

An Assembly Line Balancing Application on Oven Production Line with Hyper-Heuristics

Gökhan Seçme, Nevşehir Hacı Bektaş Veli University, Nevşehir, Turkey

Lale Özbakır, Erciyes University, Kayseri, Turkey

ABSTRACT

In this study, an oven assembly line that is planning to re-establish manufacturing to increase the efficiency of the assembly process. The importance of the problem emerges from a real-world application consisting of product-oriented restrictions. These multiple restricted problems address the single model assignment restricted ALB problem with positional constraints. A cost-based objective function is used to cope with this problem. The number of platformed and non-platformed stations, the number of direction changes in a station, the number of stations in which both connector and combiner are used are the cost factors of the objective function. Also, the main objective of the problem is to minimize the total number of stations while satisfying the restrictions. A simulated annealing-based hyper-heuristic is adapted and applied to the balancing problem of oven manufacturing process with assignments and operational restrictions with multiple objectives. The results show that better solutions can be found in the current line balance level while satisfying more restrictions. It is also observed that line balance can be improved depending on the relaxation of the restrictions.

KEYWORDS

Annealing, Assembly Line Balancing, Hyper-Heuristic Approach, Product/Operation Constraints and Restrictions

INTRODUCTION

The assembly line balancing (ALB) problem determines the workstations, which tasks are assigned to, by considering precedence constraints and cycle time so as to optimize predetermined performance measures (Ghosh & Gagnon, 1989). Many researchers have studied ALB problem since the first time it was introduced in the literature. Baybars (1986) classified ALB problems into two groups: simple assembly line balancing problem (SALBP) and generalized assembly line balancing problem (GALBP). In the SALBP, task times are considered as deterministic and a task can only be assigned to one station. While SALBP-1 aims to minimize the number of workstations for predetermined cycle time, SALBP-2 aims to minimize the cycle time for a given station number. GALBP signifies the any other ALB problem that generalizes SALBP by eliminating some assumptions. There are also many other classifications of ALB problems. The shape of the assembly line is one of the common classifying criteria for ALB problems. Miltenburg & Winjgaard (1994) and Erel, Sabuncuoğlu & Aksu (2001) studied U-line ALB problems. In addition to the single model assembly lines, mixed model (Yano & Bolat, 1989; Bard, Dar-El, & Shtub, 1992) and multi model (Burns & Daganzo, 1987; Dobson & Yano, 1994) assembly lines allow to produce similar or different products on the same line (Bukchin, Dar-El, & Rubinovitz, 2002). A detailed literature review can be found in Becker & School (2006) and School & Becker (2006). Additionally, some other specialized features such as parallel

DOI: 10.4018/IJORIS.2019070104

This article, originally published under IGI Global's copyright on July 1, 2019 will proceed with publication as an Open Access article starting on February 2, 2021 in the gold Open Access journal, International Journal of Operations Research and Information Systems (converted to gold Open Access January 1, 2021), and will be distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

workstations (McMullen & Frazier, 1998), two sided lines (Bartholdi, 1993), parallel assembly lines (Gökçen, Ağpak, & Benzer, 2006), incompatibilities and bounded loads (Pastor & Corominas, 1999), designing assembly lines with equipment selection (Buckhin & Rubinovitz, 2003) and equipment costs (Buckhin & Tzur, 2000) are presented in the assembly line literature.

All these problem types arise from the goal of meeting the real-world problem requirements. Although the ALB literature is very rich, there is still a gap between theoretical studies and real-world applications (Becker & School, 2006; Boysen, Fliedner, & Scholl, 2008). Position and/or accessibility constraints (Lapierre & Luiz, 2004; Essafi, Delorme, Dolgui, & Guschinskaya, 2010) are considered as task assignment restrictions in addition to cycle time and precedence constraints. A certain station – task assignment may occur when the position of the assembling product cannot be changed due to weight or size of the product (School, Fliedner, & Boysen, 2010). Furthermore, zoning constraints can be applied when some group of tasks should be assigned to the same station or incompatible tasks should be assigned to different stations. All these constraints appear in real world applications, so these constraints should be taken into consideration while balancing the assembly lines. In this article, a real-world assembly line balancing problem is handled. An oven producer decided to redesign their assembly lines. They need to learn that whether it is possible to balance the assembly line in higher efficiency level while satisfying the production restrictions learned from previous assembly experiences.

ALB is an NP-hard problem (Karp, 1972) due to its computational complexity. Although technological developments allow us to handle different sized problems, it is still far away to deal with real world large sized problems. Gökçen et al. (2006) stated that the optimal solution of deterministic parallel ALBP was found for only small sized problems. Since the number of operations in real world ALB problems is very high, heuristic solution algorithms have been used to produce solutions within an acceptable time. The meta-heuristics have been widely used in many areas such as vehicle routing problem (Layeb, 2015), supply chain optimization (Gupta, Kundu, & Gupta, 2017) and other subjects are reviewed in Sensuse & Cahyaningsih (2018).

The meta-heuristic approaches such as simulated annealing (Erel, Sabuncuoğlu, & Aksu, 2001; Baykasoğlu, 2006; McMullen & Frazier, 1998; Suresh & Sahu, 1994), ant colony optimization algorithm (McMullen & Tarasevich, 2003; Sabuncuoğlu, Erel, & Alp, 2009; Özbakır, Baykasoğlu, Görkemli, & Görkemli, 2011) have been used in many ALB problems in the literature. The meta-heuristics provide a generic solution mechanism and they are designed and implemented for the problem itself by considering problem specific information. In this study, a hyper-heuristic (HH) solution approach is used to solve the real-life manufacturing line balancing problem concerning the task and station restrictions. Hyper-heuristics are considered as automated solution process that solves the generalized problems independent of problem domain as a high-level solution approach (Burke et al., 2013). Hyper-heuristics provides efficient solutions to the task and station restricted ALB problem due to learning abilities and applicability of general problem area.

This study is an application of simulated annealing based hyper-heuristic approach to the real-life ALB problem. The addressed ALB problem is different from simple ALB problem by the restrictions about the direction of assembly, usage of lifter and operation type. These restrictions are formulated as a mathematical programming approach, but the size of the problem makes heuristic approaches more useful. The developed solution algorithm produces solutions according to the priorities of the company. Decision makers can evaluate alternative solutions for different priorities. The proposed solution approach can be easily adapted and applied to various types of ALB problems with different restrictions.

The paper is organized as follows: the ALB problem handled in this paper is described in detail in section 2. The restrictions on ALB problem and the mathematical model are explained. The proposed hyper-heuristic method and the literature review are given in section 3. The classification of hyper-heuristic methodologies and application examples are also discussed. In section 4, the results obtained by hyper-heuristic algorithm are compared with the current assembly line balances. Properties of the suggested solution are discussed, and some advise is given to the managerial board of the company.

STATEMENT OF THE PROBLEM

The firm produces three different oven models. All three models are similar with each other and the company does not consider them as independent products. So, these models are considered as a single model, which some tasks have zero task time regarding the model characteristics. The task times are determined as 0 for the tasks, which are not included in the specific oven model. Actually, in real world production applications most of the producers apply this perspective. This situation also causes some task – station restrictions.

The task times vary from 4 to 85 for all models. The main difference among the models is the number of tasks to be performed. The F-ratio, indicating the flexibility of network diagram, is calculated as just about 0.69 for all models. This means relatively less precedence constraints and higher flexibility in generating valid solutions. All these information addresses the single model ALB problem with station and task assignment restrictions.

The assumptions of the problem considered in this paper are given as follows:

- A single model is produced in the straight assembly line
- The precedence relations among tasks are known
- The task times are known and deterministic
- A conveyor belt type transporter is used to move the work piece with a certain speed
- Transportation times are negligible and assumed to be included in the task times

Problem Constraints and Restrictions

The problem can be classified as a special version of single model deterministic ALB problem. The restrictions, which make problem difficult to solve, are the requirements of the company about the assembly line and products. The preferences of the company are expressed as follows:

- The assembly of some components is bounded up with the direction of the product/semi product on the line. While some operations can be performed from both sides, others can be operated from front or back side. So, the company requests that the tasks that are assigned to the same station should be operated from same side. This means that the product/semi product should not turn its direction within a station. This situation decreases the time of direction change of product/semi product.
- The assembly operations may require operating from top of the product. The platform is used to reach the top of the product/semi product to perform the related operations. Two types of platforms, low and high platforms, are used to achieve predetermined height for assembly. So, a station should be designed as a platformed or non-platformed station.
- Some components of oven can be assembled by combiner or connector. Four types of combiner and two types of connector are used. Changing the tool in a station may cause waste of time. It

Table 1. Properties of problem data

	Model 1	Model 2	Model 3
Number of tasks	116	108	83
Min task time	4	4	4
Max task time	85	85	85
Average task time	23.29	26.74	29.05
F ratio	0.686	0.690	0.695

is preferred that only same type of assembly operations (combiner or connector) are performed within the same station.

- A few predefined tasks need to be assigned to a station that is not earlier than a predefined station. This helps the company to make small modifications on similar models. Also, some tasks should be assigned to same stations to minimize the process differentiations of producing different models.

While new parts/components used in the assembly, some new task assignment restrictions and positional constraints arise. These restrictions are learned by trial and error method during the production process. The firm handled this problem by considering swapping the tasks or combining/separating the stations. This approach did not meet all the restrictions and constraints as well as it decreased the production rate (increase in the cycle time). It should not be forgotten that these restrictions are preferences of the company and they can be sacrificed according to their priorities.

The restrictions and constraints mentioned above cause some new assumptions and characteristics as follows:

- Some tasks should be assigned to the same station due to need for performing these tasks together or need for the usage of platform
- Some tasks should be assigned to a certain station even if there are available previous stations to be assigned

Except those, in addition to the main objective that minimizes the number of stations for a given cycle time, problem contains multiple objectives with different priorities.

Problem Formulation

The general mathematical model of the type I ALB problem aims to minimize the number of stations by considering the precedence relations and cycle time restriction. The station/task restrictions regarding the preferences of the company are included in the model as constraints. The general integer programming model of the problem is formulated below. The notation used in the formulation is described in Table 2.

The proposed mathematical model formulation is as following:

$$\text{Min } \sum_{j \in J} Y_j \text{ Subject to} \tag{1}$$

$$\sum_{j=1}^J x_{ij} = 1 \quad \forall i \tag{2}$$

$$\sum_{i=1}^N t_i x_{ij} \leq CY_j \quad \forall j \tag{3}$$

Table 2. Problem notation

C	Cycle time
N	Total number of tasks
i	Task index
j	Station index
t_i	Operation time of task i ($i=1, \dots, N$)
x_{ij}	1 if task i assigned to station j , 0 otherwise
a_j	Station index for non-platformed stations
b_j	Station index for combiner used (connector free) stations
M	A very big number
A_p	Set of precedence relations between tasks (m,n) with $m,n \in i$
Y_j	Usage of station j
B^*	Set of tasks that can be operated from back side of the work piece
F^*	Set of tasks that can be operated from front side of the work piece
M^*	Set of tasks that can be operated from both back and front side of the work piece
P^*	Set of tasks that can be operated by platform
SR^*	Set of tasks that can be assembled by a connector
WS^*	Set of work stations which task i can be assigned to
T^*	Set of tasks that should be assigned to same station

$$\sum_{j=1}^J j x_{mj} - \sum_{j=1}^J j x_{nj} \geq 0 \quad \forall (m,n) \text{ such that } m \text{ immediately precedes } n; m,n \in A_p \quad (4)$$

$$\sum_{i \in B^*} t_i x_{ij} + \sum_{i \in M^*} t_i x_{ij} \geq \sum_{i \in F^*} C x_{ij} \quad \forall j \quad (5a)$$

$$\sum_{i \in F^*} t_i x_{ij} + \sum_{i \in M^*} t_i x_{ij} \geq \sum_{i \in B^*} C x_{ij} \quad \forall j \quad (5b)$$

$$\sum_{i \in P^*} x_{ij} - \sum_{i \notin P^*} x_{ij} C \geq -M a_j \quad \forall j \quad (6a)$$

$$\sum_{i \notin P^*} x_{ij} - \sum_{i \in P^*} x_{ij} C \geq -M (1 - a_j) \quad \forall j \quad (6b)$$

$$\sum_{i \in SR^*} x_{ij} - \sum_{i \notin SR^*} x_{ij} C \geq -M(b_j) \quad \forall j \quad (7a)$$

$$\sum_{i \notin SR^*} x_{ij} - \sum_{i \in SR^*} x_{ij} C \geq -M(1 - b_j) \quad \forall j \quad (7b)$$

$$x_{ij} = 0 \quad \forall i, j, \quad j \notin ws^*(i) \quad (8)$$

$$x_{ij} = x_{mj} = \dots = x_{zj} \quad \forall j, \forall (i, m, \dots, z) \in T^* \quad (9)$$

$$x_{ij}, Y_j, a_j, b_j = \{0, 1\} \quad \forall i, j \quad (10)$$

The objective of the model is to minimize the number of stations. Constraint 2 ensures that each task is assigned to exactly one station. Constraint 3 satisfies the cycle time restriction that workload of each station cannot exceed the predefined cycle time. Constraint 4 provides that all precedence relations are satisfied. Constraints 5a and 5b ensure that only tasks that can be operated from same side (front or back) are assigned to same station. Constraint 6a and 6b ensure that the tasks that need platform are assigned to same station, and vice versa. Constraint 7a and 7b ensure that the tasks, which assembled by connectors, are assigned to same station. It is also valid for combiners. Constraint 8 ensures that a task cannot be assigned before the assignable station. Constraint 9 ensures that some tasks should be assigned to the same stations. Constraint 10 describes the binary variables. Also note that WS^* depends on i .

Since the problem is NP-Hard, the time needed for the solution rises by the number of tasks increase. Nevertheless, under the hard constraints and restrictions, any solution may not be provided. The relaxation of the constraints makes model more flexible. The deviation from the constraints and restrictions can be a measure of how a good solution is reached. An efficient way to calculate the deviation from objectives (constraints and restrictions) is to use a cost based objective function. Using cost based objective function is an efficient way to solve the ALB problems especially with multiple objectives (Amen, 2001). The cycle time can be used as a common penalizing measure for violations of goals. The objective function for solving the problem in proposed solution approach is determined as follows:

Cost1 (minimizing the number of stations) = cycle time (C) x number of stations ($\sum Y_j$)

Cost2 (minimizing the number of direction change in a station) = cycle time (C) x number of direction change in a station (R)

Cost3 (minimizing the number of stations having both equipped with platform and not equipped with platform) = (cycle time (C) / 2) x number of station with and without equipped with platform (P)

Cost4 (minimizing the number of combiner/connector changes within a station) = (cycle time (C) / 3) x number of combiner/connector change in a station (SR)

$$\min Cost = Cost1 + Cost2 + Cost3 + Cost4$$

$$\min Cost = [c \times J] + [c \times R] + \left[\left(\frac{c}{2} \right) \times P \right] + \left[\left(\frac{c}{3} \right) \times SR \right]$$

R: number of direction turns in a station ($R=1\dots r$)

P: number of platform changes in a station ($P=1\dots p$)

SR: number of combiner/connector in a station ($SR=1\dots sr$)

As seen from the cost function, objectives are weighted by the ratio of cycle time according to their priorities, which are defined by the company preferences. Task times, operation direction of tasks to be performed, need for the platform and the combiner/connector changes are the parameters of problem and are already known. Some tasks can be performed from both directions while others should be performed from one direction. The solution should determine the station assignments of tasks with direction changes, the usage of platform in the station and the usage of combiner or connector within the station.

THE PROPOSED SOLUTION METHOD

A hyper-heuristic algorithm based on simulated annealing is used to solve the problem. The developed hyper-heuristic algorithm by Seçme (2015) operates independent of ALB specifications. The proposed algorithm does not need particular problem domain knowledge. Therefore, it can be easily applied to other types of ALB problems with minor changes. The hyper-heuristic module evaluates the low-level heuristics without any domain dependent information. The problem specific restrictions and properties can be added to the task selection and assignment module. This makes the hyper-heuristics more general and adaptive.

Hyper-heuristics

Hyper-heuristic is defined as “a search method or learning mechanism for selecting or generating heuristics to solve hard computational search problems” (Burke et al., 2010). Hyper-heuristic is a general problem-solving approach and it is independent from problem domain (Burke et al., 2003). The goal of the hyper-heuristics is to solve the problems with automated processes in higher generality levels (Burke et al., 2007). Hyper-heuristic performs a search on the space of low-level heuristics rather than the solution space of the problem (Burke et al., 2010; 2013). The main idea is to develop a generic applicable algorithm to solve problems without any specifications (Burke et al., 2013).

The motivation behind the hyper-heuristic research is heuristics having own strengths and weaknesses on different types of problems. If the good combination of heuristics can be found or a new heuristic can be developed by combining heuristics, this new heuristic performs better at general problem domain.

A hyper-heuristic can be considered as a high-level heuristic which selects (meta-) heuristics to solve the given optimization problem. A hyper-heuristic can be seen as an algorithm (on a higher level), which “picks” appropriate heuristics (at a lower level) to be applied to the problems in hand (Burke et al., 2013). A hyper-heuristic is concerned with searching on the low-level heuristics space instead of directly searching on the solution space of the problem.

The classification of hyper-heuristics has two dimensions, i) the nature of heuristic search space, ii) the source of feedback information, which is orthogonal to each other (Burke et al., 2010). The nature of the search space can be divided into two methodologies i) heuristic selection methodologies, ii) heuristic generation methodologies. Each methodology consists of two same subgroups, i) constructive heuristics, ii) perturbative heuristics. Perturbative heuristics deal with complete candidate solution

and reaches new candidate solution by making small changes in solution components. Constructive heuristics consider partial solutions by iteratively extending them.

- The nature of heuristic search space
- Heuristic selection methodologies
 - Constructive heuristics
 - Perturbative heuristics
- Heuristic generation methodologies
 - Constructive heuristics
 - Perturbative heuristics
- The source of feedback information
- Online learning
- Offline learning
- No learning

A learning mechanism can be used as a guide to select efficient heuristics into the hyper-heuristic. While the learning takes place during the search process in online learning, information that gathered from a set of training instances before the search process is an example of offline learning. This classification does not restrict with the methodologies in any of categories. Hybrid methodologies can include both constructive and perturbative strategies or heuristic selection and heuristic generation (Burke et al., 2013).

Hyper-heuristics are successfully applied to many optimization problems including educational timetabling, production scheduling, bin packing, constrained satisfaction, and vehicle routing. Detailed information and literature review can be found in Burke et al. (2013).

The Hyper-Heuristic Algorithm to the Constrained/Restricted ALB Problem

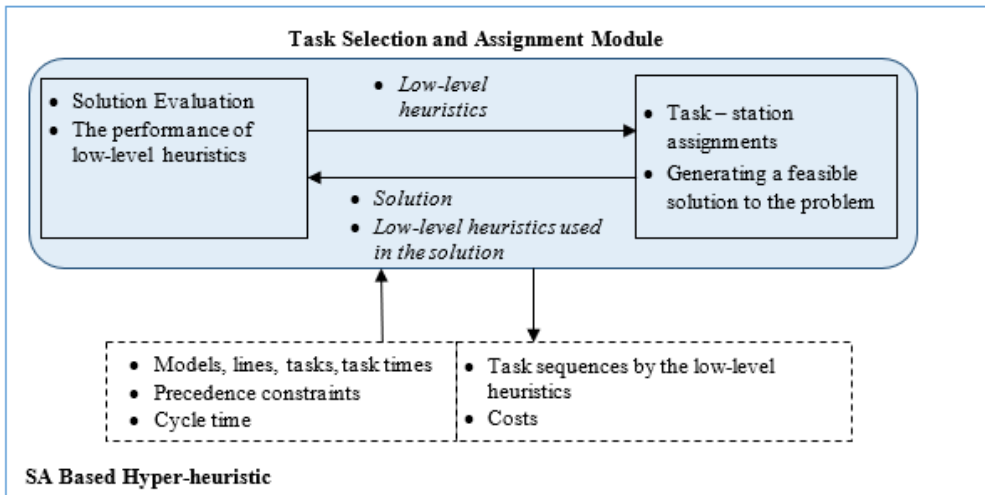
The developed SA based hyper-heuristic method has two modules, simulated annealing module, and task selection and assigning module (Figure 1). The simulated annealing module works as a high-level meta-heuristic to search on the order of low-level heuristics. It also produces neighborhood solutions and calculates the performance of the ordered low-level heuristics. The task selection and assignment module searches on the problem solution space and selects feasible solution to the problem which provides task – station assignment.

The procedure of assigning tasks to the stations is based on the COMSOAL (Arcus, 1966), a well-known heuristic for the ALB problems that selects the tasks from an assignable task list randomly. While determining the assignable tasks, constraints and restrictions are also taken into consideration. The performance of the neighborhood solution is calculated by the objective function in the task selection and assignment module.

The well-known sequencing heuristics are used to select a task from the assignable tasks set. These low-level heuristics are determined as listed below:

- Shortest Processing Time (SPT)
- Longest Processing Time (LPT)
- Maximum Total Number of Successor Tasks (MaTNST)
- Maximum Total Time of Successor Tasks (MaTTST)
- Maximum Total Number of Predecessor Tasks (MaTNPT)
- Maximum Total Time of Predecessor Tasks (MaTTPT)
- Minimum Total Number of Successor Tasks (MiTNST)
- Minimum Total Time of Successor Task (MiTTST)
- Minimum Total Number of Predecessor Tasks (MiTNPT)

Figure 1. Hyper-heuristic mechanism



- Minimum Total Time of Predecessor Tasks (MiTTPT)

The sequence of the low-level heuristics with respect to assignment position is the main factor on the performance of a solution. For tracking the performances of various low-level heuristic sequences, a HRM – heuristic rating matrix is used (Figure 2). During the iterative progress of algorithm, these values diverge from each other by updating them according to the heuristic sequence and success or failure of the solution. The values in the HRM display the ratio of eligibility of the low-level heuristics in the heuristic vector in assigning the tasks to the corresponding position. At the beginning of the iterations, all the values in HRM matrix are determined as equal. The values would change by iterations according to the performance of generated solutions by the sequence of low-level heuristics.

The performance of the neighborhood solution is evaluated by the value of objective function (cost). The better objective function value means that the sequence of the low-level heuristics,

Figure 2. Heuristic rating matrix

	Position 1	Position 2	Position 3	...	Position <i>N</i>
SPT	10	18	10	...	16
LPT	15	14	10	...	12
MiTNST	12	18	14	...	16
MaTNST	16	9	8	...	46
MaTTST	24	5	6	...	24
MiTTST	36	5	2	...	18
MaTNPT	4	5	24	...	18
MiTNPT	2	10	14	...	42
MaTTPT	22	13	6	...	10
MiTTPT	9	36	10	...	36
MaMC	14	15	12	...	16

which moves the algorithm to the better solution, provides low cost solution. The performance of low-level heuristics sequence is recorded with respect to quality of solution at each iteration step. A well performed low-level heuristics sequence promoted to make it more eligible at next iteration. As opposed to that, low-level heuristics, which produced a non-improved solution, is penalized to decrease the eligibility. Hyper-heuristic selects a combination of low-level heuristics, which is applied to the problem with respect to its performances. A roulette wheel selection method is used to select a low-level heuristic for a certain position of assignment. For more detailed explanation of the algorithm execution, please refer to Seçme (2015).

EXPERIMENTAL RESULTS AND DISCUSSION

The developed hyper-heuristic algorithm coded in Matlab and applied to the real-world problem that is addressed in this study. The data of three models are executed on the algorithm. The parameters used in the algorithm are determined by the previous research presented by authors about the SA and hyper-heuristics (Seçme, 2015). The algorithm is executed on Intel Core i3 3.10 GHz PC with 4 GB RAM. The termination criterion of the algorithm is specified as to reach successive 500 non-improved solutions. A number of experimental trials are executed to determine the temperature level and other standard parameters for SA algorithm. The experimental results and observations are as follows:

- The solutions obtained by the hyper-heuristic solution approach are better than the currently applied system. Even the current system does not consider the restrictions; algorithm provides lower station numbers with fully meeting the direction change and platform usage restrictions.
- The algorithm produced solutions compatible with objective function. The solutions with lower cost contributions are preferred by the algorithm. Therefore, the effect of cost parameters is taken into account by the solution process.
- The ratings (eligibility) of the low-level heuristics are clearly separated from each other. This means that the hyper-heuristic used the low-level heuristics properly and deliberately.
- The cost based objective function provided the algorithm to drive the solution according to the priority (coefficients) of the restrictions and constraints.
- When the problem solved as a single model ALB problem without specified restrictions, the solutions with less number of stations are found. The problem specific preferences are affecting the solutions directly and the solution approach properly operates regarding these restrictions.

The results of solution algorithm for different cycle times can be seen in Table 3. When the first two restrictions (direction and platform changes) are relaxed, better line balances are obtained. In addition, because of stability and lower weights in objective function of the last two restrictions (combiner change and connector change), we can say that they have relatively low effect on the problem solution. This is a natural result due to the preferences of the company. Without a real-life application case, effects of different objectives could be evaluated by normalizing the components of objective function.

The proposed hyper-heuristic algorithm provides better results than company's current production system. Furthermore, the flexibility of solution approach enables to run different scenarios to help the decision makers. Future research may use same data set for different solution approaches to find out the effectiveness of solution approaches.

The effectiveness of proposed approach can also be evaluated by comparing with other meta/hyper-heuristics. Simulated annealing meta-heuristic, independent from hyper-heuristic framework, is used for more fair comparisons. The learning capabilities of hyper-heuristic can be evaluated by this comparison. The performance comparison of SA based Hyper-heuristic and simulated annealing algorithm itself is presented in Table 4.

Table 3. Results for different cycle times

Cycle time (c)	Minimum station number (K)	Line balance	Number of direction changes in the line	Number of platform changes in the line	Number of combiner changes in the line	Number of connector changes in the line
114	23	0.71	0	0	8	2
108	24	0.72	0	0	8	2
105	25	0.71	0	0	5	2
98	27	0.70	0	2*	4	4
93	29	0.69	0	2*	9	3
90	25	0.82	6**	2*	8	2

* The restriction about the platform usage is relaxed.

** The restriction about direction changes is relaxed.

As seen from the Table 4, hyper-heuristic and simulated annealing solution approaches produce similar results but simulated annealing never reaches the values resulted by hyper-heuristics. When the cycle time decrease, the range between the results is widened. The hyper-heuristic approach outperformed the simulated annealing even in the lower cycle times. This presents that the learning mechanism of hyper-heuristics affects the solution more than simulated annealing in lower cycle time values. The hyper-heuristic approach outperforms the simulated annealing with respect to the specified restrictions. It is clear that hyper-heuristic meets the restrictions better than simulated annealing.

CONCLUSION

In this study, an ALB problem with product and station constraints and restrictions is addressed. Since the problem is real world application problem, the problem specific restrictions that reflect company's preferences are integrated into general single model ALB problem structure. Ten different sequencing heuristics are implemented for solving the problem by hyper-heuristic approach. While the low-level heuristics search in the solution space of the problem, hyper-heuristic searches over the proper order of low-level heuristics.

Table 4. The comparison of HH and SA solution approaches

Cycle time (c)	Minimum station number (K)		Line balance		Number of direction changes in the line		Number of platform changes in the line		Number of combiner changes in the line		Number of connector changes in the line	
	HH	SA	HH	SA	HH	SA	HH	SA	HH	SA	HH	SA
114	23	23	0.71	0.71	0	0	0	0	8	12	2	4
108	24	24	0.72	0.72	0	0	0	1	8	12	2	4
105	25	25	0.71	0.71	0	1	0	1	5	8	2	6
98	27	28	0.70	0.68	0	1	2*	5	4	9	4	4
93	29	29	0.69	0.69	0	3	2*	5	9	14	3	6
90	25	26	0.82	0.79	6**	8	2*	5	8	12	2	6

The proposed solution method obtains less station numbers than the company's current system. Also, the preferred goals are met with these station numbers. The performance of the proposed hyper-heuristic algorithm is also compared with the performance of simulated annealing algorithm. The hyper-heuristic approach significantly outperformed the simulated annealing algorithm in terms of number of stations and line balance. It also provides better solutions, which meet the problem specific restrictions such as minimum number of direction, platform, combiner, and connector changes during the assembly process. It is concluded from the results and evaluations that the proposed hyper-heuristic approach can be used effectively in the solution of the specialized ALB problems consisting product and station restrictions due to its flexible and adaptive structure.

REFERENCES

- Amen, M. (2001). Heuristic methods for cost-oriented assembly line balancing: A comparison on solution quality and computing time. *International Journal of Production Economics*, 69(3), 255–264. doi:10.1016/S0925-5273(99)00096-1
- Arcus, A. L. (1966). COMSOAL: A computer method of sequencing operations for assembly lines. *International Journal of Production Research*, 4(4), 259–277. doi:10.1080/00207546508919982
- Bard, J. F., Dar-El, E., & Shtub, A. (1992). An analytic framework for sequencing mixed model assembly lines. *International Journal of Production Research*, 30(1), 35–48. doi:10.1080/00207549208942876
- Bartholdi, J. J. (1993). Balancing two-sided assembly lines: A case study. *International Journal of Production Research*, 31(10), 2447–2461. doi:10.1080/00207549308956868
- Baybars, İ. (1986). A survey of exact algorithms for the simple assembly line balancing problems. *Management Science*, 32(8), 909–932. doi:10.1287/mnsc.32.8.909
- Baykasoğlu, A. (2006). Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems. *Journal of Intelligent Manufacturing*, 17(2), 217–232. doi:10.1007/s10845-005-6638-y
- Becker, C., & Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168(3), 694–715. doi:10.1016/j.ejor.2004.07.023
- Boysen, N., Flidner, M., & Scholl, A. (2008). Assembly line balancing: Which model to use when? *International Journal of Production Economics*, 111(2), 509–528. doi:10.1016/j.ijpe.2007.02.026
- Buckhin, J., & Rubinovitz, J. (2003). A weighted approach for assembly line design with station paralleling and equipment selection. *IIE Transactions*, 35(1), 73–85. doi:10.1080/07408170304429
- Bukchin, J., Dar-El, E. M., & Rubinovitz, J. (2002). Mixed-model assembly line design in a make-to-order environment. *Computers & Industrial Engineering*, 41(4), 405–421. doi:10.1016/S0360-8352(01)00065-1
- Bukchin, J., & Tzur, M. (2000). Design of flexible assembly line to minimize equipment cost. *IIE Transactions*, 32(7), 585–598. doi:10.1080/07408170008967418
- Burke, E., McCollum, B., Meisels, A., Petrovic, S., & Qu, R. (2007). A graph-based hyper-heuristic for educational timetabling problems. *European Journal of Operational Research*, 176(1), 177–192. doi:10.1016/j.ejor.2005.08.012
- Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., & Qu, R. (2013). Hyper-heuristics: A survey of the state of the art. *The Journal of the Operational Research Society*, 64(12), 1695–1724. doi:10.1057/jors.2013.71
- Burke, E. K., Hart, E., Kendall, G., Newall, J., Ross, P., & Schulenburg, S. (2003). Hyper-heuristics: An emerging direction in modern search technology. In F. Glover & G. Kochenberger (Ed), *Handbook of Metaheuristics* (pp. 457–474).
- Burke, E. K., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., & Woodward, J. (2010). A classification of hyper-heuristic approaches. *Handbook of Metaheuristics, International Series in Operations Research & Management Science*, 146, 449–468.
- Burns, L. D., & Daganzo, C. F. (1987). Assembly line job sequencing principles. *International Journal of Production Research*, 25(1), 71–99. doi:10.1080/00207548708919824
- Dobson, G., & Yano, C. A. (1994). Cyclic scheduling to minimize inventory in a batch flow line. *European Journal of Operational Research*, 75(2), 441–461. doi:10.1016/0377-2217(94)90087-6
- Erel, E., Sabuncuoğlu, İ., & Aksu, B. A. (2001). Balancing of U-type assembly systems using simulated annealing. *International Journal of Production Research*, 39(13), 3003–3015. doi:10.1080/00207540110051905
- Essafi, M., Delorme, X., Dolgui, A., & Guschinskaya, O. (2010). A MIP approach for balancing transfer line with complex industrial constraints. *Computers & Industrial Engineering*, 58(3), 393–400. doi:10.1016/j.cie.2009.04.009

- Ghosh, S., & Gagnon, R. J. (1989). A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. *International Journal of Production Research*, 27(4), 637–670. doi:10.1080/00207548908942574
- Gökçen, H., Ağpak, K., & Benzer, R. (2006). Balancing of parallel assembly lines. *International Journal of Production Economics*, 103(2), 600–609. doi:10.1016/j.ijpe.2005.12.001
- Gupta, M., Kundu, A., & Gupta, V. (2017). Multi-objective artificial bee colony algorithm for multi-echelon supply chain optimization problem: An Indian case study. *International Journal of Operations Research and Information Systems*, 8(4), 76–89. doi:10.4018/IJORIS.2017100105
- Karp, R. M. (1972). *Reducibility among combinatorial problems*. R. E. Miller, J. W. Thatcher, *Complexity of Computer Computation* (pp. 85–103). New York: Plenum Press.
- Lapierre, S. D., & Ruiz, A. B. (2004). Balancing assembly lines: An industrial case study. *The Journal of the Operational Research Society*, 55(6), 589–597. doi:10.1057/palgrave.jors.2601708
- Layeb, A. (2015). Two Novel Heuristics Based on a New Density Measure for Vehicle Routing Problem. *International Journal of Operations Research and Information Systems*, 6(1), 78–90. doi:10.4018/ijoris.2015010106
- McMullen, P., & Frazier, G. (1998). Using simulated annealing to solve a multi objective assembly line balancing problem with parallel workstations. *International Journal of Production Research*, 36(10), 2717–2741. doi:10.1080/002075498192454
- McMullen, P., & Tarasewich, P. (2003). Using ant techniques to solve the assembly line balancing problem. *IIE Transactions*, 35(7), 605–617. doi:10.1080/07408170304354
- Miltenburg, J., & Wijngaard, J. (1994). The U-line line balancing problem. *Management Science*, 40(10), 1378–1388. doi:10.1287/mnsc.40.10.1378
- Özbakır, L., Baykasoğlu, A., Görkemli, B., & Görkemli, L. (2011). Multiple-colony ant algorithm for parallel assembly line balancing problem. *Applied Soft Computing*, 11(3), 3186–3198. doi:10.1016/j.asoc.2010.12.021
- Pastor, R., & Corominas, A. (1999). Assembly line balancing with work zone and station load time windows constraints. In *Proceedings of the 1999 IEEE International Symposium on Assembly and Task Planning, ISATP 99*.
- Sabuncuoğlu, I., Erel, E., & Alp, A. (2009). Ant colony optimization for the single model U-type assembly line balancing problem. *International Journal of Production Economics*, 120(2), 287–300. doi:10.1016/j.ijpe.2008.11.017
- Scholl, A., & Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168(3), 666–693. doi:10.1016/j.ejor.2004.07.022
- Scholl, A., Flidner, M., & Boysen, N. (2010). Balancing assembly lines with assignment restrictions. *European Journal of Operational Research*, 200(3), 688–701. doi:10.1016/j.ejor.2009.01.049
- Seçme, G. (2015). Hyper-heuristic approach to stochastic parallel assembly line balancing problem with machine/equipment costs [Ph.D. Thesis]. Erciyes University, Institute of Social Sciences.
- Sensuse, D. I., & Cahyaningsih, E. (2018). Knowledge management models: A summative review. *International Journal of Information Systems in the Service Sector*, 10(1), 71–100. doi:10.4018/IJISSS.2018010105
- Suresh, G., & Sahu, S. (1994). Stochastic assembly line balancing using simulated annealing. *International Journal of Production Research*, 32(8), 1801–1810. doi:10.1080/00207549408957042
- Yano, C. A., & Bolat, A. (1989). Survey, development and application of algorithms for sequencing paced assembly lines. *Journal of Manufacturing and Operations Management*, 2, 172–198.

Gökhan Seçme is an Assistant Professor of Operations Management in the Faculty of Business Administration and Economics, University of Nevşehir Hacı Bektaş Veli, Turkey. He has a B.S. in Industrial Engineering from Erciyes University (Turkey), an M.S. in Business Administration and Ph.D. in Production Management and Marketing from Erciyes University. His research interests include Operations Management, Meta-heuristics, Hyper-heuristics, and optimization with AI. He has papers and conference proceedings in business administration and industrial engineering related areas.

Lale Özbakır received her B.Sc. degree in Computer Engineering from Bilkent University in 1992. She got her Ph.D. degree from Erciyes University, Production Management programme in 2004. Her research focuses on flexible scheduling, optimization, evolutionary computation, collective intelligence, neural networks and data mining. She has a particular interest in genetic programming and the interface of operational research and computer science. She is a member of IODM: Intelligent Optimization and Decision Making Research Group. She has co-authored more than 40 articles in scientific journals and several conference papers.