Int J Adv Manuf Technol (2014) 75:1809–1827
DOI 10.1007/s00170-014-6153-4

ORIGINAL ARTICLE

Multi-objective artificial bee colony algorithm for simultaneous sequencing and balancing of mixed model assembly line

Ullah Saif · Zailin Guan · Weiqi Liu · Baoxi Wang · Chaoyong Zhang

Received: 1 March 2014 / Accepted: 7 July 2014 / Published online: 30 August 2014

Abstract In recent years, mixed model assembly lines are gaining popularity to produce a variety of models on the single-model assembly lines. Mixed model assembly lines have two types of problems which include sequencing of different models on the line and balancing of assembly line. These two problems collectively affect the performance of assembly lines, and therefore, current research is aimed to balance the workload of different models on each station, to reduce the deviation of workload of a station from the average workload of all the stations and to minimize the total flow time of models on different stations simultaneously. A multi-objective artificial bee colony (multi-ABC) algorithm for simultaneous sequencing and balancing problem with Pareto concepts and local search mechanism is presented. Two kinds of mixed model assembly line problems are analysed. For the first and second problems, each model task time data and precedence relation data are taken from standard assembly line problems, from operation research library (ORL) and from a truck manufacturing company in China, respectively. Both problems are solved using the proposed multi-ABC algorithm on two different demand scenarios of models, and the results are compared against the results obtained from a famous algorithm in the literature, i.e. non-dominated sorting genetic algorithm (NSGA) II. Computational results of the selected problems indicate that the proposed multi-ABC algorithm outperforms NSGA II and gives better Pareto solutions for the selected problems on different demand scenarios of models.

Keywords Mixed model assembly line · Simultaneous sequencing and balancing · Pareto solutions · Artificial bee colony algorithm

U. Saif · Z. Guan · W. Liu · B. Wang · C. Zhang (✉)
State Key Lab of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, Hubei, People's Republic of China
e-mail: Zcyhust@mail.hust.edu.cn

U. Saif
e-mail: Saifullah47@yahoo.com

Z. Guan
e-mail: zlguan@hust.edu.cn

W. Liu
e-mail: lwqi2010@gmail.com

B. Wang
e-mail: wbxjulia@163.com

U. Saif · Z. Guan · W. Liu · B. Wang · C. Zhang
HUST–SANY Joint Laboratory of Advanced Manufacturing Technology, Huazhong University of Science and Technology, Wuhan 430074, Hubei, People's Republic of China

U. Saif
Department of Industrial Engineering, University of Engineering and Technology, Taxila, Pakistan

1 Introduction

Assembly lines are flow-oriented production systems designed to produce a large quantity of products. The assembly line concept is introduced by Ford Manufacturing Company which accelerated the production rate in manufacturing industries. In assembly lines, serially connected stations are arranged around a material handling system from which the required material moves from the upstream to downstream of stations to make different assembly products. Single-model assembly lines can produce a single type of product on large volume. However, due to increase in the customized demand of products, single-model assembly line is utilized to make more than one model of products. These lines which can produce a variety of product models on one line are called mixed model assembly lines. The literature indicates that mixed model assembly lines are concerned with two types of problems. The first problem deals with the assignment of different tasks to the stations to evenly distribute workload among stations, called balancing problem. The second problem is related to sequencing of different models of product on the assembly line, called sequencing problem. These two problems have been investigated separately in the literature [1–6] but suffered from a lack of comprehensive simultaneous deliberations. Karabati and Sayin [7] showed that the total assembly time of products in stations is a function of the model sequence in mixed model assembly lines. Furthermore, the best and effective assembly line balance can be obtained by integrating balancing with model sequencing in these assembly lines. They presented balancing and sequencing together, and in their proposed method, task assignment is performed after model sequencing is obtained. They presented a method in which tasks are allocated to different stations which are well suited with the given sequence of models. There are some other researchers who worked on simultaneously balancing and sequencing problem in mixed model assembly lines [7–18]. Karabati and Sayin [7] investigated mixed model assembly line and considered simultaneous balancing and sequencing problem with an objective to minimize cycle time. Kim et al. [13] studied simultaneous balancing and sequencing of mixed model assembly line and optimized absolute deviation in workload (ADW). Mosadegh et al. [14] presented simultaneous balancing and sequencing in mixed model assembly line with an objective to minimize total utility work. Nevertheless, most of the researches on simultaneous balancing and sequencing of mixed model assembly line problem considered optimization of single objective. However, in real production environment, more than one objective is desired to be optimized concurrently. In the literature, few research articles have studied more than one objective in balancing and sequencing problems of assembly lines. For example, Merengo et al. [19] considered balancing and sequencing of mixed model assembly line with more than one objective. They considered minimization of the rate of incomplete jobs and work in process (WIP) in mixed model assembly line. They presented separate algorithms for balancing and sequencing objectives to solve balancing and sequencing problems independently. They considered single-objective optimization during balancing problem and also considered single-objective optimization during sequencing problem with their proposed respective algorithms. However, simultaneous consideration of balancing and sequencing objectives together might be more significant to consider because unforeseeable changes in the demand of models might continuously need rebalancing of the lines as suggested by Mosadegh et al. [14] and Kim et al. [8, 9]. Ozcan et al. [15] studied simultaneous balancing and sequencing in mixed model assembly line and considered maximization of line efficiency and minimization of smoothness index as objectives in mixed model assembly line. They developed an approach for multi-objective optimization in which they combined both objectives into a single objective for optimization. Their proposed single objective considered a particular fraction value from each objective to compute the overall objective value. Hwang and Katayama [16] presented simultaneous balancing and sequencing of mixed model assembly line and considered maximization of line efficiency, minimization of the workload variance and minimization of the maximum distance between the average and actual pace of workload as objectives. They combined these three objectives into a single objective with certain ranks given to each objective. Recently, Öztürk et al. [18] considered simultaneous balancing and sequencing in mixed model assembly line. They developed mixed integer programming model, a constraint model, and also introduced a new decomposition scheme to solve balancing and sequencing problem.

Nevertheless, most of the research on multi-objective optimization of balancing and sequencing problems combined their objectives into a single objective for optimization. Some researchers in the literature developed a composite function (i.e. combination of the objectives) for optimization of multiple objectives and obtained a single solution of the problem by optimizing their presented composite function [15, 16]. However, optimization of a composite function may not give surety that which of the objective from all objectives of the problem is significantly optimized. In most of the assembly lines, these multiple objectives are conflicting and performance of each objective may not be improved without sacrificing the performance of at least one. Moreover, solution of multi-objective problems exists in the form of a set of alternative trade-off solutions called Pareto optimal set and is therefore desired to be considered in the multi-objective problems. Recently, Yang et al. [20] presented multi-objective genetic algorithm using the Pareto concept for mixed model assembly line rebalancing problem to minimize the number of stations, variation of workload of different models on stations and rebalancing costs simultaneously. However, they have addressed optimization of multiple objectives of only balancing problem and have not considered model sequencing issue in mixed model assembly line. Therefore, it is desired to consider multi-objective optimization of simultaneous balancing and sequencing of mixed model assembly lines to include both assembly line balancing and sequencing objectives together. This motivates to present here multi-objective optimization of simultaneous balancing and sequencing problem of mixed model assembly line.

In the literature, different approaches have been used to investigate a balancing and sequencing problem of assembly line. For example, Sawik [10–12], Wu et al. [21] and Öztürk et al. [18] introduced exact methods for simultaneous balancing and sequencing problems in mixed model assembly lines. These methods can give the exact solution but may take larger computational time. Therefore, some researchers, for example Kim et al. [8, 9, 13], Miltenburg [22] and Kara [23], proposed meta-heuristic approaches for simultaneous balancing and sequencing problems in mixed model assembly lines. In recent years, different types of multi-objective algorithms and their extensions for multiple-objective optimization have been studied in the literature [24–27]. For example, Guo et al. [28, 29] proposed a multi-objective optimization model by combining a famous multi-objective optimization in the literature, i.e. non-dominated sorting genetic algorithm (NSGA) II [30], with a simulation tool to solve production planning problem. Guo et al. [28] proposed a method composed of two steps to optimize their problem: in the first step, NSGA II was used to generate candidate solutions of their problem, and in the second step, a process simulator was used to determine the performance of the candidate solution. Guo et al. [29] further developed another method to optimize a multi-objective problem of the order planning. In this method, three steps are involved to optimize the solution in which multi-objective mimic optimization is used as the first step, and then, Monte Carlo simulation sub-model is used as the second step, and in the last step, a heuristic sub-model is used to optimize their considered problem. However, the involvement of many steps in their methods including simulation may take more computation time if the candidate solutions are large enough. Other methods that have been used in the literature for multi-objective problem optimization includes ant colony-based algorithm [31] and certain types of swarm-based multi-objective optimization algorithms [32, 33]. Recently, Karaboga [34] proposed an algorithm called artificial bee colony (ABC) algorithm which is based on the foraging behaviour of honey bee swarm. It is an efficient algorithm due to its fruitful characteristics such as compared to other optimization algorithms; for example, ABC algorithm needs less control parameters, and it is simple and easy to implement. Artificial bee colony algorithm is a well-known swarm-based optimization algorithm which can be used for combinatorial optimization problems [35]. Karoboga and Gorkemli [35] proposed a combinatorial artificial bee colony algorithm to solve travelling salesman problem and showed that ABC algorithm can be effectively applied to combinatorial optimization problems. Moreover, ABC algorithm has been used for multi-objective optimization problems [36, 37]. Omkar et al. [36] presented a vector evaluated artificial bee colony (VEABC) algorithm for multi-objective design optimization of composites. They evaluated the performance of their proposed VEABC algorithm with the existing naturally inspired algorithms, i.e. artificial immune system (AIS) and genetic algorithm (GA), and got quite satisfactory results of their problem from VEABC algorithm. Akbari et al. [37] presented artificial bee colony algorithm for multi-objective optimization. They used a grid-based approach to access the Pareto frontier and maintained Pareto solutions in an external archive. The trajectories of the employee bee in their algorithm are based on the non-dominated solutions stored in the archive in their proposed algorithm. Onlooker bees select the food sources and update the archive in their proposed algorithm. In recent works, Tapkan et al. [38] studied a two-sided assembly line balancing problem which is aimed to balance the line and to minimize the number of stations. However, they combined their two objectives into a single objective which may not give Pareto solutions. Pan et al. [39] presented a discrete version of artificial bee colony algorithm for a lot-streaming flow shop scheduling problem to optimize the objective of earliness and tardiness. However, they gave weightage to each objective and combine these objectives into one objective which might not be suitable to get Pareto solutions. Li et al. [40] introduced a hybrid Pareto-based discrete version of artificial bee colony (P-DABC) algorithm for multi-objective optimization of the discrete nature of problems. They proposed local searches and crossover operation in standard ABC algorithm to design P-DABC. Their proposed method includes local searches in different stages to search optimal solutions. Moreover, they investigated flexible job shop scheduling problem for optimization. However, their studied problem is different from the current problem of simultaneous balancing and sequencing of assembly line. Zhang et al. [41] proposed a hybrid version of ABC algorithm for job shop scheduling problem. They proposed a novel ABC algorithm and introduced a tree-based local search mechanism in the onlooker bee phase to increase the local search ability of their studied problem. However, their proposed algorithm is specific for the problem they studied and they considered single objective optimization. Wang et al. [42] proposed an effective ABC algorithm for flexible job shop scheduling problem. They considered local search based on the critical path to enhance the local search ability of the onlooker bees. However, they used to optimize single-objective function and their problem is different from the problem studied in the current research. Later, Wang et al. [43] proposed an enhanced Pareto-based artificial bee colony algorithm for flexible job shop scheduling problem. They used exploitation search procedure for both employee bee and onlooker bee to generate new neighbour food sources. Moreover, they used crossover operator for onlooker bee to exchange information between the bees. However, their proposed algorithm most suits for the problem they studied which is different from the current problem of sequencing and balancing of assembly line. Tasgetiren et al. [44] proposed a discrete ABC algorithm to solve permutation flow shop problem. They introduced different neighbourhood structures to generate neighbours of food sources and assigned one kind of neighbour structure to a food source to make its neighbours. However, their proposed algorithm is designed for single objective optimization. Kalayci and Gupta [45] proposed an artificial bee colony algorithm to solve a sequence-dependent disassembly line balancing problem. They developed a solution representation of their studied problem and prepared food sources using a neighbourhood method for each employee bee. Onlooker bee selects the food sources given by employee bee using their selection probability. They optimized four different objectives of assembly line balancing problem independently and compared them with the results of other famous algorithms on the basis of the independent values of these objectives. They compared their results with famous algorithms in the literature including ant colony (ACO) algorithm, GA, particle swarm optimization (PSO), river formation dynamics (RFD), simulated algorithm (SA) and tabu search (TS) algorithms and found that ABC algorithm performs well than the other algorithms. However, multiple-objective optimization based on Pareto concepts has not been used in their research. Their results indicate that ABC algorithm is a better choice to use for the combinatorial problems in the recent research. Moreover, from the literature, it can be seen that few studies have considered multi-objective optimization using Pareto concepts in ABC algorithm and the problem they focused are different from the currently studied problem of simultaneous sequencing and balancing of mixed model assembly line. This motivates to introduce multi-objective optimization ABC algorithm which can include Pareto concepts in it for the current problem.

To the best of the author's knowledge, the current study is novel to introduce multi-objective ABC (multi-ABC) algorithm for simultaneous sequencing and balancing of mixed model assembly line to get Pareto solutions. The proposed multi-ABC algorithm shows a new solution representation method which can be effectively applied to do simultaneous balancing and sequencing of mixed model assembly line. Furthermore, local search mechanism is incorporated in the proposed multi-ABC algorithm which can be significantly used to search different sequencing solutions corresponding to different balanced solutions of mixed model assembly line for multi-objective optimization. Rest of the paper is organized as follows: Section 2 describes mixed model assembly line balancing and sequencing problem, Section 3 illustrates proposed multi-ABC algorithm, Section 4 indicates computational experiments and results, and at the last, Section 5 shows the conclusion and future research directions.

2 Mixed model assembly line sequencing and balancing problem

In the current research, a mixed model assembly line is considered and the following notations and assumptions are used:

Notations

i	is the index used to represent tasks.
n	indicates the number of tasks.
m and w	are the indexes used to indicate a model.
M	represents the number of different varieties of models produced in a straight assembly line.
$D=\{D_1,D_2,\ldots,D_M\}$	indicates the demand of each model.
$d=\{d_1,d_2,\ldots,d_M\}$	indicates the demand of models in each assembly cycle called as minimum part set (MPS), and this method is very common in the literature to compute the demand of a model in one assembly cycle.
$d_m=D_m/h$	is the demand of model m in one assembly cycle.
h	is the greatest common divisor from the demand of models.
j and y	are the indexes used to show a station.
S	represents the number of stations in an assembly line.
t_{im}	is the mean processing time of task i of model m.
T_{mj}	is the mean time required to perform tasks of model m on station j.
a_m	is the production share of model m, i.e. the ratio of the demand of model m to the overall demand.
AL_j	is the average load of each station j.
n_j	represents the number of models whose tasks are assigned to station j.
t_{mj}^{sum}	is the sum of task times of model m in station j.
x	is the position of model m in a mixed model sequence.
X_{ijm}	is the binary variable which is equal to 1 if the task i of model m is assigned to station j, and otherwise, its value is equal to 0.

Assumption

- Demand of each model is assumed to be known in advance, and it remains constant in an assembly cycle.
- All models are required to process on a straight assembly line.
- Conveyer belt moves at constant speed.
- The number of stations in assembly line, the number of different models and precedence relation of different tasks for each model of product and task times are known.
- Similar and dissimilar tasks of different models can have different task times.
- Each task is required to assign only on one station.
- Unlimited buffer space is assumed between stations.
- Set-up time is included in the task times.
- Travel time of parts is taken as 0.

In the current problem, similar tasks of different models can have different task times. Further, each task of model also can have different task times. Therefore, when some of these tasks are assigned to a certain station, there is a possibility that incomplete units might be produced in the given cycle time of the line. It is therefore considered to balance workload on each station for different models as well as to balance the workload in different stations. To obtain these two objectives of balancing problem, Merengo et al. [19] introduced the concepts of horizontal and vertical balancing. They presented functions for horizontal and vertical balancing and considered one objective at a time in their proposed methodology to be optimized, i.e. either they used horizontal balancing or used vertical balancing objective to be optimized in their proposed algorithm of balancing problem. However, in real environment, both horizontal balancing and vertical balancing are needed at the same time, and therefore in the current study, the horizontal and vertical balancing objectives proposed by Merengo et al. [19] are considered to be optimized simultaneously. The horizontal and vertical balancing objectives proposed by Merengo et al. [19] are considered here as objectives of balancing problem in the current research. The objectives used for horizontal and vertical balancing of proposed mixed model assembly line are indicated in Eqs. (1) and (2), respectively.

$$Z_{\mathrm{Hbalancing}} = \mathrm{Min}\sum_{j=1}^{S}\left[\frac{\sum_{m=1}^{M}\alpha_m\left(\max_{w=1}^{M}\{T_{wj}\}-T_{mj}\right)}{\max_{m=1}^{M}\{T_{mj}\}}\right] \tag{1}$$

$$Z_{\mathrm{Vbalancing}} = \mathrm{Min}\sqrt{\sum_{j=1}^{S}\left(\mathrm{AL}_j-\sum_{y=1}^{S}\left(\frac{\mathrm{AL}_y}{S}\right)\right)^2} \tag{2}$$

where AL_j can be computed using the relation shown in Eq. (3).

$$\mathrm{AL}_j = \sum_{m=1}^{M} T_{mj}\times\alpha_m. \tag{3}$$

The objective of horizontal balancing shown in Eq. (1) is aimed to balance workload for different models in each station. The objective of vertical balancing shown in Eq. (2) is aimed to reduce the deviation of workload on a station from the average of workload of all the stations. These two objectives are significant to balance the assembly line, and they are simultaneously considered to be optimized in the current research. Furthermore, model sequencing in assembly line also contributes to reduce the possibility of generation of incomplete units in the assembly line and is therefore presented here to be optimized along with balancing objectives. Merengo et al. [19] proposed a sequencing objective which they used to optimize with a separate algorithm from the algorithm they developed for the balancing objectives. They considered optimizing their proposed sequencing objective separately from their presented balancing objectives. In the current research, a new objective of model sequencing is proposed which is considered to be optimized simultaneously along with the two objectives of balancing. It is assumed that the tasks assigned to a station belonging from different models, when assigned to process on a station, can be related as a single machine sequencing problem in which the sum of task times of a model can be considered as the time of one job and the number of models to process on a station can be treated as the number of different jobs on a single machine problem. With this assumption, different sequences of models on a station can give different amounts of waiting time of models on that station. This can affect the flow time of different models on a station and may contribute to the generation of incomplete units. In the current problem, model sequence on stations is considered as the same for all stations. Furthermore, model sequencing objective on stations is considered as minimization of the total flow time of models on stations. The proposed model sequencing objective introduced for the sequencing problem of mixed model assembly line is indicated in Eq. (4).

$$Z_{\mathrm{Sequencing}} = \mathrm{Min}\sum_{j=1}^{S}\sum_{x=1}^{n_j}\left(n_j+1-x\right)\times t_{mj}^{\mathrm{sum}}. \tag{4}$$

The presented sequencing objective in the current research can be significant in reducing the flow time of different models on stations and can contribute to further in reducing the possibility of generation of incomplete units in assembly cycle. Three objectives are considered simultaneously to be optimized here which includes horizontal balancing, vertical balancing and sequencing objective.

The constraints considered for the balancing and mixed model sequencing of assembly line are presented in Eqs. (5) to (9).

$$\sum_{j=1}^{S} X_{ijm}\leq 1\ \forall\{i=1,2,3,\ldots,n\},\ \forall\{m=1,2,3,\ldots,M\} \tag{5}$$

$$T_{mj}\leq CT\ \forall\{j=1,2,3,\ldots,S\},\ \forall\{m=1,2,3,\ldots,M\} \tag{6}$$

$$T_{mj}\times\alpha_m\leq CT\ \forall\{j=1,2,3,\ldots,S\},\ \forall\{m=1,2,3,\ldots,M\} \tag{7}$$

$$\sum_{j=1}^{S} j \times X_{ijm} \leq \sum_{j=1}^{S} j \times X_{kjm} \ \forall (i,k) \in P, \ \forall \{m = 1, 2, 3, \ldots, M\} \quad (8)$$

$$X_{ijm} \in \{0, 1\}. \quad (9)$$

The constraint shown in Eq. (5) indicates that each task of a model is assigned to only one station in the assembly line. The constraint shown in Eq. (6) shows that the mean time required to perform tasks of model m on station j should not be more than the cycle time. Moreover, the constraint presented in Eq. (7) indicates that the weighted mean time of a model on a station cannot exceed the cycle time of an assembly line. The constraint shown in Eq. (8) guarantees the precedence relation of the models, and in Eq. (8), P indicates the subset of the tasks (i,k), given the direct precedence relation. The constraint shown in Eq. (9) indicates a binary variable which is equal to 1 if the task i of model m is assigned to station j, and otherwise, its value is equal to 0.

3 Multi-objective artificial bee colony algorithm

Artificial bee colony algorithm introduced by Karaboga [34] is an efficient algorithm which has main advantages that it needs less control parameters and is easy to implement. It is based on the foraging behaviour of honey bee swarm, and it uses three kinds of bees called, employee bee, onlooker bee and scout bee to solve the optimization problem. In ABC algorithm, the food source represents the solution of an optimization problem, and the nectar amount indicates the fitness or value of an optimizing objective of the problem. In ABC algorithm, employee bees are responsible to visit the food sources and taste these food sources to get their nectar amount. Onlooker bees wait for the employee bees in the hive. Employee bees after visiting food sources can share the information of a nectar amount of food sources with onlooker bees. Onlooker bees select or reject the food sources on the basis of their nectar amount. If there is no improvement in the nectar amount of a food source by an employee bee for some known number of cycles (called limit cycles), the corresponding employee bee becomes a scout bee that searches out a new food source randomly. The original structure of ABC is significant for optimization of continuous problems, but recently, Karoboga and Gorkemli [35] introduced ABC for combinatorial optimization problems, called CABC. In this algorithm, they introduced new food sources in the vicinity of employee bee and these food sources are compared on the basis of the probability of selection by using greedy selection process. The more the nectar amount of food source, the more is its probability of selection. The onlooker bee in their proposed algorithm also used a greedy method to select a food source and store it for the next cycle of the algorithm. However, in every step, using a greedy method may give a possibility to give local optimal results because some kind of diversity mechanism is important to consider in the algorithm. A little effort has been paid to develop a discrete version of ABC algorithm in the literature [40]. Li et al. [40] used local search methods in different stages of their proposed algorithm to solve a job shop problem which is different from the current problem. They used local search and perform non-dominated sorting in different stages of their proposed ABC algorithm. This may increase the computation. Furthermore, the problem they analysed is different from the current problem.

In the current research, mixed model sequencing and assembly line balancing objectives are desired concurrently and, therefore, the algorithm is designed to consider the solutions which can optimize the desired objectives. Therefore, the current research is focused to develop artificial bee colony for multi-objective optimization of simultaneous sequencing and balancing problem. Current problem has different solution requirements as compared to a simple scheduling problem because it not only includes the sequence of mixed models on the line but the solution also includes assembly line balancing solution simultaneously. So, a new food source representation for simultaneously studying of sequencing and balancing problem is presented here. Furthermore, due to multi-objective optimization problem, the solution may not have a single solution but may have a set of solutions called Pareto set, and therefore, Pareto concepts are also incorporated in the proposed multi-ABC algorithm for the search of the best solution from a set of Pareto solutions. Moreover, local search technique is introduced to find different model sequences in different assembly line balancing solutions to find Pareto solutions. The stepwise procedure of the proposed multi-ABC algorithm is presented in this section.

3.1 Food source representation

In the proposed multi-ABC algorithm, two kinds of food sources are introduced, i.e. food sources A and B. The first type, food source A, represents tasks and their respective stations assigned to them and also can answer if which tasks are needed to produce which product model. Food source A is indicated by a matrix as shown in Fig. 1. In the example shown in Fig. 1, eight tasks, three stations and two models are considered.

The numbers in the first row of food source A representation indicates tasks, and the second row shows stations assigned to each task. It can be seen from Fig. 1 that the number 2 under tasks 1 and 7 columns indicates that they will be processed in station 2. The third and fourth rows indicate the identification of tasks which are needed for models 1 and 2, respectively. For example, task 1 is required by both models of the product, so there is a black dot mark in both the third

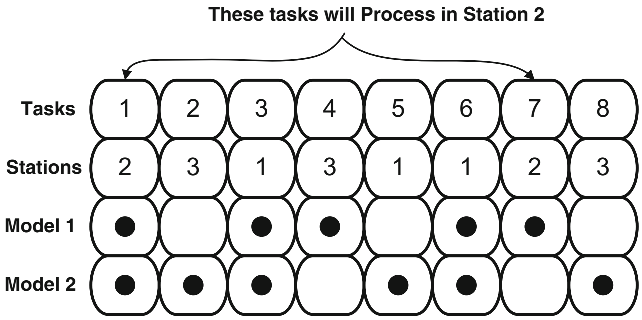


Fig. 1 Food source A representing tasks, stations and the models

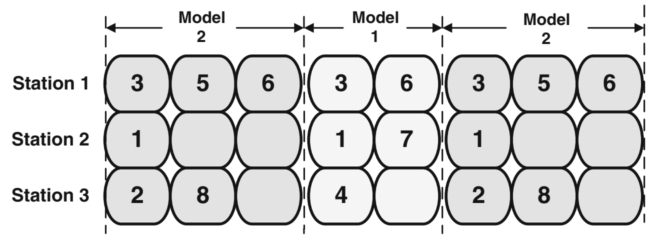


Fig. 3 A food representing information from types A and B food sources

and fourth rows under task 1 column. Similarly, task 4 is required by only model 1 so the dot mark is shown only in the third row under the column of task 4 and there is an empty space in the fourth row under the column of task 4. This empty space indicates that there is no requirement of task 4 in the preparation of model 2.

The model sequencing in an assembly line problem is obtained by first finding the model quantity in a cycle, i.e. $d=\{d_1,d_2,\ldots,d_M\}$. Suppose the quantity of model 2 product is double than the quantity of model 1 product. Then, the model sequence is made between two model 2 products and one model 1 product. The sequence between models 1 and 2 is generated (i.e. randomly or by some neighbourhood structure) and is considered as food source B. The generated food source B as an example is shown in Fig. 2.

The food source includes the information of both food sources of types A and B. The food source containing information from both types A and B is shown in Fig. 3. It can be seen from Fig. 3 that it represents the sequence of models and their tasks assigned in all stations. In this representation, the number of rows indicates the number of stations and the number of columns is equal to the maximum number of tasks on any station for processing all models in the required quantity. In Fig. 3, the first row indicates the first station, the second row shows the second station, and the third row shows the third station. The numbers in these rows describe the tasks which have to be processed in different stations (as obtained from a food source of type A) and are arranged in an order of model sequence which is already obtained, i.e. 2, 1, 2 (as obtained from a type B food source). For example, in Fig. 3, the tasks of model 2 which have to be processed in station 1 (i.e. tasks 3, 5 and 6) are written first in the first row, then all the tasks of model 1 which are needed to be processed in station 1 (i.e. tasks 3 and 6) are described in the first row, and then, all the tasks of model 2 which need to be processed in

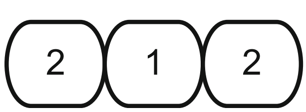


Fig. 2 A food source B indicating a model sequence

station 1 are written, respectively. Similarly, tasks are mentioned in their respective stations, keeping the model sequence identical (i.e. 2, 1, 2) in all stations. In this representation, model sequence can be changed to generate new food sources. In this representation, a large variety of food sources can be formed by varying model sequence and balancing solutions. Food source A is used to describe a balancing solution of the assembly line, while food source B is used to produce sequencing solutions of the balanced solutions.

In the proposed algorithm, the neighbour food sources are generated by three different neighbourhood generation mechanisms. First, the demand of model for an assembly cycle is obtained and then a model sequence is generated, i.e. a food source of type B (similar to the model sequence of the supposed example as shown in Fig. 2). After generation of the model sequence for an assembly cycle, the proposed mechanisms are used to generate the neighbourhood food sources of these solutions. The mechanisms used to make neighbourhood food sources are indicated below:

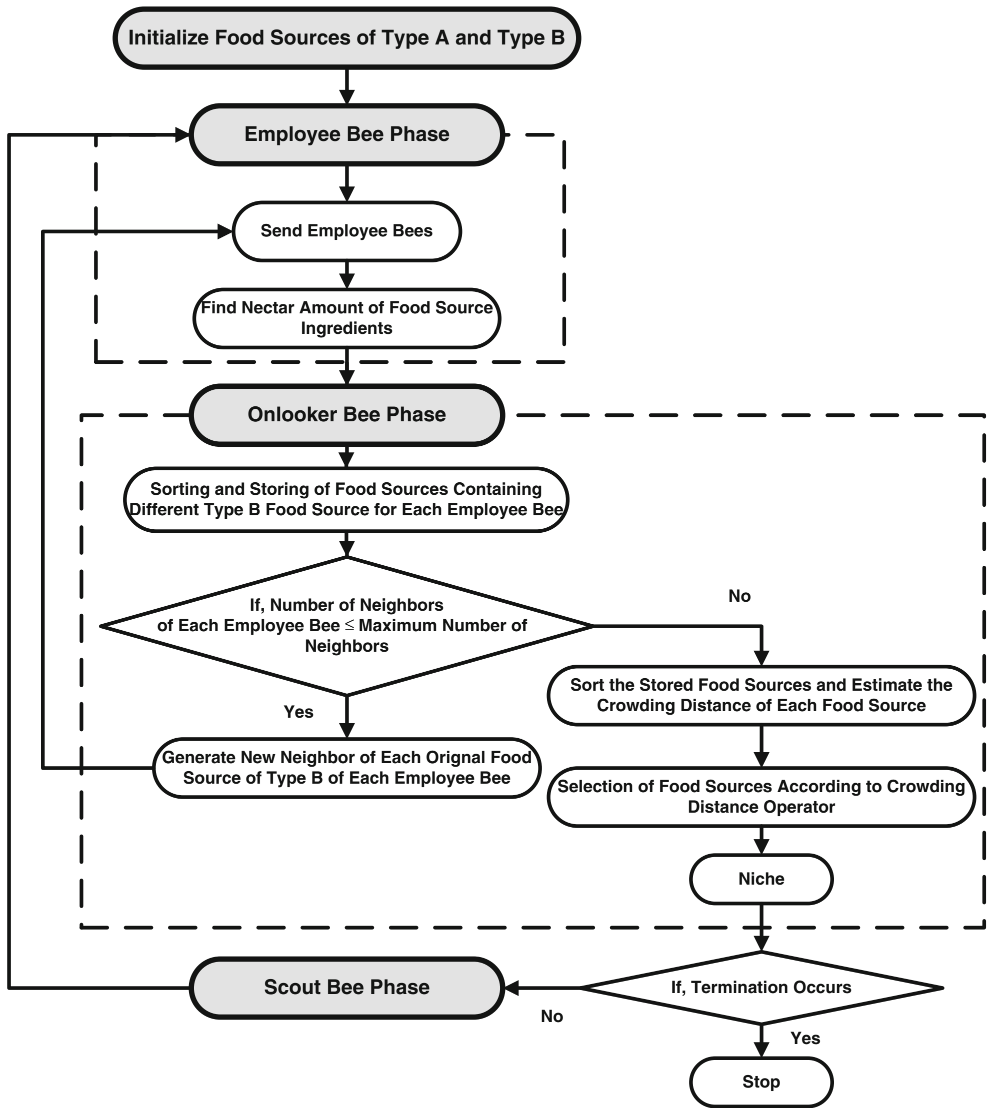
- N_1 move: In this mechanism, two models at two different positions in food source B are randomly selected and are exchanged to make a new model sequence.
- N_2 move: In this mechanism, a randomly selected model is removed from the sequence and is inserted to some new location in food source B.
- N_3 move: In this mechanism, two models at different locations in a model sequence are randomly selected and both are inserted to some new positions in food source B.

The flow chart of the proposed algorithm is shown in Fig. 4, and the stepwise procedure of the proposed multi-ABC algorithm is presented in this section.

3.2 Food source initialization and employee bee phase

In the proposed multi-ABC algorithm, the initial population of food source A is randomly generated which is equal to the population size of employee bees. Each employee bee is also assigned a food source of type B. In the current problem, both food sources A and B are needed to obtain the proposed multiple-objective solution. Once, both food sources A and

Fig. 4 Flow chart of the proposed multi-ABC algorithm



B for each employee bee are defined; the nectar amount of food sources is obtained. The proposed problem is designed to optimize more than one objective, so the nectar amount of food sources is not dependent on one function. The number of food source ingredients is considered as the number of objectives here. The nectar amount of food source ingredient corresponding to each objective for a food source r is indicated in Eqs. (10), (11) and (12), respectively. After getting the nectar amounts of food source ingredients, the information is sent to the onlooker bees.

$$\mathrm{NEC}\left(Z_{\mathrm{Hbalancing}}\right)_r = \sum_{j=1}^{S}\left[\frac{\sum_{m=1}^{M}\alpha_m\left(\max_{w=1}^{M}\{T_{wj}\}-T_{mj}\right)}{\max_{m=1}^{M}\{T_{mj}\}}\right]_r \tag{10}$$

$$\mathrm{NEC}\left(Z_{\mathrm{Vbalancing}}\right)_r = \left[\sqrt{\sum_{j=1}^{S}\left(\mathrm{AL}_j-\sum_{y=1}^{S}\left(\frac{\mathrm{AL}_y}{S}\right)\right)^2}\right]_r \tag{11}$$

$$\mathrm{NEC}\left(Z_{\mathrm{Sequencing}}\right)_r = \left[\sum_{j=1}^{S}\sum_{x=1}^{n_j}\left(n_j+1-x\right)\times t_{mj}^{\mathrm{sum}}\right]_r. \tag{12}$$

3.3 Onlooker bee phase

Onlooker bee phase composed of some steps which are explained in the following section.

3.3.1 Sorting and storing of food source containing different type B food sources

In the first cycle of the algorithm, all the food sources provided by the employee bee are stored in the onlooker bee phase. In the proposed algorithm, each employee bee is not restricted only to visit one kind of a food source of type B. However, each employee bee is required to visit a population of food source of type B using neighbourhood structures presented in the previous section. Furthermore, the food source of type A of each employee bee remains the same in 1 cycle of algorithm and only neighbour food sources of type B are modified for each employee bee. This can increase the local search because each neighbourhood food source of an employee bee has the same type A and different type B food sources. So, for each employee bee, each assembly line balance solution is presented in type A food sources of employee bees, while the variety of mixed model sequences are formed in its neighbours containing different type B food sources. This can increase the number of different possible mixed model sequences to be observed for each assembly line balance solution.

For the first cycle of algorithm, the initial food sources of type B are considered as the best type B food source for each employee bee and are stored. After storing the food sources in the onlooker bee phase, the neighbourhood food source of type B of each employee bee is created and each employee bee is again sent to employee bee phase. The structure of neighbourhood generation of type B food source is randomly selected from N_1, N_2 and N_3 moves to generate every neighbour of type B. Each employee bee visits their corresponding neighbour and computes the nectar amount of food source ingredients. This information is sent to the onlooker bees, and the solution quality of a food source of each employee bee which contains a newly generated neighbour food source of type B is compared with the solution quality of the stored solution of the corresponding employee bee. This comparison of type B food source neighbours is performed separately for each employee bee and between the food source containing a newly generated neighbourhood food source of type B and the stored food source containing the original structure of type B food source. This comparison is performed by determining the dominance relation between them. A food source r dominates another food source s, i.e. $r \prec s$, if food source r is better than food source s in all of its food source ingredients. Further, r is strictly better than s in at least one of the food source ingredients. The food source of each employee bee which dominates, called the best food source, is stored and if there is no dominance relation between them, then any one of the solutions is stored and a new neighbour food source of type B is created for the employee bees. The procedure continues unless the predefined number of neighbours of type B food source is observed. The proposed procedure can increase the local search mechanism of the algorithm and helps to search different neighbourhood food sources of type B to find out the best model sequences for each variety of balancing solutions. In the proposed method, different types of food source of type A (equal to the number of employee bees) are significant to generate different balancing solutions of the assembly line balancing problem, while neighbourhood food sources of type B for each employee bee are significant to investigate a variety of model sequencing solutions for each assembly line balanced solution.

3.3.2 Sorting and crowding distance estimation

Once the maximum number of neighbours of type B for each employee bee is observed, the nectar amount of food source ingredients of all employee bees is ranked according to the dominance relation between them. A food source r dominates another food source s, i.e. $r \prec s$, if food source r is better than food source s in all of its food source ingredients. Further, r is strictly better than s in at least one of the food source ingredients. Food sources after getting their dominance relations are ranked, and their ranks are assessed using the ranking method proposed by Deb et al. [30]. In this method, non-dominated food sources are identified and defined as belonging to grade 1 food sources. The food sources that are graded are deleted, and in the next step, the non-dominated food sources from the remaining food sources are determined and defined as grade 2 food sources. The food sources that are graded are deleted after each step, and the grade value is increased by 1 for non-dominated food sources until the entire food sources have grade values. After getting the grades, all food sources are sorted according to their grades and crowding distance of each food source is obtained [30].

3.3.3 Selection of food sources

The current problem contains more than one objective, and in this case, distance-based measure is used to improve the crowding between the food sources. There are different distance-based measures used in the literature, e.g. Euclidean distance and Mahalanobis distance measure [46]. The Euclidean distance has straightforward geometric interpretation, computationally inexpensive and simple to code as compared to Mahalanobis distance [45]. The estimation of a crowd of the point near some points is important to estimate the density of the solution which is surrounded by other solutions. Deb et al. [30] computed the crowding distance by taking the average distance of the two points on either side of the examining point along each objective. In their method, crowding distance between two solutions is obtained by first sorting the population of solution in ascending order according to the magnitude of each objective function. In each objective, the order of solutions in sorting a list of solution might be different. For each objective function, the boundary solutions (solutions containing the smallest and largest objective function values) are assigned with infinite values of their distances. The remaining solutions are assigned with distance values equal to the absolute normalizing difference of their objective values. This is performed for all objective functions, and later, the overall crowding distance can be calculated by adding the distances corresponding to each objective function of the solutions. Mahalanobis distance can provide the relative measure of the distance of points from a common point and needs the information of the distribution of the points which might be hard to know in the current case of multi-objective optimization. Furthermore, Mahalanobis distance is computationally expensive to compute and, therefore, crowding distance comparison operator [30] is used here for the selection and rejection of the food sources and for sorting of food sources according to their ranks and crowding distances. The crowding distance of the solutions is calculated using the following steps:

Step 1: In this step, the food sources are ranked and non-dominated fronts are identified, i.e., $F_1, F_2, \ldots, F_R$, and the later steps are repeated.

$$\mathrm{CD}_k S_{i,k} = \frac{Z_k\left[S_{i+1,k}\right] - Z_k\left[S_{i-1,k}\right]}{Z_k^{\max} - Z_k^{\min}} \tag{13}$$

where $Z_k[S_{i,k}]$ shows the value of an objective function of objective k for food source $S_{i,k}$ ranked in position i.

Step 2: In this step, the food sources in each front F_{rank} are sorted with respect to each objective O. Suppose $Q = F_{\mathrm{rank}}$ and $S_{i,k}$ indicates the ith food source in the sorted list with respect to the objective function k. Then, the crowding distance is assumed to be $\mathrm{CD}_k S_{1,k} = \infty, \mathrm{CD}_k S_{Q,k} = \infty$, and for $1 < i \leq Q-1$, the crowding distance can be obtained from the relation shown in Eq. (9).

Step 3: The total crowding distance $\mathrm{CD_S}$ of a food source S is obtained by adding the crowding distance of food source with respect to each objective, i.e.

$$\mathrm{CD}_S = \sum_{k=1}^{O} \mathrm{CD}_k S.$$

In crowding comparison method, two attributes computed in the employee bee phase, i.e. ranks of food source P_{rank}, and the crowding distance of food sources P_{distance} are used for sorting according to the crowded comparison operator.

$$p \prec_n q \quad \text{if} \quad , \quad (p_{\mathrm{rank}} < q_{\mathrm{rank}})$$
$$\text{Or}, \ [(p_{\mathrm{rank}} = q_{\mathrm{rank}}) \ \text{and} \ (p_{\mathrm{distance}} > q_{\mathrm{distance}}).$$

These relations indicate that from the two food sources p and q of different non-dominated ranks, the food source with a lower rank is preferred. If both food sources belong to the same front (i.e. containing the same rank), then the food source which is located in a less crowded region is preferred.

Niche Niche step is significant so that solution may not trap in the local optima. In this method, some percentage of selected solutions are modified using a swap mutation operator and are moved to the next cycle of the algorithm.

3.4 Scouts bee phase

New food sources of type A and type B are randomly generated in scout bee phase and also, the selected food sources are sent in the next cycle of algorithm. This can increase the diversity mechanism and also stores the best food sources into the next cycles of algorithm.

4 Computational experiments and results

In this section, the performance of proposed multi-ABC algorithm is analysed to solve mixed model assembly line problems. Two kinds of mixed model assembly line problems are considered for this analysis. For first type of mixed model assembly line problem, task time data and precedence relation data of each model are taken from standard assembly line balancing problems taken from operation research library (ORL) (i.e. Sawyer, Buxey and Heskia) given by Scholl [47, 48]. For the second mixed model assembly line problem, each model task time data and precedence relations are taken from a truck manufacturing company in China. Both problems of mixed model assembly line are solved from the proposed multi-ABC algorithm, and its performance is tested against a famous algorithm in the literature, i.e. NSGA II [30]. Both proposed multi-ABC algorithm and NSGA II are coded in Visual C++, and the same system is used for both algorithms for this analysis. The algorithm parameters used for the proposed multi-ABC algorithm and NSGA II are summarized in Table 1.

Table 1 Parameters used for multi-ABC and NSGA II algorithm

Parameter	Values
Multi-ABC algorithm	
Food source population	800
Number of employee bee	800
Number of onlooker bee	800
Limit cycle	10
Maximum number of neighbourhood food sources	10
Percentage of nich food sources sent to the next cycle	25
Maximum number of cycles	1,000
NSGA II algorithm	
Population size	800
Maximum number of generations	6,000
Crossover rate	0.6
Mutation rate	0.7

Selection is done using the tournament selection method

4.1 Mixed model assembly line based on benchmark problem data

In mixed model assembly line, three models X, Y and Z are considered to be processed in eight stations in the proposed simultaneous sequencing and balancing problem. Task time data and precedence constraints of model X, Y and Z are respectively considered as task time data and precedence constraints of the standard benchmark problems (i.e. Sawyer, Buxey and Heskia). Sawyer benchmark data consist of 30 tasks and 8 stations, and it is used to describe model X; Buxey benchmark data consist of 29 tasks and 8 stations, and it is used represent model Y, and Heskia benchmark data consist of 28 tasks and 8 stations, and it is used to define model Z.

4.1.1 Mixed model assembly line simultaneous sequencing and balancing results

Mixed model assembly line sequencing and balancing problem is analysed with two different demand scenarios of models in this section. The current multi-objective problem has a Pareto solution which is a set of solutions from which only two results are described in detail in Table 2 and one result is shown in Table 3 for two different demand scenarios, respectively. Table 2 indicates the sequencing results of different models in assembly line and the tasks assigned to each station of each model for the first demand scenario of models for both multi-ABC algorithm and NSGA II. It can be seen from Table 2 that the demand of model X is 2, the demand of model Y is 1, and the demand of model Z is 2 in one assembly

Table 2 Sequencing and balancing results for the first scenario of demand of models

	Multi-ABC	NSGA II
Number of models	3	3
Demand (X, Y, Z)	2, 1, 2	2, 1, 2
Model sequence	X, X, Y, Z, Z	Z, X, Z, X, Y
Objective values	(2.50028, 22.6172, 6,960)	(2.77641, 15.9916, 8,360)
Tasks of models assigned on all stations		
X	(10, 1, 2, 12, 13), (11, 3, 14), (15, 5, 17, 6, 4, 16), (20, 21), (22, 23, 7), (8, 9, 24, 25, 26), (27, 29), (28, 30, 18, 19)	(2, 10, 11, 1, 4), (5, 12, 3), (16, 17, 18, 19), (6, 7, 8, 9, 13), (14, 15, 20), (21, 24, 25), (26, 22, 23), (27, 28, 29, 30)
Y	(1, 7, 12, 9, 10), (3. 2, 6), (14, 15, 19, 21, 4, 5), (8, 11), (25, 16, 18), (22, 26, 27, 13, 17), (20, 23), (24 28 29)	(1, 7, 25, 3), (4, 5, 13, 2), (9, 26, 12, 10, 15, 27), (14, 19, 21, 6, 8), (16, 18, 22), (11, 17), (20, 23), (24, 28, 29)
Z	(1, 8), (4, 24, 26, 5, 19), (25, 3), (21, 27, 23, 9, 10, 22), (12, 2, 6, 11), (20, 13), (14, 16, 7, 15), (17, 18, 28)	(1, 26, 8), (9, 19, 20), (3, 21, 24, 25), (10, 12, 13), (2, 16, 23, 4, 11, 15), (22, 17), (27, 14, 6, 7), (18, 5, 28)
Model sequence	Y, X, X, Z, Z	Z, Y, X, Z, X
Objective values	(2.50068, 22.509, 6,960)	(2.81387, 16.4263, 7,660)
Tasks of models assigned on all stations		
X	(1, 2, 12, 13), (14, 11, 3), (10, 5, 6, 4, 16, 15), (20, 21), (22, 7, 8), (9, 24, 25, 26, 17), (23, 27), (29, 28, 30, 18, 19)	(10, 2, 3, 17, 18), (11, 16, 1, 5, 6), (4, 7, 19), (12, 13, 14, 20), (24, 25, 21), (15, 22, 8, 9, 26), (23, 27), (29, 30, 28)
Y	(1, 7, 12, 9, 10, 14), (3. 2, 6), (15, 19, 21, 4, 5), (8, 11), (25, 16, 18), (22, 26, 27, 13, 17), (20, 23), (24 28 29)	(1, 3, 4, 5), (2, 26, 6, 8), (11, 7, 12, 9), (10, 15, 19, 21, 14, 16), (18, 22, 25), (13, 17, 20), (23, 24), (28, 27, 29)
Z	(1, 8), (4, 24, 26, 27, 19), (25, 3), (21, 5, 23, 9, 10, 22), (12, 2, 6, 11), (20, 13), (14, 16, 7, 15), (17, 18, 28)	(2, 6), (17, 7, 18), (1, 21, 19), (3, 24, 25), (8, 9, 10), (26, 22, 27, 20, 12), (13, 14), (16, 11, 15, 23, 5, 4, 28)

Table 3 Sequencing and balancing results for the second scenario of demand of models

	Multi-ABC	NSGA II
Number of models	3	3
Demand (X, Y, Z)	3, 2, 1	3, 2, 1
Model sequence	Y, X, Y, X, Z, X	Y, X, Z, X, X, Y
Objective values	(3.44314, 5.13428, 8,204)	(3.44314, 5.13428, 9,604)
Tasks of models assigned on all stations		
X	(2, 11, 12), (13, 14, 10, 1, 4), (3, 16, 20), (5, 6, 7, 8, 9, 17, 24), (25, 26, 15), (21, 22), (23, 27), (18, 29, 19, 30, 28)	(2, 11, 12), (13, 14, 10, 1, 4), (3, 16, 20), (5, 6, 7, 8, 9, 17, 24), (25, 26, 15), (21, 22), (23, 27), (28, 29, 19, 30, 28)
Y	(1, 3, 7, 12), (4, 5, 9, 10, 15), (2, 6, 8), (13, 19, 21, 26, 27, 14, 16), (18, 22, 25), (11, 17), (20, 23), (24, 28, 29)	(1, 3, 7, 12), (4, 5, 9, 10, 15), (2, 6, 8), (13, 19, 21, 26, 27, 14, 16), (18, 22, 25), (11, 17), (20, 23), (24, 28, 29)
Z	(1, 2, 22), (8, 17), (21, 26, 23, 24, 9), (25, 3), (10, 12, 13), (27, 14, 6, 4, 5), (11, 19. 16, 7, 18), (20, 15, 28)	(1, 2, 22), (8, 17), (21, 26, 23, 24, 9), (25, 3), (10, 12, 13), (27, 14, 6, 4, 5), (11, 19, 16, 7, 18), (20, 15, 28)

cycle. The tasks of a model assigned to a station are shown in parentheses "()", the tasks assigned to the stations of a model are presented in parentheses, and the tasks assigned to different stations are separated with a comma ",". For example, it can be seen from the results of multi-ABC algorithm illustrated in Table 2 that the tasks of model X which are assigned to the first station are 10, 1, 2, 12 and 13 in the assembly line. While the tasks of model X which are assigned to the second station are 11, 3 and 14, the tasks of model X for the third station are 15, 5, 17, 6, 4 and 16 and so on, in the results of multi-ABC algorithm.

Table 3 indicates a solution from the set of Pareto solution and describes the sequencing results of different models in the assembly line and the tasks assigned to each station of each model for the second demand scenario of models for both multi-ABC algorithm and NSGA II. It can be seen from Table 3 that the demand of model X is 3, the demand of model Y is 2, and the demand of model Z is 1 in one assembly cycle. It can be seen from the results shown in Table 3 corresponding to multi-ABC algorithm that the tasks of model X which are assigned to the first station are 2, 11 and 12 in the assembly line. While the tasks of model X which are assigned to the second station are 13, 14, 10, 1 and 4, the tasks of model X for the third station are 3, 16 and 20 and so on. Furthermore, the model sequence obtained from multi-ABC algorithm is Y, X, Y, X, Z and X while the sequence of models is Y, X, Z, X, X and Y from NSGA II.

4.1.2 Comparison of results

Comparison of results based on inverted generational distance: The current problem has three different objectives, so the results of the current problem obtained from both multi-ABC algorithm and NSGA II are in the form of Pareto fronts. Therefore, inverted generational distance (GD) concept given by Coello and Cortes [49] is used to estimate the elements of distance of the Pareto solutions of multi-ABC algorithm and NSGA II from the true Pareto front to investigate the performances of the multi-ABC algorithm and NSGA II. The value of GD is computed from the relation indicated in Eq. (14).

$$\mathrm{GD} = \frac{\sqrt{\sum_{v=1}^{h} {d_v}^2}}{h}. \tag{14}$$

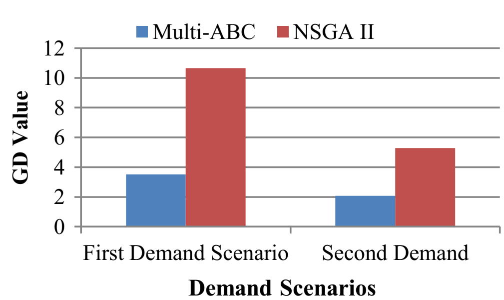


Fig. 5 GD values for different demand scenarios of models

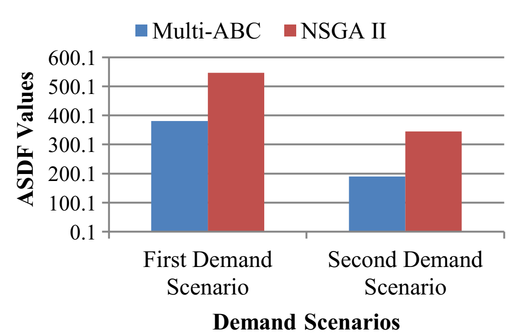


Fig. 6 ASDF values for different scenarios of demand of models

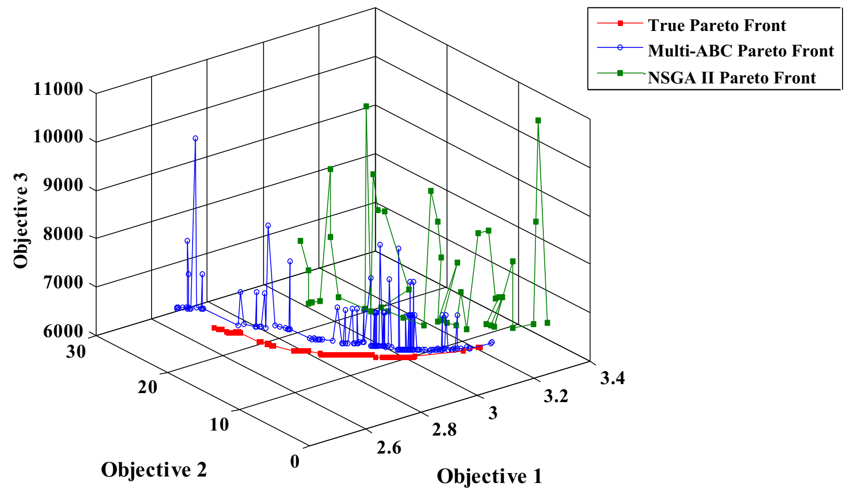


Fig. 7 Graphical representation and comparison of Pareto fronts for first demand scenario

The smaller value of GD indicates that the Pareto solution is closer towards true Pareto front. In order to compute the GD values for the proposed multi-ABC algorithm with NSGA II, true Pareto front is needed. The true Pareto front is difficult to obtain due to large search space. Therefore, in the current analysis, true Pareto front is assumed as a combined population of non-dominated solutions which are obtained from multi-ABC algorithm and NSGA II by running the proposed multi-ABC algorithm and NSGA II for a large number of cycles. This population of non-dominated solutions has contribution of Pareto solutions both from multi-ABC algorithm and NSGA II and can be assumed as true Pareto front. The non-dominated solutions obtained from multi-ABC algorithm and NSGA II at any other numbers of running cycles (i.e. less than the number of algorithm cycles which were used to get true Pareto front) are considered as the Pareto front of the respective algorithms. The GD value can be easily obtained from the Pareto fronts obtained from the respective algorithms and the true Pareto front for comparison of the Pareto front quality of the proposed multi-ABC algorithm and NSGA II.

The comparison of the multi-ABC algorithm and NSGA II on the basis of GD value for different demand scenarios of models is indicated in Fig. 5. It can be seen from Fig. 5 that GD values obtained from multi-ABC algorithm, for both demand scenarios of models, are significantly smaller as compared to those obtained from NSGA II. These results indicate that the Pareto solutions obtained from multi-ABC algorithm are more towards the true Pareto front and are more near to the optimal results.

The comparison of GD values is very common in the literature to investigate the performance of multi-objective results of different algorithms. It can compare the distance of a given set of Pareto optimal solutions from the true Pareto front. The comparison is only of this distance, i.e. GD value may not be sufficient to decide the optimality of multi-objective solutions [50]. This is because the optimization of multi-objectives depends on three different parameters and not only on GD value. These parameters include GD value, the distribution of Pareto solutions on the front and the number of solutions on the front [49]. Therefore, another comparison function is introduced here and is named as “average spacing

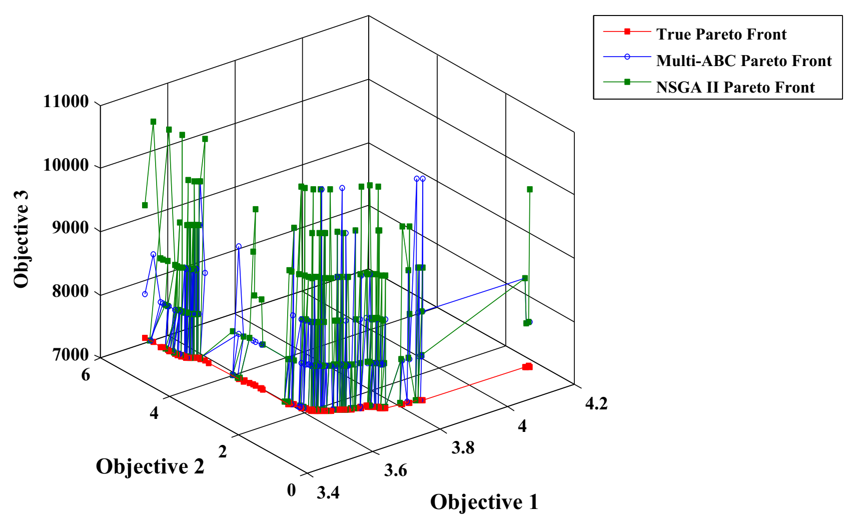


Fig. 8 Graphical representation and comparison of Pareto fronts for second demand scenario

deviation function (ASDF)" which can include the effect of distribution of Pareto solutions on the front and the number of solutions on the front.

Comparison of results based on average spacing deviation function: It is assumed that there are K numbers of Pareto solutions on the front. Then, ASDF can be computed from Eq. (15).

$$\mathrm{ASDF} = \sqrt{\frac{\sum_{i=1}^{K} (dav_i - dv)^2}{K}} \tag{15}$$

where

$$dv = \frac{\sum_{i=1}^{K} dav_i}{K} \tag{16}$$

$$dav_i = \frac{\sum_{j=1}^{K-1} d_{i,j}}{K-1} \quad \forall j = \{1, 2, \ldots, K\} \text{excluding solution } i \tag{17}$$

where $d_{i,j}$ represents the distance from Pareto solution i to Pareto solution j on the same front. The literature indicates that more number of solutions on the front are desired, and it is also desired to have uniformly distributed Pareto solution points on the front, i.e. the spacing between the solution points on the front is desired to be uniform [50]. It can be seen from Eq. (10) that these two goals can be achieved by minimizing ASDF. Smaller values of ASDF indicate that the Pareto front contains more evenly distributed solution points, and there are a large number of solution points on the front. The comparison results of proposed multi-ABC algorithm and NSGA II on the basis of ASDF values are indicated in Fig. 6.

It can be seen from Fig. 6 that the ASDF value is significantly smaller for the proposed multi-ABC algorithm as compared to NSGA II on different demand scenarios. These results indicate that the proposed multi-ABC algorithm gives the Pareto front which contains more number of solutions and are more evenly distributed as compared to the Pareto front obtained from NSGA II.

Comparison of results based on Pareto fronts generated: The graphical representation of Pareto fronts obtained from proposed multi-ABC algorithm and NSGA II for mixed model assembly line benchmark data is illustrated in Figs. 7 and 8 for the first and second demand scenarios of models, respectively. It can be seen from Figs. 7 and 8 that the Pareto front generated from multi-ABC algorithm is more towards true

Table 4 Precedence relation among tasks and task time data of each model from case company

Precedence relation of tasks for each model	(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 9), (1, 10), (1, 11), (1, 12), (1, 13), (1, 14), (1, 15), (1, 16), (1, 17), (1, 18), (1, 19), (1, 20), (1, 21), (1, 22), (2, 23), (23, 24), (3, 25), (24, 25), (25, 26), (26, 27), (27, 28), (26, 29), (26, 30), (26, 31), (26, 32), (26, 33), (26, 34), (26, 35), (5, 36), (26, 36), (26, 37), (5, 37), (17, 38), (26, 38), (11, 39), (26, 39), (18, 40), (26, 40), (26, 41), (26, 42), (26, 43), (2, 43), (2, 44), (26, 44), (26, 45), (26, 46), (46, 47), (46, 48), (46, 49), (47, 50), (46, 51), (46, 52), (48, 53), (49, 53), (50, 53), (51, 53), (52, 53), (26, 54), (53, 54), (47, 55), (53, 55), (55, 56), (47, 57), (53, 57), (47, 58), (26, 59), (26, 60), (26, 61), (48, 61), (53, 61), (61, 62), (26, 63), (26, 64), (53, 64), (22, 65), (26, 65), (53, 65), (64, 66), (65, 66), (66, 67), (26, 68), (48, 69), (49, 69), (63, 69), (48, 70), (49, 70), (63, 70), (59, 71), (64, 71), (69, 71), (70, 71), (71, 72), (72, 73), (73, 74), (72, 75), (72, 76), (72, 77), (77, 78), (78, 79), (73, 80), (26, 81), (78, 82), (78, 83), (83, 84), (72, 85), (85, 86), (86, 87), (87, 88), (86, 89), (80, 90), (90, 91), (72, 92), (91, 93), (92, 93), (16, 94), (93, 95), (95, 96), (72, 97), (72, 98), (98, 99), (98, 100), (72, 101), (80, 101), (72, 102), (80, 103), (72, 104), (80, 104), (72, 105), (105, 106), (105, 107), (105, 108), (108, 109), (108, 110), (105, 111), (105, 112), (105, 113), (112, 114), (114, 115), (114, 116), (114, 117), (111, 118), (112, 118), (113, 118), (72, 119), (28, 120), (72, 120), (105, 121), (105, 122), (105, 123), (78, 124), (105, 125), (66, 126), (124, 127), (127, 128), (127, 129), (127, 130), (127, 131), (82, 132), (83, 133), (127, 134), (107, 135), (127, 135), (127, 136), (127, 137), (127, 138), (127, 139), (127, 140), (94, 141), (127, 142), (127, 143), (95, 144), (127, 145), (136, 146), (138, 147), (100, 148), (38, 149), (148, 149), (149, 150)
Task time data for model	
X	5, 18, 4, 2, 3, 4, 3, 2, 1, 4, 1, 1, 2, 5, 26, 3, 3, 3, 1, 1, 3, 3, 9, 10, 5, 4, 12, 3, 3, 3, 3, 3, 6, 5, 5, 3, 3, 5, 3, 4, 2, 2, 3, 3, 5, 1, 1, 6, 3, 7, 4, 2, 3, 5, 5, 2, 3, 7, 2, 1, 13, 2, 2, 4, 2, 9, 6, 5, 2, 15, 13, 8, 10, 14, 3, 2, 4, 5, 10, 3, 9, 2, 12, 12, 5, 4, 4, 8, 10, 1, 4, 5, 5, 10, 2, 3, 18, 3, 3, 10, 6, 5, 4, 4, 4, 8, 2, 2, 4, 15, 2, 2, 2, 5, 6, 8, 5, 3, 4, 10, 10, 12, 2, 13, 5, 3, 2, 10, 1, 1, 6, 6, 4, 4, 3, 3, 8, 7, 14, 3, 9, 7, 5, 2, 2, 8, 14, 3, 3, 3, 14
Y	2, 12, 6, 1, 5, 3, 4, 2, 3, 2, 3, 4, 1, 3, 26, 4, 6, 6, 3, 3, 7, 2, 12, 5, 7, 2, 7, 5, 5, 5, 3, 3, 7, 20, 5, 3, 4, 7, 2, 6, 1, 1, 6, 6, 1, 3, 7, 8, 3, 5, 1, 22, 4, 10, 3, 2, 9, 6, 5, 20, 1, 5, 3, 6, 18, 3, 2, 1, 15, 23, 18, 11, 5, 3, 2, 8, 7, 12, 8, 11, 2, 25, 25, 5, 4, 6, 6, 10, 2, 6, 8, 8, 17, 3, 3, 22, 6, 2, 18, 4, 7, 7, 7, 7, 8, 3, 3, 7, 15, 4, 1, 1, 7, 6, 4, 4, 6, 7, 8, 18, 17, 14, 13, 15, 6, 12, 10, 15, 13, 4, 5, 7, 25, 36, 6, 25, 7, 24, 3, 8, 27, 4, 1, 1, 8, 16, 5, 5, 12, 20
Z	1, 22, 5, 5, 2, 6, 2, 12, 12, 12, 5, 5, 5, 7, 30, 2, 16, 16, 12, 14, 6, 7, 20, 6, 9, 3, 4, 8, 8, 8, 8, 4, 9, 17, 7, 6, 2, 7, 3, 9, 10, 11, 6, 2, 3, 3, 3, 5, 5, 3, 6, 12, 30, 12, 23, 4, 6, 7, 5, 20, 2, 26, 12, 5, 15, 7, 3, 1, 20, 11, 23, 4, 6, 6, 2, 10, 7, 12, 8, 11, 2, 25, 23, 33, 5, 2, 8, 12, 2, 6, 10, 10, 16, 5, 5, 20, 34, 3, 12, 5, 9, 12, 24, 7, 25, 12, 12, 9, 2, 4, 2, 12, 9, 4, 23, 4, 2, 9, 10, 12, 20, 19, 12, 22, 9, 10, 9, 12, 4, 4, 8, 12, 22, 15, 17, 17, 8, 24, 13, 9, 23, 6, 2, 5, 8, 17, 3, 2, 9, 17

Table 5 Sequencing and balancing results for the first scenario of demand of models

	Multi-ABC	NSGA II
Number of models	3	3
Demand (X, Y, Z)	2, 2, 1	2, 2, 1
Model sequence	X, X, Y, Y, Z	X, Z, Y, X, Y
Objective values	(2.34505, 2.89272, 14,710)	(2.56575, 2.43999, 16,431)
Tasks of models assigned on all stations		
X	(1, 2, 15, 9, 7, 17, 3), (23, 24, 25, 26, 46, 52, 49, 51, 48, 59, 8, 63), (70, 69, 60, 68, 29, 34, 47, 58, 50), (53, 57, 64, 71, 27, 72, 119, 105), (123, 121, 77, 78, 83, 84, 79, 124), (127, 44, 142, 98, 99, 76, 73, 80, 104), (101, 92, 111, 125, 122, 107, 113, 38, 32, 81, 42, 41, 33, 35, 20, 13, 11, 39, 10, 4, 12), (6, 43, 133, 31, 16, 94, 141, 135, 22, 75, 19, 30, 28, 45, 18, 40, 54, 100), (61, 62, 102, 148, 149, 97, 85, 86, 5, 37, 36, 129, 108, 110, 74, 112, 114, 117, 116, 118), (103, 150, 115, 109, 90, 91, 93, 95), (144, 96, 82, 132, 87, 88, 120), (65, 66, 67, 126, 55, 56, 14, 106, 89, 143, 128, 21, 134, 136, 131, 139), (146, 138, 147, 140, 145, 137, 130)	(1, 22, 3, 8, 15, 19, 2), (23, 24, 25, 26, 60, 45, 46, 47), (59, 52, 17, 38, 14, 41, 16, 58, 50, 48, 49, 51, 53, 54, 64, 6, 11, 42, 35), (63, 69, 70, 71, 57, 72, 85), (92, 77, 78, 82, 124, 127, 130, 83), (133, 84, 98, 100, 148, 29, 44, 81, 33, 34, 10, 7, 68, 20, 13, 21, 4, 5, 36), (37, 12, 9, 30, 39, 18, 40, 94, 141, 97, 31, 32, 128, 143, 79, 142, 132, 86, 89, 87), (88, 129, 76, 73, 138, 74, 80, 101), (90, 102, 75, 65, 66, 126, 67, 61, 62, 55, 56, 137, 134, 139, 131, 147), (145, 140, 43, 136, 146, 105, 125, 121), (113, 122, 106, 111, 107, 135, 123, 112, 118, 114, 115, 117, 116, 108), (109, 110, 119, 27, 28, 120, 99), (149, 150, 103, 91, 93, 95, 96, 144, 104)
Y	(1, 15, 8, 13, 2, 9, 16, 94, 21, 5, 141), (10, 17, 12, 4, 20, 6, 19, 11, 18, 23, 24, 7, 22, 3, 25, 26, 63, 39, 68, 32, 35, 42, 31, 45, 41, 36), (38, 27, 59, 28, 46, 49, 48, 51, 47, 69, 43, 44, 50, 60), (70, 40, 81, 30, 37, 58, 52, 53, 61, 62, 64), (71, 54, 65, 66, 67, 126, 55, 72, 92, 97), (14, 76, 77, 78, 124, 85, 102, 127, 128, 131, 140), (139, 130, 145, 143, 134, 129, 136), (146, 33, 82, 120, 137, 79, 73), (80, 101, 74, 56, 103, 104, 90, 29, 86, 87, 89, 91, 93, 95, 144), (96, 75, 57, 138, 147, 105, 112, 114, 117, 116, 115), (108, 109, 110, 121, 106, 122, 113, 98, 99, 100), (148, 149, 107, 135, 83, 133, 84, 123), (119, 125, 88, 142, 111, 150, 118, 132, 34)	(1, 6, 20, 9, 19, 7, 18, 4, 13, 5, 3, 17, 14, 22, 10, 21, 8, 12, 11, 16, 94, 141), (15, 2, 23, 24, 25, 26, 46, 47, 50, 48, 51, 49), (52, 60, 45, 59, 81, 36, 63, 69, 70), (29, 33, 44, 37, 35, 53, 65, 54, 61, 62, 57, 64, 66), (71, 72, 75, 76, 105, 125, 113, 121, 122), (107, 111, 108, 77, 78, 83, 133, 84), (82, 132, 79, 124, 127, 142, 138), (55, 56, 147, 139, 131, 140, 137, 129, 145, 130, 136, 102), (92, 85, 67, 126, 58, 27, 28, 120, 32, 42, 31, 68, 30, 39, 41, 38), (34, 43, 143, 112, 114, 98, 99, 100, 128, 109), (146, 119, 134, 117, 115, 86, 89, 87, 88), (110, 73, 80, 103, 101, 135, 90, 91, 93, 95, 144, 96), (148, 149, 150, 116, 118, 123, 97, 104, 106, 40, 74)
Z	(1, 16, 13, 12, 14, 17, 19, 15, 11, 10, 5, 4, 21), (18, 20, 8, 3, 94, 141, 2, 23), (24, 25, 7, 26, 63, 60, 43, 30, 37, 59, 36, 31, 33, 81, 41, 32), (39, 34, 68, 38, 35, 29, 27, 28, 46, 48, 52, 49, 70, 69), (47, 50, 58, 44, 40, 42, 22, 6, 51, 53, 61, 55), (56, 64, 71, 72, 75, 73, 85, 80, 103, 101, 98, 99), (97, 74, 92, 102, 77, 86, 87, 105, 123), (107, 121, 112, 114, 115, 111, 116, 78, 124), (119, 82, 79, 83, 133, 84), (117, 122, 108, 65, 132, 89, 125, 100, 148, 149, 150, 109, 90, 104), (66, 67, 126, 45, 127, 136, 62, 110, 113, 118, 76, 129, 128), (139, 131, 137, 135, 146, 140, 138, 130, 145, 91, 93), (95, 144, 96, 147, 134, 57, 88, 9, 143, 106, 142, 120, 54)	(1, 15, 16, 4, 13, 12, 9, 2, 23, 22, 14), (18, 24, 10, 3, 25, 26, 33, 45, 40, 29, 63, 35, 41, 32), (31, 60, 30, 43, 44, 8, 94, 141, 6, 19, 20), (81, 11, 46, 47, 48, 51, 49, 69, 50, 70, 52, 53, 64), (54, 55, 61, 65, 66, 67, 126, 62, 7, 57, 59), (71, 72, 77, 78, 83, 133, 85, 86, 89, 87, 88), (105, 107, 111, 113, 121, 119, 92, 75, 97), (98, 100, 102, 73, 74, 80, 103, 90, 5, 36, 37, 42, 34), (17, 38, 84, 124, 127, 128, 138), (142, 134, 136, 146, 140, 130, 143, 145, 147, 129, 137, 139), (82, 132, 79, 148, 149, 150, 135, 39, 104, 101, 106), (125, 112, 118, 114, 115, 116, 117, 123, 122, 131, 27, 28), (91, 93, 95, 96, 144, 120, 68, 108, 110, 109, 56, 76, 58, 21, 99)

Pareto fronts for both demand scenarios of models. This indicates that the results from multi-ABC algorithm are more towards near optimality even the demand of models is changed. Furthermore, Figs. 7 and 8 also illustrate that the Pareto front obtained from NSGA II is at a higher distance from the true Pareto front. These graphical results clearly indicate the significance of the results obtained from multi-ABC algorithm as compared to NSGA II results. Moreover, these results also suggest that GD values obtained from multi-ABC are be smaller as compared to GD values of NSGA II for both demand scenarios. Furthermore, from Figs. 7 and 8, the Pareto solution points obtained from multi-ABC are more evenly distributed on the front and there is larger number of Pareto solutions on the front as compared to the Pareto fronts obtained by NSGA II.

4.2 Mixed model assembly line based on case company data

Mixed model assembly line of a case company in China is investigated which produces three different models X, Y and Z. Each model contains 150 numbers of tasks processed in 13 stations. The precedence relation among the tasks of all models is the same. The precedence relation among tasks and task time data of each model is indicated in Table 4. The precedence relation between tasks of all models is represented in parentheses. As can be seen from Table 4, the first precedence relation is given as 1 and 2 and illustrates that task 1 is an immediate predecessor of task 2. The task time of each task for different models is separated by a comma in Table 4.

Table 6 Sequencing and balancing results for the second scenario of demand of models

	Multi-ABC	NSGA II
Number of models	3	3
Demand (X, Y, Z)	3, 1, 2	3, 1, 2
Model sequence	X, X, Z, X, Y, Z	Z, X, X, X, Y, Z
Objective values	(1.97646, 1.59677, 21,276)	(1.97935, 1.62031, 22,666)
Tasks of models assigned on all stations		
X	(1, 2, 23, 24, 3, 25), (15, 17, 6, 26, 27, 34, 63), (60, 35, 46, 47, 49, 48, 69), (70, 45, 58, 43, 21, 22, 7, 59, 51, 38, 52, 50, 53, 55, 64, 61, 54), (57, 33, 71, 72, 92, 119, 105, 123), (113, 65, 121, 108, 109, 77, 78), (83, 84, 124, 127, 137, 138, 145), (139, 97, 12, 28, 29, 31, 42, 32, 62, 19, 8, 44, 5, 36, 102, 9, 129, 125, 11, 128, 37, 39, 75, 106, 13, 16, 94), (141, 18, 40, 143, 147, 30, 110, 4, 81, 10, 76, 41, 98, 100, 148, 134, 66, 149, 67), (142, 140, 99, 120, 73, 74, 130, 80), (101, 104, 82, 132, 90, 91, 93, 95, 96), (126, 103, 112, 114, 20, 116, 107, 135, 111, 122, 118, 117, 85, 86, 115, 79, 68, 89, 56, 144, 133), (136, 146, 14, 150, 131, 87, 88)	(1, 2, 23, 24, 3, 25), (15, 17, 6, 26, 27, 34, 63), (60, 35, 46, 47, 49, 48, 69), (70, 45, 58, 43, 21, 22, 7, 59, 51, 38, 52, 50, 53, 55, 64, 61, 54), (57, 33, 71, 72, 92, 119, 105, 123), (113, 65, 121, 108, 109, 77, 78), (83, 84, 124, 127, 137, 138, 145), (139, 97, 12, 28, 29, 31, 42, 32, 62, 19, 8, 44, 5, 36, 102, 9, 129, 125, 11, 128, 37, 39, 75, 106, 13, 16, 94), (141, 18, 40, 143, 147, 20, 30, 110, 4, 81, 10, 76, 41, 98, 100, 148, 134, 66, 126, 67), (142, 140, 99, 120, 73, 74, 130, 80), (101, 104, 82, 132, 90, 91, 93, 95, 96), (149, 103, 112, 114, 116, 107, 135, 111, 122, 118, 117, 85, 86, 115, 79, 68, 89, 56, 144, 133), (136, 146, 14, 150, 131, 87, 88)
Y	(1, 16, 2, 23, 24, 3, 25, 10, 94, 141), (15, 22, 18, 21, 26, 40, 35, 33, 60, 63, 27), (28, 31, 13, 43, 46, 5, 37, 47, 50, 58, 51, 4, 9, 8, 7, 68, 59, 6, 17, 32, 45, 81, 49, 11, 20, 44), (34, 48, 69, 70, 52, 53), (64, 71, 72, 98, 105, 122, 112, 77, 76, 102, 57), (61, 62, 65, 66, 126, 107, 113, 78, 79, 124, 29), (121, 111, 125, 123, 82, 100, 85, 106, 73, 74, 80), (103, 83, 133, 127, 143, 128), (136, 54, 138, 129, 137, 30), (99, 86, 145, 140, 130, 146, 90, 91, 119, 148, 139), (114, 117, 118, 92, 93, 104, 38, 149, 150, 142), (131, 95, 132, 84, 96, 134, 120), (101, 67, 41, 42, 55, 56, 135, 36, 108, 110, 109, 115, 75, 97, 147, 89, 87, 39, 144, 19, 88, 14, 116, 12)	(1, 16, 2, 23, 24, 3, 25, 18, 15, 26, 94), (141, 10, 21, 40, 29, 35, 33, 60, 63, 22, 27), (28, 31, 13, 43, 46, 5, 37, 47, 50, 58, 51, 4, 9, 8, 7, 68, 59, 6, 17, 32, 81, 49, 11, 39, 20, 44), (34, 48, 69, 70, 52, 53), (64, 71, 72, 98, 105, 122, 112, 77, 76, 102, 57), (61, 62, 65, 66, 126, 107, 113, 78, 124, 79), (121, 111, 125, 123, 82, 100, 85, 106, 73, 74, 80), (103, 83, 133, 127, 143, 128), (136, 54, 138, 129, 137, 30), (99, 147, 145, 140, 130, 146, 90, 91, 119, 148, 139, 114), (115, 117, 118, 92, 93, 104, 38, 149, 150, 142), (131, 95, 132, 84, 96, 134, 120), (101, 36, 55, 56, 41, 12, 97, 144, 67, 42, 135, 108, 109, 110, 75, 86, 89, 87, 45, 19, 88, 14, 116)
Z	(1, 3, 14, 4, 20, 18, 6, 11, 2, 16, 94, 17), (15, 23, 24, 25, 26, 32, 46, 52, 47, 58, 50, 48, 49, 51), (42, 59, 43, 29, 45, 33, 9, 5, 22, 13, 53, 61, 54), (65, 55, 62, 60, 34, 31, 27), (28, 44, 63, 69, 35, 56, 39, 70, 64, 66, 67, 71, 72), (105, 111, 92, 75, 113, 106, 108, 110, 37, 36, 68, 76, 121), (102, 85, 109, 97, 125, 98, 99, 100, 148, 81, 77, 78, 79), (83, 84, 120, 107, 86, 89, 87, 126, 82), (132, 88, 133, 10, 38, 149, 150, 19, 124), (123, 41, 57, 112, 114, 116, 115, 117, 118, 21, 73, 74, 127), (135, 131, 142, 129, 145, 143, 30, 122, 40, 136, 128, 137), (134, 146, 139, 119, 138, 80, 90, 101, 103), (104, 12, 91, 93, 95, 144, 96, 147, 130, 7, 140, 8, 141)	(1, 3, 14, 4, 20, 18, 6, 11, 2, 16, 94, 17), (15, 23, 24, 25, 26, 32, 46, 52, 47, 58, 50, 48, 49, 51), (42, 59, 43, 29, 45, 33, 9, 5, 22, 13, 53, 61, 54), (65, 55, 62, 60, 34, 31, 27), (28, 44, 63, 69, 35, 56, 39, 70, 64, 66, 67, 71, 72), (105, 111, 92, 75, 113, 106, 108, 110, 37, 36, 68, 76, 121), (102, 85, 109, 97, 125, 98, 99, 100, 148, 81, 77, 78, 79), (83, 84, 120, 107, 86, 89, 87, 126, 82), (132, 88, 133, 10, 38, 149, 150, 19, 124), (123, 41, 57, 112, 114, 116, 115, 117, 118, 21, 73, 74, 127), (131, 142, 129, 145, 143, 30, 122, 40, 136, 128, 137, 134), (146, 139, 119, 138, 135, 80, 90, 101, 103), (104, 12, 91, 93, 95, 144, 96, 147, 130, 7, 140, 8, 141)

4.2.1 Mixed model assembly line simultaneous sequencing and balancing results

Sequencing and balancing problem of assembly line in case company is also analysed with two different demand scenarios of models. Table 5 illustrates only one solution from the set of Pareto solutions in detail to show model sequencing results of different models in assembly line and the tasks assigned to each station of each model for the first demand scenario of models for both multi-ABC algorithm and NSGA II. It can be

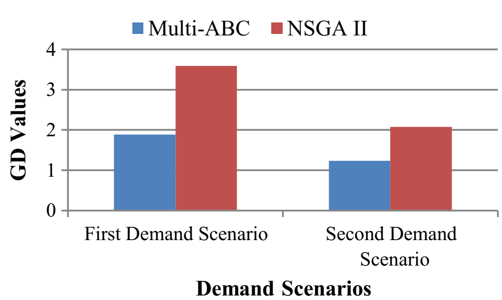


Fig. 9 GD values for different scenarios of demand of models

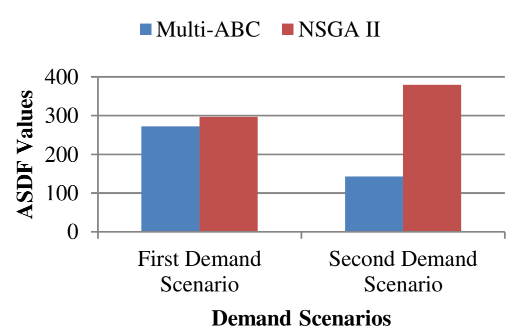


Fig. 10 ASDF values for different scenarios of demand of models

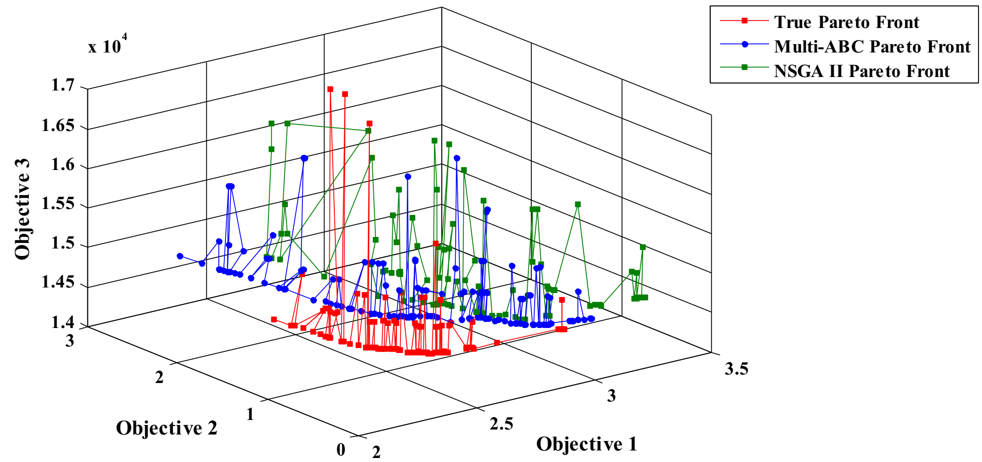


Fig. 11 Graphical representation and comparison of Pareto fronts for the first demand scenario

seen from Table 5 that the demand of model X is 2, the demand of model Y is 2, and the demand of model Z is 1 in one assembly cycle. It can be seen from the results shown in Table 5 corresponding to multi-ABC algorithm that the tasks of model X which are assigned to the first station are 1, 2, 15, 9, 7, 17 and 3 in the assembly line. While the tasks of model X which are assigned to the second station are 23, 24, 25, 26, 46, 52, 49, 51, 48, 59, 8 and 63, the tasks of model X for the third station are 70, 69, 60, 68, 29, 34, 47, 58 and 50 and so on.

Table 6 illustrates one solution from the set of Pareto solution in detail to describe the sequencing results of different models in assembly line and the tasks assigned to each station of each model for the second demand scenario of models for both multi-ABC algorithm and NSGA II. It can be seen from Table 6 that the demand of model X is 3, the demand of model Y is 1, and the demand of model Z is 2 in one assembly cycle. It can be seen from the results shown in Table 6 corresponding to multi-ABC algorithm that the tasks of model X which are assigned to the first station are 1, 2, 23, 24, 3 and 25 in the assembly line, while the tasks of model X which are assigned to the second station are 15, 17, 6, 26, 27, 34 and 63 and so on.

4.2.2 Comparison of results

Comparison of results based on inverted generational distance: The results of proposed multi-ABC and NSGA II algorithm on the basis of GD value is indicated in Fig. 9. It can be seen from Fig. 9 that the proposed multi-ABC algorithm gives smaller GD values on both scenarios. These results indicate that the proposed multi-ABC algorithm can give more near-optimal Pareto results for both scenarios of demands as compared to NSGA II.

Comparison of results based on average spacing deviation function: The results of proposed multi-ABC algorithm on the basis of ASDF are compared with the results obtained from NSGA II in Fig. 10. It can be seen from Fig. 10 that proposed multi-ABC algorithm gives a smaller value of ASDF as compared to NSGA II for both scenarios of demand.

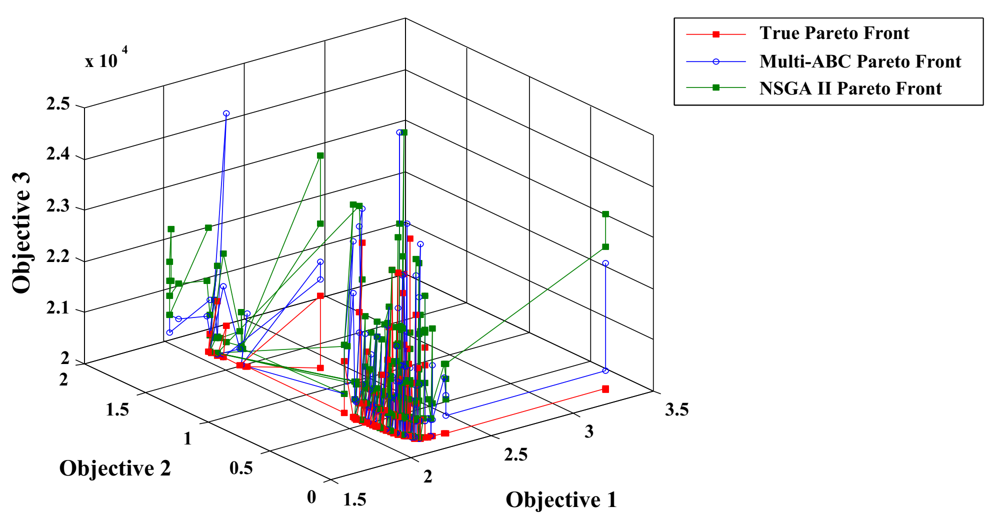


Fig. 12 Graphical representation and comparison of Pareto fronts for the second demand scenario

These results indicate that multi-ABC outperforms NSGA II on the basis of ASDF and can give more evenly distributed Pareto solutions on the Pareto front.

Comparison of results based on Pareto fronts generated: The graphical representation of Pareto fronts obtained from the proposed multi-ABC algorithm and NSGA II for mixed model assembly line data of the case company is illustrated in Figs. 11 and 12 for the first and second demand scenarios of models, respectively. It can be seen from Figs. 11 and 12 that the Pareto fronts generated from multi-ABC algorithm are more towards true Pareto fronts for both demand scenarios of models. These results indicate that the Pareto solutions obtained from multi-ABC algorithm are more towards near optimality even the demand of models is changed. These graphical results clearly show the significance of the results obtained from multi-ABC as compared to NSGA II results.

5 Conclusion

In recent years, due to an increased demand of customized products, mixed model assembly lines have got a lot of attention. Mixed model assembly lines have two problems, i.e. model sequencing and balancing problems. These two problems are simultaneously addressed in the current research. Three objectives are concurrently studied for simultaneous consideration of model sequencing and balancing problems in mixed model assembly line. Current research presented the issue to balance workload for different models on each station to reduce the possibility of generation of incomplete units. In addition to this, the current research aimed to reduce the deviation of workload of stations from the average workloads of all the stations and to minimize total flow time of models on stations simultaneously. A multi-objective artificial bee colony (multi-ABC) algorithm which can optimize these three objectives simultaneously is proposed to generate a Pareto solution for the current problem. A local search mechanism is incorporated in the proposed multi-ABC algorithm to search different sequencing solutions for each balanced solution simultaneously. Two different mixed model assembly line problems are solved to test the performance of the proposed multi-ABC algorithm. For the first type of mixed model assembly line problem, each model task time data and precedence relation data are taken from standard assembly line problems taken from operation research library (OR). For the second mixed model assembly line problem, each model task time data and precedence relations are taken from a truck manufacturing company in China.

The performance of the proposed multi-ABC algorithm is tested against the performance of a famous multi-objective algorithm, i.e. NSGA II, to solve these two problems on two different model demand scenarios. End results indicated that the proposed multi-ABC algorithm outperforms NSGA II to generate better near-optimal Pareto solutions for the current problem on both demand scenarios. Future research can be extended to include task time uncertainty to generate a robust version of simultaneous sequencing and balancing problem. Moreover, the performance of algorithm can be improved by integrating artificial bee colony algorithm with some variable neighbourhood search schemes to design a hybrid algorithm for the better exploration of the search space.

Acknowledgments This work has been supported by MOST (the Ministry of Science & Technology of China) under the grants No. 2013AA040206, 2012BAF12B20, & 2012BAH08F04, and by the National Natural Science Foundation of China (Grants No. 51035001, 51121002 & 71131004).

References

1. Simaria AS, Vilarinho PM (2004) A genetic algorithm based approach to the mixed model assembly line balancing problem of type II. Comput Ind Eng 47:391–407
2. Celano G, Costa A, Fichera S (2004) A comparative analysis of sequencing heuristics for solving the Toyota goal chasing problem. Robot Comput Integr Manuf 20:573–581
3. Mansouri SA (2005) A multi-objective genetic algorithm for mixed-model sequencing on JIT assembly lines. Eur J Oper Res 167:696–716
4. Al-e-hashem SMJM, Aryanezhad MB, Jabbarzadeh A (2011) A new approach to solve a mixed-model assembly line with a bypass sub line sequencing problem. Int J Adv Manuf Technol 52:1053–1066
5. Dar-El EM, Nadivi A (1981) A mixed-model sequencing application. Int J Prod Res 19:69–84
6. Kim YK, Hyun CJ, Kim Y (1996) Sequencing in mixed model assembly lines: a genetic algorithm approach. Comput Oper Res 23:1131–1145
7. Karabati S, Sayin S (2003) Assembly line balancing in a mixed-model sequencing environment with synchronous transfers. Eur J Oper Res 149(2):417–429
8. Kim YK, Kim JY, Kim Y (2000) A coevolutionary algorithm for balancing and sequencing in mixed model assembly lines. Appl Intell 13:247–258
9. Kim YK, Kim SJ, Kim JY (2000) Balancing and sequencing mixed-model U-lines with a co-evolutionary algorithm. Prod Plan Control 11:754–764
10. Sawik T (2000) Simultaneous vs. sequential loading and scheduling of flexible assembly systems. Int J Prod Res 38:3267–3282
11. Sawik T (2002) Monolithic vs. hierarchical balancing and scheduling of a flexible assembly line. Eur J Oper Res 143:115–124
12. Sawik T (2004) Loading and scheduling of a flexible assembly system by mixed integer programming. Eur J Oper Res 154:1–19
13. Kim YK, Kim JY, Kim Y (2006) An endosymbiotic evolutionary algorithm for the integration of balancing and sequencing in mixed-model U-lines. Eur J Oper Res 168:838–852
14. Mosadegh HM, Zandieh SMT, Ghomi F (2012) Simultaneous solving of balancing and sequencing problems with station-dependent assembly times for mixed-model assembly lines. Appl Soft Comput 12:1359–1370

15. Ozcan U, Cercioglu H, Gokcen H, Toklu B (2010) Balancing and sequencing of parallel mixed-model assembly lines. Int J Prod Res 48(17):5089–5113
16. Hwang R, Katayama H (2010) Integrated procedure of balancing and sequencing for mixed-model assembly lines: a multi-objective evolutionary approach. Int J Prod Res 48:6417–6441
17. Öztürk C, Tunali S, Hnich B, Örnek MA (2013) Balancing and scheduling of flexible mixed model assembly lines with parallel stations. Int J Adv Manuf Technol 67:2577–2591
18. Öztürk C, Tunali S, Hnich B, Örnek MA (2013) Balancing and scheduling of flexible mixed model assembly lines. Constraints 18(3):434–469
19. Merengo C, Nava F, Pozzetti A (1999) Balancing and sequencing manual mixed-model assembly lines. Int J Prod Res 37(12):2835–2860
20. Yang C, Gao J, Sun L (2013) A multi-objective genetic algorithm for mixed-model assembly line rebalancing. Comput Ind Eng 65(1): 109–116
21. Wu EF, Jin J, Bao JS, Hu XF (2008) A branch-and-bound algorithm for two-sided assembly line balancing. Int J Adv Manuf Technol 39(9–10):1009–1015
22. Miltenburg J (2002) Balancing and scheduling mixed-model U-shaped production lines. Int J Flex Manuf Syst 14:119–151
23. Kara Y (2008) Line balancing and model sequencing to reduce work overload in mixed-model U-line production environments. Eng Optim 40(7):669–684
24. Wang, Y., Dang, C., Li, H., Han, L., Wei, J (2009) A clustering multi-objective evolutionary algorithm based on orthogonal and uniform design. In: Proceeding of IEEE congress on evolutionary computation, Trondheim, Norway, 18–21 May, 2009, 2927–2933
25. Liu, M., Zou, X., Chen, Y., Wu, Z (2009) Performance assessment of DMOEA-DD with CEC 2009 MOEA competition test instances. In: Proceeding of IEEE congress on evolutionary computation, Trondheim, Norway, 18–21 May, 2009, 2913–2918
26. Kukkonen, S., Lampinen, J (2009) Performance assessment of generalized differential evolution with a given set of constrained multi-objective test problems. In: Proceeding of IEEE congress on evolutionary computation, Trondheim, Norway, 18–21 May, 2009, 1943–1950
27. Chen, C.M., Chen, Y., Zhang, Q (2009) Enhancing MOEA/D with guided mutation and priority update for multi-objective optimization. In: Proceeding of congress on evolutionary computation, Trondheim, Norway, 18–21 May, 2009, 209–216
28. Guo ZX, Wong WK, Li Z, Ren P (2013) Modeling and Pareto optimization of multi objective order scheduling problems in production planning. Comput Ind Eng 64:972–986
29. Guo ZX, Wong WK, Leung SYS (2013) A hybrid intelligent model for order allocation planning in make to order manufacturing. Applied Soft Computing 13:1376–1390
30. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–97
31. Baykasoglu A, Dereli T (2008) Two-sided assembly line balancing using an ant-colony-based heuristic. Int J Adv Manuf Technol 36(5–6):582–588
32. Agrawal S, Dashora Y, Tiwari MK, Son YJ (2008) Interactive particle swarm: a Pareto-adaptive metaheuristic to multiobjective optimization. IEEE Trans Syst Man Cybern A 38(2):258–277
33. Coello Coello CA, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. IEEE Trans Evol Comput 8(3):256–279
34. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report TR06, Computer Engineering Department. Erciyes University, Turkey
35. Karoboga and Gorkemli, B (2008) A combinatorial artificial bee colony bee algorithm for traveling salesman problem. 2011 international symposium on innovations in intelligent systems and applications (INISTA)
36. Omkar S, Senthilnath N, Khandelwal JR, Narayana Naik G, Gopalakrishnan S (2011) Artificial bee colony (ABC) for multi-objective design optimization of composite structures. Appl Soft Comput 11(1):489–499
37. Akbari R, Hedayatzadeh R, Ziarati K, Hassanizadeh B (2012) A multi-objective artificial bee colony algorithm. Swarm Evol Comput 2:39–52
38. Tapkan P, Ozbakir L, Baykasoglu L (2012) Modeling and solving constrained two aided assembly line balancing problem via bee algorithms. Appl Soft Comput 12(1):3343–3355
39. Pan QK, Tasgetiren MF, Suganthan PN, Chua TJ (2011) A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. Inf Sci 181(12):2455–2468
40. Li JQ, Pan QK, Gao KZ (2011) Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems. Int J Adv Manuf Technol 55:1159–1169
41. Zhang R, Song S, Wu C (2013) A hybrid artificial bee colony algorithm for job shop scheduling problem. Int J Prod Res 141(1): 167–178
42. Wang L, Zhou G, Xu Y, Wang S (2012) An effective artificial bee colony algorithm for flexible job shop scheduling problem. Int J Adv Manuf Technol 60:303–315
43. Wang L, Zhou G, Xu Y, Liu M (2012) An enhanced Pareto-based artificial bee colony algorithm for the multi-objective flexible job-shop scheduling. Int J Adv Manuf Technol 60: 1111–1123
44. Tasgetiren MF, Pan QK, Suganthan PN, Chen AH-L (2011) A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops. Inf Sci 181(16): 3459–3475
45. Kalayci CB, Gupta SM (2013) Artificial bee colony algorithm for solving sequence dependent disassembly line balancing problem. Expert Syst Appl 40(18):7231–7241
46. Vollmer D. T., Soule T., and Manic M (2010) A distance measure comparison to improve crowding in multi-modal optimization problems. In Proceedings of the 3rd international symposium on resilient control systems (ISRCS '10), Idaho Fall, USA, Aug 10–12, 1020, 31–36
47. Scholl, A (1993) Data of assembly line balancing problems. Schriften zur quantitativen betriebswirtschaftslehre 16/93, TU Darmstadt
48. Scholl A (1999) Balancing and sequencing assembly lines, 2nd edn. Physica, Heidelberg
49. Coello CAC, Cortes NC (2005) Solving multi objective optimization problems using an artificial immune system. Genet Program Evolvable Mach 6:163–190
50. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. J Evol Comput 8(2):173–195