



Mixed-model assembly line sequencing using beam search

Erdal Erel , Yasin Gocgun & İhsan Sabuncuoğlu

To cite this article: Erdal Erel , Yasin Gocgun & İhsan Sabuncuoğlu (2007) Mixed-model assembly line sequencing using beam search, International Journal of Production Research, 45:22, 5265-5284, DOI: [10.1080/00207540600806497](https://doi.org/10.1080/00207540600806497)

To link to this article: <http://dx.doi.org/10.1080/00207540600806497>



Published online: 03 Oct 2007.



Submit your article to this journal [↗](#)



Article views: 120



View related articles [↗](#)



Citing articles: 24 View citing articles [↗](#)

Mixed-model assembly line sequencing using beam search

ERDAL EREL*†, YASIN GOÇGUN‡ and
İHSAN SABUNCUOĞLU†

†Bilkent University, Turkey

‡University of Washington, USA

(Revision received 29 January 2006)

In today's manufacturing environments, companies have to produce a large variety of products in small quantities on a single assembly line. In this paper, we use a beam search (BS) approach to solve the model-sequencing problem of mixed-model assembly lines (MMALs). Specifically, we develop six BS algorithms for part-usage variation and load-leveling performance measures. The results of computational experiments indicate that the proposed BS methods are competitive with the well-known heuristics in the literature. A comprehensive bibliography is also provided.

Keywords: Mixed-model assembly line sequencing; Beam search; Heuristic

1. Introduction

In today's competitive business environment, many industries are facing a diversification of customer demand that requires an increasing variety of products. Hence, many companies have to produce a large variety of products on a single assembly line. These lines that are capable of producing different models simultaneously are called mixed-model assembly lines (MMALs). These systems assemble a variety of products in small quantities with no significant changeover delay or setup costs. Thus, they can respond quickly to changes in market demand and avoid large inventories.

MMALs are generally used in multi-level production systems in which raw materials are manufactured into components that are then combined into subassemblies. These subassemblies are assembled into products on a final assembly line. These systems are considered an integral part of just-in-time (JIT) production systems where the subassemblies, components and raw materials are pulled forward as they are needed, and production is initiated by one level's requirement for the output of another level. Hence, it is sufficient to focus on the final assembly line to control the mixed-model lines (Miltenburg and Sinnamon 1992).

In this paper, we study the MMAL sequencing problem defined as determining the sequence of the products on the assembly line to optimize some selected performance measures. We consider two measures: *Leveling Parts Usage* to maintain

*Corresponding author. Email: erel@bilkent.edu.tr

a constant usage rate of all the parts that are fed into the assembly line; and *Leveling Workload* to smooth the workload on the assembly line to reduce production delays and stoppages. These measures are commonly used in practice, especially for JIT production systems.

The first measure is equivalent to minimizing variability of part usage. Similar to Monden (1983), we study variability at the subassembly level. Variability is typically calculated as the deviation between actual usage and desired usage at the subassembly level.

The size of the search space for this problem is too large for optimum-seeking algorithms; only small sizes of the problem can be solved by exact methods (e.g. branch-and-bound). Hence, heuristics are generally recommended for problems of realistic sizes.

In this paper, we propose a beam search (BS) approach to solve this problem. Specifically, we develop six BS-based algorithms and compare their performance against well-known heuristics in the literature. Our computational results indicate that the proposed algorithms are generally competitive to the existing heuristics for the benchmark problems reported in the literature. In general, BS is used as a constructive heuristic to solve various combinatorial optimization problems, ranging from scheduling to marketing and data mining. In most of these applications, it has been reported as a very effective heuristic in terms of both solution quality and CPU times. This is partly due to the fact that BS explores search space in an intelligent manner. Our previous experience with BS for the U-type assembly line balancing problem (Erel *et al.* 2005) has also encouraged us to apply the BS-based approach to the assembly sequencing problem.

The rest of the paper is as follows: in section 2, we present the related work in this area. In section 3, we give the formulation of the problem, and discuss the heuristics developed for the problem. In section 4, we explain the structure of the proposed BS algorithms. We present the computational results in section 5. Finally, we give concluding remarks and further research directions in section 6.

2. Literature review

Following the seminal work of Kilbridge and Webster (1963), several researchers study the MMAL sequencing problem (Thomopoulos 1967, Dar-El and Cother 1975, Dar-El and Cucuy 1977, Yamashita and Okamura 1979). These studies have different objectives such as minimizing line length or line stoppages, but their common feature is that they all study the final assembly line and ignore other levels in the multi-level production systems. The first analyses of mixed-model, multi-level production systems are made by Monden (1983), and Miltenburg and Sinnamon (1989). A detailed explanation of research on the MMAL sequencing problem is given below (also summarized in table 1).

Miltenburg (1989) studies the mixed-model sequencing problem by considering the variation in production rates of the finished products. The author develops an exact algorithm to solve the problem, which has a worst-case complexity that grows exponentially with the number of products. He also develops two heuristics for the problem. In another study, Miltenburg and Sinnamon (1989) consider multi-level model production systems to solve the same problem with the objective of keeping

Table 1. Summary of research on the MMAL sequencing problem.

Author	Type of production system	Objective	Solution method
Thomopoulos (1967)	Single level	Optimally utilizing line operators	Heuristic
Dar-El and Cother (1975)	Single level	Minimizing line length	Heuristic
Dar-El and Cucuy (1977)	Single level	Minimizing line length	Integer programming
Dar-El (1978)	Single level	Minimizing line length, and throughput time	Heuristic
Yamashita and Okamura (1979)	Single level	Minimizing line stoppages	Heuristic
Monden (1983)	Multi-level	Leveling part usage	Heuristic
Miltenburg (1989)	Single level	Leveling product usage	Heuristic
Miltenburg and Sinnamon (1989)	Multi-level	Leveling product and part usage	Heuristic
Sumichrast and Russell (1990)	Multi-level	Leveling part usage	None
Kubiak and Sethi (1991)	Single level	Leveling product usage	Optimization algorithm
Miltenburg and Goldstein (1991)	Multi-level	Leveling product and part usage	Heuristic
Inman and Bulfin (1991)	Single level	Leveling product usage	EDD algorithm
Yano and Rachamadugu (1991)	Single level	Minimizing work overload	Heuristic
Miltenburg and Sinnamon (1992)	Multi-level	Leveling product and part usage	Heuristic
Kubiak (1993)	Multi-level	Leveling part usage	None
Ding and Cheng (1993)	Single level	Leveling product usage	Heuristic
Ng and Mak (1994)	Single level	Leveling product usage	Branch-and-Bound algorithm
Bard <i>et al.</i> (1994)	Single level	Leveling product usage, and minimizing line length	Branch-and-Bound algorithm and heuristic
Bautista <i>et al.</i> (1996)	Multi-level	Leveling part usage	Exact algorithm and heuristic
Duplaga <i>et al.</i> (1996)	Multi-level	Leveling part usage	None
Kim <i>et al.</i> (1996)	Single level	Minimizing line length	Heuristic
Duplaga and Bragg (1998)	Multi-level	Leveling part usage	None
Hyun <i>et al.</i> (1998)	Single level	Leveling product usage, minimizing total utility work, and setup	Heuristic
Merengo <i>et al.</i> (1999)	Single level	Leveling product usage	Heuristic
Sumichrast <i>et al.</i> (2000)	Multi-level	Leveling part usage	Heuristic

(continued)

Table 1. Continued.

Author	Type of Production System	Objective	Solution Method
Zhu and Ding (2000)	Multi-level	Leveling part usage	Heuristic
McMullen and Frazier (2000)	Single level	Leveling product usage, and minimizing setups	Heuristic
Zeramini <i>et al.</i> (2000)	Multi-level	Leveling part usage and workload	Heuristic
Drexel and Kimms (2001)	Multi-level	Leveling part usage and workload	Integer programming
Korkmazel and Meral (2001)	Single level	Leveling product usage and workload	Heuristic
Ponnambalam <i>et al.</i> (2003)	Multi-level	Leveling part usage, minimizing total utility work, and minimizing total setup cost	Heuristic
McMullen and Tarasewich (2005)	Single level	Leveling product usage, and minimizing the number of setups	Heuristic
Mansouri (2005)	Single level	Leveling product usage, and minimizing setups	Heuristic
Celano <i>et al.</i> (2004)	Multi-level	Leveling part usage	Heuristic

a constant rate of supply of every part used by the system. They develop a mathematical model for the problem, and extend the heuristics proposed by Miltenburg (1989) to include all levels in the multi-level system. In a follow-up study, Miltenburg and Sinnamon (1992) consider the same problem and propose heuristic procedures to solve large-sized problems.

Kubiak and Sethi (1991) study the MMAL sequencing problem with the objective of minimizing product-usage variation. They develop an assignment model to generate optimal schedules for MMALs. In another study, Kubiak (1993) reviews the research conducted on this problem. In this paper, he also discusses the research efforts related to both product-usage rate and component-usage rate variation using various objective functions such as maximum and total deviation between actual usage and the expected usage.

In another study, Zhu and Ding (2000) develop a transformed two-stage method for the part-level problem. The single-stage (two-stage) heuristic minimizes the one-stage (two-stage) variation each time a model is added to the sequence. Their computational results indicate that the proposed method generally produces better solutions than the one-stage method. In a recent study, Celano *et al.* (2004) study the MMAL sequencing problem with the objective of smoothing parts-usage. They study not only the traditional goal chasing (GC) approaches, but also models which take into account the effective length of the assembly line. The authors propose a simulated annealing (SA) algorithm for this problem and compare it with GC algorithms. The experimental results indicate that in most cases, SA outperforms other heuristics.

There is also a large number of studies in the literature that considers the MMAL sequencing problem with multiple objectives. Hyun *et al.* (1998) propose a genetic algorithm (GA) to solve the problem with three objectives: minimizing total utility work, minimizing total setup cost, and keeping model production constant. In another study, McMullen and Frazier (2000) propose a SA-based heuristic that simultaneously considers both setups and the leveling product usage rates to solve MMAL sequencing problem. The performance of the SA algorithm is compared with that of the tabu search (TS) approach. The results indicate that SA generally outperforms TS.

McMullen and Tarasewich (2005) consider the problem of mixed-model sequencing with setups. The problem has two objectives: minimizing the product usage variation and number of setups. The authors develop a BS heuristic to generate efficient frontiers. Their experimental results indicate that the proposed approach performs well in terms of both solution quality and computation times. In a recent study, Mansouri (2005) develops a multi-objective GA for the MMAL sequencing problem to simultaneously optimize the variation of product usage rates and number of setups. The results show that the proposed method is better than other methods in terms of both CPU time and solution quality.

In summary, substantial research has been conducted on the MMAL sequencing problem to minimize the parts usage variation and the workload variation with more emphasis on the former problem due to its importance for JIT systems. Although small-size problems can be handled by exact procedures (branch-and-bound and dynamic programming), heuristic methods are generally recommended to solve large-size problems. These heuristics range from myopic greedy methods to other custom-made constructive heuristics and meta-heuristics. To the

best of our knowledge, there is not a comprehensive study that compares the existing solution procedures in the literature.

Among the available comparative studies, Leu *et al.* (1997) test the relative performance of the BS methods against the goal chasing (GC) and the 2-step methods. Their results indicate that the BS method outperforms both the GC and the 2-step methods. They also show that the BS method offers a substantial improvement over the 2-step method when the problem size increases. In another study, Duplaga and Bragg (1998) compare the performance of six sequencing heuristics: two versions of GC, Hyundai's heuristic, two heuristics of Miltenburg and Sinnamon, and the extended GC method. The results of the computational experiments indicate that the extended GC and Miltenburg and Sinnamon's second heuristic have significantly better performance than the others. Finally, in a recent study, Jin and Wu (2002) show that the variance method is superior to the GC and the 2-step methods. However, this study lacks a comparison of these methods against the BS method.

In this context, our study will be the first study that compares several state-of-the-art heuristics in the literature together with the proposed BS algorithms.

3. Problem formulation and existing heuristics

In this section, we first give the formulation of the problem for the part-usage and load-leveling measures and then discuss the solution methods proposed in the literature.

3.1 Problem formulation

3.1.1 Part-usage measure. The formulation of the MMAL sequencing problem is given by Jin and Wu (2002) as follows:

There are N different models to be produced on the final assembly line, and C different parts that can be used by a model. The following notation is used in formulating the problem.

- d_i : demand for model i , $i = 1, \dots, N$
- $c_{j,i}$: number of part j required for model i , $i = 1, \dots, N$, $j = 1, \dots, C$
- D_T : total demand for all models, $D_T = \sum_{i=1}^N d_i$
- T_j : total number of part j required for the full sequence, $j = 1, \dots, C$,

$$T_j = \sum_{i=1}^N c_{j,i} d_i$$

- r_j : desired consumption rate of part j , $r_j = T_j/D_T$, $j = 1, \dots, C$
- $x_{i,k}$: number of times model i is sequenced in the first k position for a specific sequence

The desired number of part j consumed in the first k positions for a specific sequence is kr_j , and the cumulative consumption of part j for one specific sequence

at position k is $\sum_{i=1}^N x_{i,k}c_{j,i}$. Hence, the part-usage variation at any level (i.e. level k) is calculated as follows:

$$V = \sum_{j=1}^C \left(\sum_{i=1}^N x_{i,k}c_{j,i} - kr_j \right)^2 \tag{1}$$

Using the above notation, the MMAL sequencing problem is formulated as follows:

$$\text{Min. SDQ} = \sum_{k=1}^{D_T} \sum_{j=1}^C \left(\sum_{i=1}^N x_{i,k}c_{j,i} - kr_j \right)^2 \tag{2}$$

$$\text{s.t. } \sum_{i=1}^N x_{i,k} = k \quad k = 1, \dots, D_T \tag{3}$$

$$x_{i,k} - x_{i,k-1} \leq 1 \quad i = 1, \dots, N, k = 1, \dots, D_T \tag{4}$$

$$x_{i,k} - x_{i,k-1} \geq 0 \quad i = 1, \dots, N, k = 1, \dots, D_T \tag{5}$$

$$0 \leq x_{i,k} \leq d_i \quad i = 1, \dots, N, k = 1, \dots, D_T \tag{6}$$

where $x_{i,k}$ is a non-negative integer

The objective function minimizes the cumulative variation in parts consumption. Constraint (3) ensures that at any stage k , the total number of sequenced models is k . Constraints (4) and (5) require that, at any stage, the number of the sequenced model i should increase by one or remain the same. Constraint (6) guarantees that the number of the sequenced model i at any position k should not exceed the demand for this model. This problem is an integer nonlinear problem and it is *NP-Hard* in any sense, even if the objective is linearized (Jin and Wu 2002).

3.1.2 Load-leveling objective. The mathematical formulation of the load-leveling problem is developed by Miltenburg and Goldstein (1991) with the following assumptions: (1) there are N models to be assembled on the final assembly line, and S stations where different models usually require significantly different operation times; (2) the available production time at each station (cycle time) on the assembly line is fixed.

The variables d_i , D_T , and $x_{i,k}$, used to formulate the part-usage problem, also hold true for the load-leveling problem. The additional notation is presented below.

T_i^s : production time required to produce model i at station $s, s = 1, \dots, S$

\bar{T}^s : average production time required at station $s, s = 1, \dots, S$

$$\bar{T}^s = \frac{\sum_{i=1}^N T_i^s d_i}{D_T}$$

The total actual time at station S for model i sequenced up to level k is $T_i^s x_{i,k}$, and the total actual time at station S for all models sequenced up to level k is: $\sum_{i=1}^N T_i^s x_{i,k}$. Moreover, the desired production time over the first k positions is $k\bar{T}^s$.

Using these expressions, the loading variation at position k is calculated as:

$$LV = \sum_{s=1}^S \left(\sum_{i=1}^N T_i^s x_{i,k} - k \bar{T}^s \right)^2 \quad (7)$$

Since the objective function is the sum of the loading variation at each position, equation (7) is summed over all positions to express the objective function. The formulation for the load-leveling problem is similar to that for the part-usage problem; however, it emphasizes the workloads at stations instead of subassemblies and parts usage (see Miltenburg and Goldstein (1991) for a detailed formulation of the problem).

3.2 Existing solution methods

Several solution procedures have been proposed for the MMAL sequencing problem with the objective of minimizing parts usage.

3.2.1 Goal chasing method. Monden (1983) develops a greedy heuristic, the Goal Chasing Method (GCM), to level parts usage by selecting the model that yields the minimum part-usage variation at any level. Hence, it is a myopic heuristic that generally yields low quality solutions. The steps of the algorithm are shown below (Jin and Wu 2002):

Step 1: Set $k = 1$, $x_{i,0} = 0$, $i \in \{1, \dots, N\}$

Step 2: Select model m with $x_{m,k-1} < d_m$ that minimizes the variation at position k :

$$m^* = \arg \min_{m \in \{1, \dots, N\}} \left\{ \sum_{j=1}^C \left(\sum_{i=1}^N ((x_{i,k-1} + z_{i,m}) c_{j,i}) - kr_j \right)^2 \right\} \quad (8)$$

$$\text{where } z_{i,m} = \begin{cases} 1 & \text{if } i = m \\ 0 & \text{otherwise} \end{cases} \quad m \in \{1, \dots, N\}$$

Step 3:

$$x_{i,k} = \begin{cases} x_{i,k-1} + 1 & \text{when } i = m^* \\ x_{i,k-1} & \text{otherwise} \end{cases} \quad i \in \{1, \dots, N\}$$

Step 4: Set $k = k + 1$
If $k > D_T$ stop

Step 5: Go to Step 2

As seen from Step 2, at any level the GCM calculates the variation by positioning each of the additional models into the next level, and then selects the model with the smallest variation.

3.2.2 2-step method. Bautista *et al.* (1996) propose a two-stage heuristic to reduce the myopic feature of the GCM. The procedure searches two models for the next two stages to minimize the combined variation (i.e. total variation at two positions) by

considering all combinations. The combination with the minimum combined variation is chosen and only the first model is placed in the sequence. Note that the same methodology is also developed by Miltenburg and Sinnamon (1989) for the multi-level production system.

3.2.3 Variance method. Jin and Wu (2002) develop a heuristic method to improve the performance of the GCM. The drawback of the GCM is that the good units (i.e. models) are inserted in the early iterations, leaving the bad units for later iterations. The authors propose a *variance* improvement procedure to alleviate this problem.

A good unit is defined as a model with a parts structure that is close to the desired consuming rate. In order to measure goodness, they use *model variance* v_i :

$$v_i = \sum_{j=1}^C (r_j - c_{j,i})^2 \tag{9}$$

Hence, a model with small v_i is a good unit. For one specific composition of the units, they also define the total *composition variance* as:

$$t = \sum_{i=1}^N d_i v_i \tag{10}$$

A model with a small composition variance value contains many good units and probably yields a good sequence for the usage problem.

The variance improvement procedure reduces the myopic feature of the GCM by integrating the composition variance as the opportunity cost for the remaining composition into the total cost. Hence, the current cost (i.e. variation at the current position) and the opportunity cost move in opposite directions. The opportunity cost is multiplied by a discounting coefficient and the model with the minimum total cost is selected at each stage. The variance method is implemented by changing Step 2 in the GCM as follows:

Step 2: Choose model m with $x_{m,k-1} < d_m$ that minimizes the total cost:

$$m^* = \arg \min_{m \in \{1, \dots, N\}} \left\{ \begin{aligned} & \sum_{j=1}^C \left(\sum_{i=1}^N ((x_{i,k-1} + z_{i,m})c_{j,i}) - kr_j \right)^2 + \\ & w \sum_{i=1}^N \left((d_i - x_{i,k-1} - z_{i,m}) \sum_{j=1}^C (r_j - c_{j,i}) \right)^2 \end{aligned} \right\} \tag{11}$$

$$\text{where } z_{i,m} = \begin{cases} 1 & \text{if } i = m \\ 0 & \text{otherwise} \end{cases} \quad m \in \{1, \dots, N\},$$

where w is the discounting factor for the opportunity cost.

The opportunity cost added to the usage variation in equation (11) improves the solution quality of GCM. Furthermore, it does not increase the CPU time requirements dramatically since the procedure is still greedy by nature.

3.2.4 2-step variance method. Similar to the 2-step heuristic, the 2-step variance method positions two models for the next two stages and compares all alternatives

with respect to the combined total variation. The combination with the minimum total variation is chosen and only the first model is positioned into the sequence. In this way, the procedure further enhances the look-ahead feature of the variance method.

3.2.5 BS method. Leu *et al.* (1997) develop a BS technique for the problem to minimize the parts-usage variation. At each stage, the procedure selects a certain number of nodes called *beam width nodes* using an evaluation function that minimizes the variation in parts consumption. The evaluation function evaluates all solution paths at any given level by calculating the cumulative parts-usage variation at the current stage. It then selects the beam width nodes that yield the minimum cumulative variation. The search terminates when all the models are allocated. The sequence is decoded by tracing back up the best solution path.

3.2.6 Performance of the heuristics. The relative performances of the existing heuristics are tested in a few studies. Leu *et al.* (1997) test the performance of the BS method against GCM and the 2-step method. Their computational results show that the BS method outperforms both GCM and the 2-step method in solution quality. The BS method also offers a substantial improvement over the 2-step method when problem size increases. In another study, Jin and Wu (2002) demonstrate that the variance method is superior to the GCM in solution quality. They also show that the variance method is faster than the 2-step method.

4. Proposed method

BS is an adaptation of the branch-and-bound method in which a limited number of solution paths (i.e. *beam width* number of paths) are explored in parallel. In this methodology, only the best β promising nodes, called the beam width number of nodes, are kept for further branching at any level (Sabuncuoglu and Bayiz 1999). The potential promise of each node is determined by a global evaluation function, which typically estimates the minimum total cost of the best solution obtained from the partial schedule represented by the node. For each beam node, one node (the beam node for the next level) is chosen among its descendants using the global evaluation function. Hence, the search progresses through β independent beams. BS can also invoke a filtering procedure during which some nodes are eliminated by a computationally fast method (i.e. local evaluation function), and only the remaining nodes (*filter width*) are considered by the global evaluation function. The previous applications of the BS to various combinatorial optimization problems indicate that it typically generates high quality solutions with reasonable computational times (Sabuncuoglu and Kizilisik 2003, Erel *et al.* 2005). The success is partly due to its ability to use the optimizing properties of the branch-and-bound methodology with limited number beams. However, when the number of beams (i.e. beam width) gets large, the computational requirements of BS increases significantly.

Figure 1 shows an example of a BS tree in which beam width and filter width are both equal to 2. The dashed circles represent the nodes which are eliminated during local evaluation. The solid circles are the nodes that are globally evaluated.

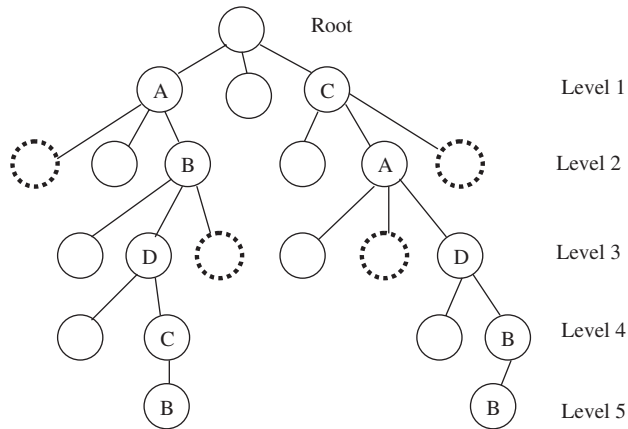


Figure 1. An illustration of BS tree.

The beam width number of promising nodes (beam nodes) is chosen using local and global evaluations and the search proceeds through these selected nodes. Next, each beam node is further explored independently to generate a partial tree and the search progresses through these parallel beams. At the end of the search tree, the superior beam is chosen. In figure 1, the BS procedure results in the product sequences of A-B-D-C-B and C-A-D-B-B. The sequence that yields the minimum cumulative usage variation is selected as the final solution.

The proposed algorithm is based on BS with each node corresponding to a solution state and represents the partial sequence of products. The value of the local evaluation function is the parts-usage variation (see equation 1). The global evaluation function is defined as the total parts-usage variation, which is the sum of the parts usage variation at the current level (i.e. one level ahead of the beam node) and subsequent levels.

The procedure outlined above can also be called BS with independent beams (i.e. beams proceed independently from each other). A BS-based algorithm can also be implemented with dependent beams, i.e. all descendant nodes are evaluated at any level and the best β nodes are chosen among them as the beam nodes. Since all the nodes at a given level are considered in the latter case, the information accumulated during the search process is made available to all the nodes, leading to possibly better quality solutions.

The proposed BS-based approach includes various enhancement tools such as backtracking and information exchange (i.e. sharing). *Backtracking* is used to revisit previous solution states in the search tree with the expectation of obtaining better solutions. The motivation of this procedure stems from the fact that whenever two or more beams are equivalent in some sense, some of the beams are further explored by returning to their previous solution states so that available computational resources are utilized on more promising nodes. The other enhancement tool is the exchange of information (EOI) by which we take a part of a solution from one beam and transfer it to another beam, hoping that the resulting beam with the additional information will lead to a better solution. This is especially beneficial for independent beams to reduce computational times.

4.1 Different versions of the proposed method

The proposed method can be implemented in different ways, leading to the following versions:

- BS-1: The standard BS in which beams progress independently.
- BS-2: A BS in which beams progress independently with a backtracking procedure.
- BS-3: A BS in which beams progress independently with an EOI procedure.
- BS-4: A BS in which beams progress independently with both backtracking and EOI procedures.
- BS-5: A BS with dependent beams.
- BS-6: A BS in which beams progress dependently and an EOI procedure is invoked.

The purpose of having these six versions is to measure the effect of the enhancements tools on the solution quality.

5. Computational results

In this section, we compare two versions of the proposed method (BS-4 and BS-6) with the five well-known heuristics in the literature (GCM, 2-step method, variance method, 2-step variance method, and BS algorithm). In the implementation process, we also fine-tune the parameters of these methods: the discounting coefficient of the opportunity cost for variance and 2-step variance methods; beam width and filter width for BS methods. When we examine the performance of the methods with independently progressing beams (BS-1, BS-2, BS-3, and BS-4), we observe that BS-1 performs poorly since it has no enhancement tools. The performance of BS-2 and BS-3 is quite close to each other implying that the effects of backtracking and EOI procedures are comparable. However, BS-4 performs better than the other versions since it includes both of the enhancement tools. The difference between the performance of BS-5 and BS-6 is not significant; this implies that the EOI procedure is not an effective tool to improve solution quality of BS with dependent beams. BS-6 performs slightly better than BS-5. Hence, we only consider BS-4 and BS-6 in the comparisons. The complete set of experiments and comparisons can be found in Gocgun (2005).

5.1 Computational results for the parts usage measure

The heuristics are first tested with the problem data set given in Bautista *et al.* (1996) and Jin and Wu (2002). The results are presented with 95% confidence level for different structures (part-product combinations) and demand patterns. In the data set, there are six structures and 45 demand patterns for each structure. The value of the objective function (Z_{avg}) is the average of the variations obtained for demand patterns.

We also generate additional problem instances to further test the performance of the proposed algorithms. Specifically, we generate problem instances using the factors given in Leu *et al.* (1997). These are: (1) number of products; (2) quantity per assembly; and (3) degree of commonality. Additionally, for each one of these eight

configurations, we use nine different demand patterns as suggested by Ding and Cheng (1993); thus, generating 10 random problems for each one of these 72 combinations results in 720 problem instances.

As discussed in Leu *et al.* (1997), the number of products factor determines the size of the search space. We consider low and high levels as 5 and 20, respectively. The other factors define the degree of product complexity (DOC) and the quantity per assembly (QPA), which indicates the number of units of a part required for a product. It is generated from a discrete uniform distribution with the parameters given in table 2 (i.e. the parameter range [1, 10] is used for the low level and [1, 20] is used for the high level). The number of parts is twice the number of products. Degree of commonality shows the approximate percentage of common parts used by the products (e.g. 80% commonality means that a part is used by approximately 80% of the products). The demand patterns of Ding and Cheng (1993) are presented in table 2, the percentage range 0–20% is used for the low level and the percentage range 60–80% is used for the high level).

The results are presented in tables 4 and 5. Table 4 contains the results on the benchmark problems in the literature, and table 5 presents the results of the additional problem instances generated in our study. In general, BS-6 is better than the best known heuristics (2-step/var and BS(Leu)) in the literature. It yields better performance than these two heuristics for all structures (worse only in one structure against the variance method). As seen in table 4, BS-6 also outperforms all the heuristics reported in the literature for all structures except structures 3 and 5 (the *p*-values are 0.478 and 0.203, respectively). In the new dataset, BS-6

Table 2. Experimental factors and their levels.

Factors	Levels
Number of products	5 20
Quantity per assembly, QPA	[1, 10] [1, 20]
Degree of commonality, DOC	0–20% 60–80%

Table 3. Demand patterns.

Demand Pattern	<i>N</i>	
	5	20
1	16,1,1,1,1	21,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
2	15,2,1,1,1	18,4,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
3	13,4,1,1,1	15,5,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
4	10,5,2,2,1	12,6,4,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
5	8,7,2,2,1	9,6,5,3,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
6	6,6,5,2,1	6,6,5,4,3,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
7	5,5,5,3,2	4,4,4,4,4,3,3,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1
8	5,4,4,4,3	4,4,4,4,4,3,2,2,2,1,1,1,1,1,1,1,1,1,1,1,1,1
9	4,4,4,4,4	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2

Table 4. Computational results obtained by the data sets given in the literature.

Structure	N	D	# of demand patterns	β	Heuristic	Z_{avg}	p -value
1	4	20	45	–	GC	93.333	0.003
				–	2-step	64.711	
				–	Variance	66.116	
				–	2-step/var.	61.973	
				4	BS(Leu)	72.124	
				4	BS-4	60.818	
4	BS-6	60.124					
2	4	20	45	–	GC	214.185	0.0005
				–	2-step	151.856	
				–	Variance	141.749	
				–	2-step/var.	138.376	
				4	BS(Leu)	157.183	
				4	BS-4	134.849	
4	BS-6	133.529					
3	4	20	45	–	GC	210.744	0.478
				–	2-step	151.798	
				–	Variance	164.073	
				–	2-step/var.	137.984	
				4	BS(Leu)	160.082	
				4	BS-4	137.598	
4	BS-6	137.309					
4	4	20	45	–	GC	17.897	0.032
				–	2-step	15.817	
				–	Variance	16.688	
				–	2-step/var.	15.732	
				4	BS(Leu)	15.834	
				4	BS-4	15.714	
4	BS-6	15.652					
5	4	20	45	–	GC	195.734	0.203
				–	2-step	176.749	
				–	Variance	153.421	
				–	2-step/var.	157.505	
				4	BS(Leu)	185.818	
				4	BS-4	154.827	
4	BS-6	155.367					
6.1	5	20	45	–	GC	61.073	0.0001
				–	2-step	51.718	
				–	Variance	50.753	
				–	2-step/var.	48.216	
				4	BS(Leu)	51.473	
				4	BS-4	47.287	
4	BS-6	46.402					
6.2	5	48	1	–	GC	226.875	–
				–	2-step	153.040	
				–	Variance	146.040	
				–	2-step/var.	138.040	
				5	BS(Leu)	136.042	
				5	BS-4	130.208	
5	BS-6	125.708					

(continued)

Table 4. Continued.

Structure	<i>N</i>	<i>D</i>	# of demand patterns	β	Heuristic	Z_{avg}	<i>p</i> -value
6.3	5	280	1	–	GC	1344.089	–
				–	2-step	896.590	
				–	Variance	565.375	
				–	2-step/var.	574.018	
				5	BS(Leu)	1057.161	
				5	BS-4	560.660	
5	BS-6	549.446					

Table 5. Computational results obtained by the newly generated data sets.

Structure	<i>N</i>	<i>D</i>	<i>QPA</i>	<i>DOC</i>	β	Heuristic	Z_{avg}	<i>p</i> -value
1	5	20	1–10	0–20%	–	GC	1088.9	0.002
					–	2-step	1054.4	
					–	Variance	1066.4	
					–	2-step/var.	1043.8	
					3	BS(Leu)	1043.5	
					3	BS-4	1038.4	
3	BS-6	1032.4						
2	5	20	1–10	60–80%	–	GC	1387.1	0.001
					–	2-step	1299.4	
					–	Variance	1328.7	
					–	2-step/var.	1290.8	
					3	BS(Leu)	1287.4	
					3	BS-4	1279.5	
3	BS-6	1273.2						
3	5	20	1–20	0–20%	–	GC	4018.6	0.003
					–	2-step	3881.8	
					–	Variance	3917.6	
					–	2-step/var.	3840	
					3	BS(Leu)	3848.7	
					3	BS-4	3822.7	
3	BS-6	3796.5						
4	5	20	1–20	60–80%	–	GC	4895.7	<0.0001
					–	2-step	4660.7	
					–	Variance	4742.6	
					–	2-step/var.	4627.7	
					3	BS(Leu)	4653.2	
					3	BS-4	4584.7	
3	BS-6	4564.2						
5	20	40	1–10	0–20%	–	GC	11872.6	<0.0001
					–	2-step	11582.2	
					–	Variance	11218.8	
					–	2-step/var.	11093.5	
					10	BS(Leu)	11283.4	
					10	BS-4	10899.2	
10	BS-6	10983.7						

(continued)

Table 5. Continued.

Structure	N	D	QPA	DOC	β	Heuristic	Z_{avg}	p -value	
6	20	40	1–10	60–80%	–	GC	31634.9	0.002	
						–	2-step		30190.7
						–	Variance		30438.2
						–	2-step/var.		29329.6
						10	BS(Leu)		29009.1
						10	BS-4		28734.8
10	BS-6	28389.1							
7	20	40	1–20	0–20%	–	GC	39688.8	0.001	
						–	2-step		38550.4
						–	Variance		37363.8
						–	2-step/var.		36902.4
						10	BS(Leu)		37741.7
						10	BS-4		36443.9
10	BS-6	36564.2							
8	20	40	1–20	60–80%	–	GC	119856.9	0.003	
						–	2-step		114541.3
						–	Variance		115109.0
						–	2-step/var.		112033
						10	BS(Leu)		110648.2
						10	BS-4		109158.3
10	BS-6	108688.6							

also performs better than all the five heuristics for all the structures except structures 5 and 7 where BS-4 is the best in these cases (see table 5). Among the existing heuristics, the 2-step variance method and Leu's (1997) BS seem to be the most competitive methods. Note that in tables 4 and 5, the paired t test is also used for comparison of the best proposed method with the best known method in the literature to show the statistical significance.

In terms of computational times, all of these heuristics generate solutions on the order of seconds. The largest instance solved by BS-6 takes about less than 200 s on a moderate PC. Since the computational requirements are not a practical issue, we only report the results in terms of the solution quality.

5.2 Computational results for the load-leveling problem

The objective of the load-leveling problem is to minimize the variation in workloads assigned to each station over all model sequences. Note that this objective is structurally similar to the part-usage variation objective; in fact, a slight modification in the formulation of the former problem can convert it to the formulation for the latter problem. Specifically, the terms in objective function are different in these formulations.

Since there are no benchmark problems reported in the literature for this case, we generate the problem dataset by using the procedure outlined in the previous section. Note that degree of commonality is not relevant in this problem.

The list of other factors and their levels are given in table 6. The demand patterns are again taken from Ding and Cheng (1993) (see table 7). We generate

Table 6. Experimental factors and their levels.

Factors	Levels
Number of products	5 10
Processing time	[1, 10] [1, 20]

Table 7. Demand pattern for the newly generated problems.

Demand pattern	<i>N</i>	
	5	10
1	16,1,1,1,1	11,1,1,1,1,1,1,1,1,1
2	15,2,1,1,1	10,2,1,1,1,1,1,1,1,1
3	13,4,1,1,1	9,3,1,1,1,1,1,1,1,1
4	10,5,2,2,1	8,4,1,1,1,1,1,1,1,1
5	8,7,2,2,1	7,5,1,1,1,1,1,1,1,1
6	6,6,5,2,1	6,5,2,1,1,1,1,1,1,1
7	5,5,5,3,2	5,5,3,1,1,1,1,1,1,1
8	5,4,4,4,3	4,4,4,2,1,1,1,1,1,1
9	4,4,4,4,4	2,2,2,2,2,2,2,2,2,2

10 independent problems for each factor combination. This results in a total of 360 problem instances ($2 \times 2 \times 9 \times 10 = 360$).

The results indicate that BS-6 performs very well. The significance levels (*p*-values) in table 8 show that the performance of the proposed algorithm is significantly better than the most competitive existing method for each configuration. This verifies that the proposed BS methods are effective in solving MMAL for the load-leveling measure as well as the part-usage variation.

6. Conclusion

In this paper, we study the mixed-model assembly line sequencing problem for two objectives: part-usage variation and load-leveling. We propose six different BS algorithms (BS-1 to BS-6) to solve these problems. Some of these algorithms are developed by incorporating enhancement tools (backtracking and EOI) in the standard BS procedure.

The results of extensive experiments indicate that BS-4 and BS-6 are generally better than the well-known heuristics on both the benchmark problems and the data sets generated in this study.

As a future research topic, we should further test the proposed algorithms using additional problem instances. We can also develop iterative heuristics or meta-heuristics (simulated annealing, TS, GA, and ant colony algorithms) to solve the same problem. Another research direction would be to study this problem in asynchronous lines and hybrid systems. Finally, it may be interesting to study the

Table 8. The computational results for the load-leveling problem.

Structure	N	Processing times	β	Heuristic	Z_{avg}	p -value
1	5	1–10	–	GC	304.154	0.0001
				2-step	278.502	
				Variance	283.842	
				2-step/var.	273.770	
				3 BS(Leu)	278.981	
				3 BS-4	272.006	
3 BS-6	269.876					
2	5	1–20	–	GC	1305.1	<0.0001
				2-step	1228.1	
				Variance	1229.1	
				2-step/var.	1198.9	
				3 BS(Leu)	1227.4	
				3 BS-4	1193.3	
3 BS-6	1179.8					
3	10	1–10	–	GC	934.525	<0.0001
				2-step	859.914	
				Variance	862.802	
				2-step/var.	831.287	
				5 BS(Leu)	835.784	
				5 BS-4	824.710	
5 BS-6	815.685					
4	10	1–20	–	GC	4084.1	0.0004
				2-step	3777.9	
				Variance	3812.8	
				2-step/var.	3634.8	
				5 BS(Leu)	3672.7	
				5 BS-4	3607.9	
5 BS-6	3572.7					

line balancing and sequencing problems simultaneously. A solution of the combined problem can be more useful than those obtained by treating them separately.

References

- Bard, J.F., Shtub, A. and Joshi, S.B., Sequencing mixed-model assembly lines to level parts usage and minimize line length. *Int. J. Prod. Res.*, 1994, **32**, 2431–2454.
- Bautista, J., Companys, R. and Corominas, A., Heuristics and exact algorithms for solving the Monden problem. *Eur. J. Oper. Res.*, 1996, **88**, 101–113.
- Celano, G., Costa, A. and Fichera, S., A comparative analysis of sequencing heuristics for solving the Toyota goal chasing problem. *Rob. Comp. Integ. Manufac.*, 2004, **20**, 573–581.
- Dar-El, E.M. and Cothor, R.F., Assembly line sequencing for model mix. *Int. J. Prod. Res.*, 1975, **13**, 463–477.
- Dar-El, E.M. and Cucuy, S., Optimal mixed-model sequencing for balanced assembly lines. *Omega*, 1977, **5**, 333–342.
- Dar-El, E.M., Mixed-model assembly line sequencing problems. *Int. J. Manag. Sci.*, 1978, **6**, 343–323.

- Ding, F.Y. and Cheng, L., An effective mixed-model assembly line sequencing heuristic for just-in-time production systems. *J. Oper. Manag.*, 1993, **11**, 45–50.
- Drexel, A. and Kimms, A., Sequencing JIT mixed-model assembly lines under station-load and part-usage constraints. *Manag. Sci.*, 2001, **47**, 480–491.
- Duplaga, E.A., Hahn, C.K. and Hur, D., Mixed-model assembly line sequencing at Hyundai motor company. *Prod. Inven. Manag. J.*, 1996, **37**, 20–26.
- Duplaga, E.A. and Bragg, D.J., Mixed-model assembly line sequencing heuristics for component parts usage: a comparative analysis. *Int. J. Prod. Res.*, 1998, **36**, 2209–2224.
- Erel, E., Sabuncuoglu, I. and Sekerci, H., Stochastic assembly line balancing using beam search. *Int. J. Prod. Res.*, 2005, **43–7**, 1411–1426.
- Gocgun, Y., Beam search algorithms for the mixed model assembly line sequencing problem, MS thesis, Bilkent University, 2005.
- Hyun, C.J., Kim, Y. and Kim, Y.K., A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines. *Comp. Oper. Res.*, 1998, **25**, 675–690.
- Inman, R.R. and Bulfin, R.L., Sequencing JIT mixed-model assembly lines. *Manag. Sci.*, 1991, **37**, 901–904.
- Jin, M. and Wu, S.D., A new heuristic method for mixed model assembly line balancing problem. *Comp. Indus. Eng.*, 2002, **44**, 159–169.
- Kilbridge, M. and Webster, L., The assembly line model mix sequencing problem, in *Proceedings of The Third International Conference on Operations Research*, Oslo, 1963, pp. 247–260.
- Kim, Y.K., Hyun, C.J. and Kim, Y., Sequencing in mixed model assembly lines: a genetic algorithm approach. *Comp. Oper. Res.*, 1996, **23**, 1131–1145.
- Korkmazel, T. and Meral, S., Bicriteria sequencing methods for the mixed-model assembly line in just-in-time production systems. *Eur. J. Oper. Res.*, 2001, **131**, 188–207.
- Kubiak, W. and Sethi, S.P., A note on level schedules for mixed-model assembly lines in just-in-time production systems. *Manag. Sci.*, 1991, **37**, 121–122.
- Kubiak, W., Minimizing variation of production rates in just-in-time systems: a survey. *Eur. J. Oper. Res.*, 1993, **66**, 259–271.
- Leu, Y., Huang, P.Y. and Russell, R.S., Using beam search techniques for sequencing mixed-model assembly lines. *Ann. Oper. Res.*, 1997, **70**, 379–397.
- Mansouri, S.A., A multi-objective genetic algorithm for mixed-model sequencing on JIT assembly lines. *Eur. J. Oper. Res.*, 2005, **167**, 696–716.
- McMullen, P.R. and Frazier, G.V., A simulated annealing approach to mixed-model sequencing with multiple objectives on a just-in-time line. *IIE Trans.*, 2000, **32**, 679–686.
- McMullen, P.R. and Tarasewich, P., A beam search heuristic method for mixed-model scheduling with setups. *Int. J. Prod. Econ.*, 2005, **96**, 273–383.
- Merengo, C., Nava, F. and Pozzetti, A., Balancing and sequencing manual mixed-model assembly lines. *Int. J. Prod. Res.*, 1999, **37**, 2835–2860.
- Miltenburg, J., Level schedules for mixed-model assembly lines in just-in-time production systems. *Manag. Sci.*, 1989, **35**, 192–207.
- Miltenburg, G.J. and Goldstein, T., Developing production schedules which balance part usage and smooth production loads for just-in-time production systems. *Naval Res. Log.*, 1991, **38**, 893–910.
- Miltenburg, J. and Sinnamon, G., Scheduling mixed-model multi-level just-in-time production systems. *Int. J. Prod. Res.*, 1989, **27**, 1487–1509.
- Miltenburg, J. and Sinnamon, G., Algorithms for scheduling multi-level just-in-time production systems. *IIE Trans.*, 1992, **24**, 121–130.
- Monden, Y., *Toyota Production System*, 1983 (Institute of Industrial Engineers: Norcross, Georgia, USA).
- Ng, W.C. and Mak, K.L., A branch and bound algorithm for scheduling just-in-time mixed-model assembly lines. *Int. J. Prod. Econ.*, 1994, **33**, 169–183.
- Ponnambalam, S.G., Aravindan, P. and Rao, M.S., Genetic algorithms for sequencing problems in mixed model assembly lines. *Comp. Indus. Eng.*, 2003, **45**, 669–690.
- Sabuncuoglu, I. and Bayiz, M., Job shop scheduling with beam search. *Eur. J. Oper. Res.*, 1999, **118**, 390–412.

- Sabuncuoğlu, I. and Kizilisik, O.B., Reactive scheduling in a dynamic and stochastic FMS environment. *Int. J. Prod. Res.*, 2003, **41**, 4211–4231.
- Sumichrast, R.T. and Russell, R.S., Evaluating mixed-model assembly line sequencing heuristics for just-in-time production systems. *J. Oper. Manag.*, 1990, **9**, 371–390.
- Sumichrast, R.T., Oxenrider, K.A. and Clayton, E.R., An evolutionary algorithm for sequencing production on a paced assembly line. *Dec. Sci.*, 2000, **31**, 149–172.
- Thomopoulos, N.T., Line balancing-sequencing for mixed-model assembly. *Manag. Sci.*, 1967, **14**, 59–75.
- Yamashita, H. and Okamura, K., A heuristic algorithm for the assembly line model-mix sequencing problem to minimize the risk of stopping the conveyor. *Int. J. Prod. Res.*, 1979, **17**, 233–246.
- Yano, C.A. and Rachamadugu, R., Sequencing to minimize work overload in assembly lines with product options. *Manag. Sci.*, 1991, **37**, 572–586.
- Zeramardini, W., Aigbedo, H. and Monden, Y., Bicriteria sequencing for just-in-time mixed-model assembly lines. *Int. J. Prod. Res.*, 2000, **38**, 3451–3470.
- Zhu, J. and Ding, F.Y., A transformed two-stage method for reducing the Part-usage variation and a comparison of the product-level and part-level solutions in sequencing mixed-model assembly lines. *Eur. J. Oper. Res.*, 2000, **127**, 203–216.