generating the instances are summarized in Table 7. All notations in Table 7 are identical to the notation presented in Tables 1 and 3. Moreover, the theoretical minimum number of tours (represented by  $X$ ) is calculated based on the following:

$$X = \text{Max} \left\{ M(i) = \frac{(TB(i) - 1)}{(C_i - 1)} \quad \forall i = 1, \dots, n, \right. \\ \left. \frac{\sum_{i=1}^n TB(i)}{A} \right\}. \quad (17)$$

It should be noted that the capacity of each wagon of a tow train ( $A$ ) is 50 bins and each tow train has a maximum of two wagons, and in some especial cases three. This limitation is mainly imposed because of the actual situation in the assembly area where there is scarce space on the shop floor and difficult sharp turns. Moreover, the maximum number of tours in the planning horizon (which is one day) is assumed to not be more than 24. This limitation (maximum number of tours per day) is mainly defined based on our experience in case studies, as a frequency of less than one hour is too much effort for both drivers and logistics personnel.

All 29 instances are solved by the proposed algorithm and CPLEX 12.4 to compare and show the efficiency and effectiveness of the proposed heuristic.

The proposed algorithm was coded in MATLAB 2010b and both the algorithm and CPLEX-MATLAB interface are performed on a personal computer with Intel Core i3, a 3.3 GHz CPU, and 8 GB of RAM. Furthermore, as with previous studies (e.g., Kang and Kim [28]), a two-hour running time limitation was applied for CPLEX.

In order to evaluate the performance of the proposed algorithm, the results from the algorithm and the MILP model solved by CPLEX are compared with respect to two improvement criteria (number of tours and inventory level).

TABLE 7: Selected criteria for generating the instances.

Assumption	Large	Medium	Small
NT	$X \leq NT \leq 24$	$X \leq NT \leq 24$	$X \leq NT \leq 24$
$n$	$n > 60$	$30 < n \leq 60$	$n \leq 30$
$C_i$	$(TB/24) \times 3 \leq C_i \leq 10$	$(TB/24) \times 2 \leq C_i \leq 10$	$(TB/24) \times 1.5 \leq C_i \leq 10$
$A$	$\sum_i^n [TB(i)/NT]^+$	$\sum_i^n [TB(i)/NT]^+$	$\sum_i^n [TB(i)/NT]^+$
$TB(i)$	$5 \leq TB(i) \leq 50$	$10 \leq TB(i) \leq 75$	$10 \leq TB(i) \leq 100$

Moreover, to cover all the possible differences in the solutions obtained from both the algorithm and CPLEX, a performance index called Workload Variation ( $V$ ), presented by Hwang et al. [29], is also calculated according to (18), in order to compare the tow trains workload variation. It is obvious that smaller  $V$  value represent smoother workloads as follows:

$$V = \sqrt{\frac{\sum_{t=1}^{NT} (U_t - \overline{LT})^2}{NT}}, \quad (18)$$

where  $\overline{LT} = \sum_{t=1}^{NT} U_t / NT$  is the average utilization of the total tow trains and  $U_t = LT_t / LT_{(t=1, \dots, NT) \max}$  for all  $t = 1, \dots, NT$  is the utilization of tow train(s) on the  $t$ th tour. Moreover,  $LT_t = \sum_{i=1}^n AB_{it}$  for all  $t = 1, \dots, NT$  is the total number of loaded bins on tow train(s) on the  $t$ th tour.

The inventory level presented in Table 8 is the Average Inventory Level (AIL), calculated by

$$AIL = \frac{(\sum_t^{NT} \sum_i^n IL_{it})}{(NT \times n)}. \quad (19)$$

The computational results for both the proposed heuristic and CPLEX are given in Table 8. To facilitate the comparison, the worst results obtained by the SA-based heuristic relative to the results from CPLEX appear in bold, and the best are marked with “\*.” The notations used in Table 8 are the same as Table 1.

Due to the importance of the first objective (number of tours), in Table 8 the comparison of the average inventory level and variation of tow train workload between both CPLEX and the proposed algorithm can be made only if the number of tours is the same.

The analysis of the results given in Table 8 shows that the proposed algorithm provided the optimum number of tours in all solved instances. Moreover, it successfully found the optimum inventory level and variation of tow train workload in 26 and 27 cases out of 29 solved instances, respectively. Additionally, although CPLEX could not find the optimum solution during the limited computation time (2 hours) for 11 instances (which were mostly large-scale problems), the heuristic presented here found an optimum or very nearly an optimum solution in almost all the solved instances, and in a very short computational time. Moreover, in two of the instances where CPLEX was not able to find the optimum solution, the SA-based heuristic algorithm found one less tour as compared to CPLEX. Furthermore, the SA-based heuristic algorithm also found a better value for the variation of tow train workload in one of the solved instances.

To facilitate comparison, the percentage error of the algorithm proposed from CPLEX for the average inventory level has also been presented in Table 8. It can be observed that in roughly 89% of the solved samples, the algorithm found solutions for minimizing the inventory level that were as good as the solutions provided by the MILP model solved by CPLEX.

Regarding the real-cases studied here, it is remarkable that the results of this paper have been productively implemented in VW-Navarra, which has caused the minimization of the number of tours, the inventory level, or both in the assembly lines that were under study.

## 7. Concluding Remarks and Future Directions

In this study, we tried to treat the part feeding problem at a mixed-model assembly line while considering the supermarket concept. Firstly, the optimization model that is considered to be a mixed integer linear programming model was presented based on the actual situation encountered in an automobile assembly plant. Secondly, an SA-based heuristic algorithm was presented, and as a part of the algorithm, a set of new priority heuristics is also presented. In both the MILP model and the SA-based heuristics, the main decision problems to solve were tow train loading and delivery schedule while considering two conflicting improvement criteria with different levels of importance (the number of tours and the inventory level).

Both the MILP model and the SA-based heuristics were tested on two real case studies and a set of generated instances. The results of the comparisons proved that the proposed SA-based heuristic is interestingly efficient as it found a collection of good results in most cases. In fact, the proposed algorithm found the best number of tours as compared to the MILP in all the solved instances. Moreover, it also found good results; around 89% for “average inventory level” and 93% for “variation of tow train workload.” It is also interesting to note that although the computational time for the solved MILP model by CPLEX was significantly higher, the proposed heuristic found a good solution for almost all of the solved instances in a very short time.

This study can serve as a starting point for further research in the area of the assembly line part feeding problem. Moreover, the proposed algorithm can be employed to study other types of assembly lines while considering the same or different improvement criteria with deterministic or stochastic demand. Furthermore, due to the novelty of the ALFPF introduced here, future studies can use

TABLE 8: Results from heuristic and CPLEX for the solved instances.

Sample	$n$	SA-based heuristic							CPLEX			
		NT	A	NT	AIL	PR-AIL (%) <sup>a</sup>	V	CPU (s)	NT	AIL	V	CPU (s)
1	110	24	350	23	0.515	0.000	0.089	2.476	23	0.515	0.089	7200
		23	350	22	0.573*	0.348	0.080	2.548	22	0.575	0.080	7200
		20	350	20	0.413*	0.242	0.056*	2.047	20	0.414	0.057	7200
2	95	24	300	23	0.542	0.000	0.089	2.277	23	0.542	0.089	7200
		23	300	22	0.588	0.000	0.076	2.264	22	0.588	0.076	7200
		19	350	19	0.438	0.000	0.047	1.775	19	0.438	0.047	7200
3	75	24	250	18*	1.255	—	0.124	1.815	19	1.115	0.146	7200
		22	250	20	0.776	0.000	0.104	1.744	20	0.776	0.104	7200
		17	300	17	0.458	0.000	0.046	1.128	17	0.458	0.046	34.937
4	60	24	150	17	<b>1.482</b>	-2.845	<b>0.112</b>	4.562	17	1.441	0.084	7200
		15	200	15	0.398	0.000	0.058	0.748	15	0.398	0.058	4.507
		13	250	13	0.415	0.000	0.053	0.562	13	0.415	0.053	2.062
5	50	22	150	16	<b>1.515</b>	-5.722	<b>0.921</b>	2.147	16	1.433	0.067	7200
		16	200	16	0.433	0.000	0.058	0.916	16	0.433	0.058	4.084
		11	250	11	0.409	0.000	0.041	0.449	11	0.409	0.041	1.928
6	40	23	100	17*	1.986	—	0.117	9.748	18	1.125	0.101	7200
		15	150	15	0.385	0.000	0.059	0.557	15	0.385	0.059	2.097
		12	200	12	0.375	0.000	0.054	0.429	12	0.375	0.054	1.281
7	30	19	100	12	<b>0.983</b>	-0.614	0.220	0.509	12	0.977	0.220	15.786
		17	100	16	0.515	0.000	0.107	0.470	16	0.515	0.107	87.166
		15	100	15	0.406	0.000	0.072	0.498	15	0.406	0.072	1.303
8	20	21	50	13	1.216	0.000	0.088	0.421	13	1.216	0.088	27.511
		16	50	13	1.126	0.000	0.069	0.961	13	1.126	0.069	5.943
		11	100	11	0.386	0.000	0.057	0.211	11	0.386	0.057	0.633
9	10	15	50	8	1.380	0.000	0.122	0.185	8	1.380	0.122	2.019
		11	50	10	0.672	0.000	0.119	0.182	10	0.672	0.119	0.783
		8	50	8	0.343	0.000	0.056	0.101	8	0.343	0.056	0.208
Case 1	26	15	100	13	0.484	0.000	0.169	0.121	13	0.484	0.169	2.076
Case 2	23	15	100	15	0.339	0.000	0.077	0.108	15	0.339	0.077	1.574

<sup>a</sup>Percentage error (PR) of the heuristic algorithm from CPLEX for the average inventory level (AIL).

the other metaheuristic algorithms such as Tabu search, Genetic Algorithm, and Ant Colony to effectively tackle ALPFP and compare their performance with the algorithm presented here. Additionally, MILP model as presented in this paper, would facilitate a solution to various problems in other similar empirical works.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

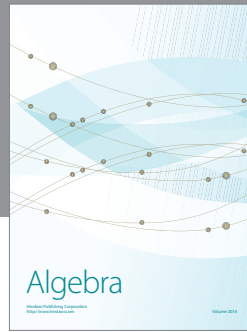
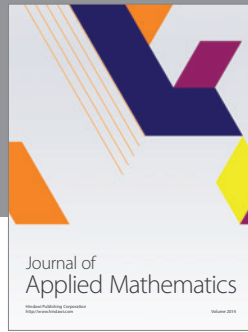
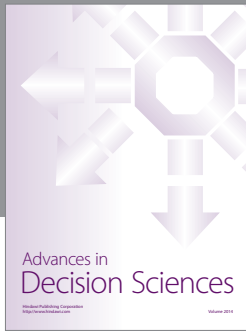
The authors greatly appreciate the collaboration of Volkswagen Navarra, the department of logistics in particular, as well as the Volkswagen Chair Universidad de Navarra for partially funding this research. This work was also partially funded by the regional (Basque) government (Saiotek: MEPLAM) and

the EASY (Energy-Aware Feeding Systems Project), an era-net project of the ECO-INNOVERA 1st Joint Call. The first author would also like to thank the Association of Friends of the University of Navarra for financially supporting his stay at the University of Navarra.

## References

- [1] J. Golz, R. Gujjula, H.-O. Günther, S. Rinderer, and M. Ziegler, "Part feeding at high-variant mixed-model assembly lines," *Flexible Services and Manufacturing Journal*, vol. 24, no. 2, pp. 119–141, 2012.
- [2] D. Battini, N. Boysen, and S. Emde, "Just-in-Time supermarkets for part supply in the automobile industry," *Journal of Management Control*, vol. 24, no. 2, pp. 209–217, 2013.
- [3] S. Emde and N. Boysen, "Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines," *International Journal of Production Economics*, vol. 135, no. 1, pp. 393–402, 2012.

- [4] S. Emde and N. Boysen, "Optimally routing and scheduling tow trains for JIT-supply of mixed-model assembly lines," *European Journal of Operational Research*, vol. 217, no. 2, pp. 287–299, 2012.
- [5] S. Emde, M. Fliedner, and N. Boysen, "Optimally loading tow trains for just-in-time supply of mixed-model assembly lines," *IIE Transactions*, vol. 44, no. 2, pp. 121–135, 2012.
- [6] G. J. Beaujon and M. A. Turnquist, "A model for fleet sizing and vehicle allocation," *Transportation Science*, vol. 25, no. 1, pp. 19–45, 1991.
- [7] J. F. Cordeau, G. Laporte, M. W. P. Savelsbergh, and D. Vigo, "Vehicle routing," in *Handbook in Operations Research and Management Science*, C. Barnhart and G. Laporte, Eds., vol. 14, pp. 367–428, Elsevier, Amsterdam, The Netherlands, 2007.
- [8] Y.-Q. Rao, M.-C. Wang, K.-P. Wang, and T.-M. Wu, "Scheduling a single vehicle in the just-in-time part supply for a mixed-model assembly line," *Computers & Operations Research*, vol. 40, no. 11, pp. 2599–2610, 2013.
- [9] W. Choi and Y. Lee, "A dynamic part-feeding system for an automotive assembly line," *Computers & Operations Research*, vol. 43, no. 1-2, pp. 123–134, 2002.
- [10] S. Eilon and N. Christofides, "The loading problem," *Management Science*, vol. 17, no. 5, pp. 259–268, 1971.
- [11] F. Parreño, R. Alvarez-Valdes, J. F. Oliveira, and J. M. Tamarit, "A hybrid GRASP/VND algorithm for two- and three-dimensional bin packing," *Annals of Operations Research*, vol. 179, no. 1, pp. 203–220, 2010.
- [12] M. Florian, J. K. Lenstra, and A. M. G. R. Kan, "Deterministic production planning: algorithms and complexity," *Management Science*, vol. 26, no. 7, pp. 669–679, 1980.
- [13] M. Fathi, M. K. A. Ariffin, and N. Ismail, "A note on 'A multi-objective genetic algorithm for solving assembly line balancing problem,'" *The International Journal of Advanced Manufacturing Technology*, vol. 50, no. 5–8, pp. 771–773, 2010.
- [14] M. Gourgand, N. Grangeon, and S. Norre, "Metaheuristics based on bin packing for the line balancing problem," *RAIRO—Operations Research*, vol. 41, no. 2, pp. 193–211, 2007.
- [15] A. Baykasoğlu, "Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems," *Journal of Intelligent Manufacturing*, vol. 17, no. 2, pp. 217–232, 2006.
- [16] A. Baykasoğlu and T. Dereli, "Two-sided assembly line balancing using an ant-colony-based heuristic," *The International Journal of Advanced Manufacturing Technology*, vol. 36, no. 5-6, pp. 582–588, 2008.
- [17] S. M. Kazemi, R. Ghodsi, M. Rabbani, and R. Tavakkoli-Moghaddam, "A novel two-stage genetic algorithm for a mixed-model U-line balancing problem with duplicated tasks," *The International Journal of Advanced Manufacturing Technology*, vol. 55, no. 9–12, pp. 1111–1122, 2011.
- [18] A. Baykasoğlu and L. Özbakir, "Stochastic U-line balancing using genetic algorithms," *The International Journal of Advanced Manufacturing Technology*, vol. 32, no. 1-2, pp. 139–147, 2007.
- [19] A. N. Haq, K. Rengarajan, and J. Jayaprakash, "A hybrid genetic algorithm approach to mixed-model assembly line balancing," *The International Journal of Advanced Manufacturing Technology*, vol. 28, no. 3-4, pp. 337–341, 2006.
- [20] V. F. Yu, S.-W. Lin, W. Lee, and C.-J. Ting, "A simulated annealing heuristic for the capacitated location routing problem," *Computers & Industrial Engineering*, vol. 58, no. 2, pp. 288–299, 2010.
- [21] N. Boysen and S. Bock, "Scheduling just-in-time part supply for mixed-model assembly lines," *European Journal of Operational Research*, vol. 211, no. 1, pp. 15–25, 2011.
- [22] H. Avila-George, J. Torres-Jimenez, and V. Hernandez, "New bounds for ternary covering arrays using a parallel simulated annealing," *Mathematical Problems in Engineering*, vol. 2012, Article ID 897027, 19 pages, 2012.
- [23] D.-C. Li and P.-H. Hsu, "Competitive two-agent scheduling with learning effect and release times on a single machine," *Mathematical Problems in Engineering*, vol. 2013, Article ID 754826, 9 pages, 2013.
- [24] W.-H. Wu, "A two-agent single-machine scheduling problem with learning and deteriorating considerations," *Mathematical Problems in Engineering*, vol. 2013, Article ID 648082, 18 pages, 2013.
- [25] G.-G. Wang, L. Guo, A. H. Gandomi, A. H. Alavi, and H. Duan, "Simulated annealing-based krill herd algorithm for global optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 213853, 11 pages, 2013.
- [26] V. Rodriguez, M. J. Alvarez, and L. Barcos, "Hub location under capacity constraints," *Transportation Research E*, vol. 43, no. 5, pp. 495–505, 2007.
- [27] S. M. Homayouni and S. H. Tang, "Multi objective optimization of coordinated scheduling of cranes and vehicles at container terminals," *Mathematical Problems in Engineering*, vol. 2013, Article ID 746781, 9 pages, 2013.
- [28] J.-H. Kang and Y.-D. Kim, "Coordination of inventory and transportation managements in a two-level supply chain," *International Journal of Production Economics*, vol. 123, no. 1, pp. 137–145, 2010.
- [29] R. K. Hwang, H. Katayama, and M. Gen, "U-shaped assembly line balancing problem with genetic algorithm," *International Journal of Production Research*, vol. 46, no. 16, pp. 4637–4649, 2008.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

