Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 186 (2008) 915–930
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Discrete Optimization

Backtracking and exchange of information: Methods to enhance a beam search algorithm for assembly line scheduling

İhsan Sabuncuoğlu [a,*], Yasin Gocgun [b], Erdal Erel [c]

[a] Industrial Engineering Department, Bilkent University, 06800 Ankara, Turkey
[b] Industrial Engineering, University of Washington, Seattle, WA 98195-2650, United States
[c] Management Department, Bilkent University, 06800 Ankara, Turkey

Received 19 November 2005; accepted 6 February 2007
Available online 23 March 2007

Abstract

Beam search (BS) is used as a heuristic to solve various combinatorial optimization problems, ranging from scheduling to assembly line balancing. In this paper, we develop a backtracking and an exchange-of-information (EOI) procedure to enhance the traditional beam search method. The backtracking enables us to return to previous solution states in the search process with the expectation of obtaining better solutions. The EOI is used to transfer information accumulated in a beam to other beams to yield improved solutions.

We developed six different versions of enhanced beam algorithms to solve the mixed-model assembly line scheduling problem. The results of computational experiments indicate that the backtracking and EOI procedures that utilize problem specific information generally improve the solution quality of BS.

Keywords: Beam search; Heuristic; Assembly line sequencing

1. Introduction

In this paper, we propose an enhanced beam search (BS) algorithm to solve combinatorial optimization problems. The proposed algorithm is developed by incorporating specific enhancement tools into the traditional BS method.

BS is a constructive type heuristic and has been around for at least two decades. It was first used in artificial intelligence for the problem of speech recognition (Lowerre, 1976). Later, it was applied to optimization problems (see Ow and Morton, 1988; Chang et al., 1989; Sabuncuoglu and Karabuk, 1998).

It is a fast and approximate branch and bound method, which operates in a limited search space to find good solutions for optimization problems. It searches a limited number of solution paths in parallel, and progresses level by level without backtracking.

In this paper, we introduce two new features, namely backtracking and exchange-of-information (EOI); these enhance the traditional BS method.

* Corresponding author. Tel./fax: +90 312 2664126.
E-mail addresses: sabun@bilkent.edu.tr (İ. Sabuncuoğlu), gocgun@u.washington.edu (Y. Gocgun), erel@bilkent.edu.tr (E. Erel).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2007.02.024

The enhanced BS is applied to the mixed-model assembly line (MMAL) sequencing problem. The results of our computational experiments indicate that the proposed BS algorithm with these additional enhancements is superior to the traditional BS method and other heuristic approaches in the literature. Based on the experience gained in this study, we see great potential for the BS enhancement tools to solve other optimization problems.

The rest of the paper is organized as follows: we review the literature in Section 2. We discuss the problem domain and the state-of-the-art heuristic procedures in Section 3. We give the description of the proposed algorithm and the enhancement tools in Section 4. We present the results of our computational experiments in Section 5. Finally, we give concluding remarks and further research directions in Section 6.

2. Literature review

Beam search (BS) is an adaptation of the branch and bound method in which only some nodes are evaluated in the search process. In this method, only β promising nodes, called beam width number of nodes, are kept for further sprouting at any level (Sabuncuoglu and Bayiz, 1999). The potential promise of each node is determined by a global evaluation function that selects the best nodes and eliminates others. In order to reduce the computational burden of global evaluation, a filtering mechanism can also be used, by which some nodes are eliminated by a local evaluation function prior to the global evaluation.

Since BS was first employed in artificial intelligence (Lowerre, 1976), it has been used in various problem areas. Ow and Morton (1988) use BS to solve the single machine early/tardy problem and the flow shop problem. Chang et al. (1989) develop a BS algorithm for the FMS scheduling problem. In another study, Sabuncuoglu and Karabuk (1998) develop a filtered BS for the FMS scheduling problem with finite buffer capacity, routing and sequencing flexibilities. The studies of Sabuncuoglu and Bayiz (2000), Shayan and Al-Hakim (2002), and Pacciarelli and Pranzo (2004) are other scheduling examples of BS.

BS has also been applied to other problems: assembly line sequencing (Leu et al., 1997; McMullen and Tarasewich, 2005), assembly line balancing (Erel et al., 2005), stochastic programming (Beraldi and Ruszczynski, 2005), marketing (Alexouda and Paparrizos, 2001), and tool management (Zhou et al., 2005). There are also a few studies in which the solution construction mechanism of local search methods such as the ant colony optimization (ACO) approach and genetic algorithms are hybridized with BS applications (see Alexouda and Paparrizos, 2001; Tillmann and Ney, 2003; Blum, 2005).

In recent years, several enhancement tools have been developed to improve the performance of BS. For example, Honda et al. (2003) propose a backtracking BS algorithm for a multi-objective flowshop problem. In the proposed method, the traditional BS is first performed, and then a backtracking mechanism is repeatedly invoked at some selected nodes to obtain non-dominated solutions. The results of their computational experiments indicate that the proposed algorithm yields better solutions than the standard BS.

Della Croce and T'kindt (2002) and Della Croce et al. (2004) develop a recovering BS (RBS) method for combinatorial optimization problems. The recovering phase aims to recuperate the previous decisions. This step is invoked for each of the beam-width number of best child nodes. For a given node, the recovering phase, by means of interchange operators applied to the current partial schedule, checks whether the current solution is dominated by another partial solution sharing the same search tree level. If so, the current solution is replaced by the new solution. The results indicate that RBS outperforms the traditional BS. Several RBS approaches have also been proposed for other problems (see Valente and Alves, 2005; Ghirardi and Potts, 2005; Esteve et al., 2006). Table 1 further summarizes all these existing studies and BS application in various problem domains.

3. Problem domain

Even though the idea of the proposed enhancement tools is general enough to be applied to any optimization problem, its details are problem specific. Hence, we first introduce our problem domain prior to the description of the algorithm.

Mixed-model assembly lines (MMALs) are multi-level production lines in which a variety of product models are simultaneously assembled one after each other. In these systems, raw materials are fabricated into components, which in turn are combined into sub-assemblies that are transformed into final products.

Table 1
The variations and application areas of beam search

Author	Type of BS	Enhancement/hybridized methods	Application area
Ow and Morton (1988)	Standard	–	Scheduling
Chang et al. (1989)	Standard	–	FMS scheduling
Shahookar et al. (1993)	Hybridized	Genetic algorithm	Layout problem
Leu et al. (1997)	Standard	–	Assembly line sequencing
Sabuncuoglu and Karabuk (1998)	Filtered	–	FMS scheduling
Kim and Kim (1999)	Standard	–	Transportation
Matsuda et al. (2002)	Hybridized	Graph-based Induction Method	Data mining
Sabuncuoglu and Bayiz (2000)	Filtered	–	Scheduling
Ortmanns and Ney (2000)	Enhanced	Look-ahead techniques	Artificial intelligence
Alexouda and Paparrizos (2001)	Hybridized	Genetic algorithm	Marketing
Shayan and Al-Hakim (2002)	Standard	–	Sequencing
Zeng and Martinez (2002)	Enhanced	Varying BS parameters	Neural networks
Wang (2002)	Hybridized	Fuzzy approach	Project scheduling
Honda et al. (2003)	Enhanced	Backtracking	Scheduling
Tillmann and Ney (2003)	Hybridized	Dynamic programming	Artificial intelligence
Pacciarelli and Pranzo (2004)	Filtered	–	Scheduling
Della Croce et al. (2004)	Enhanced	Recovering phase	Scheduling
Kim et al. (2004)	Filtered	–	Sequencing
Abdou and Scordilis (2004)	Standard	–	Artificial intelligence
Zhou and Hansen (2004)	Hybridized	Divide-and-conquer method	Automated planning
Lee and Woodruff (2004)	Standard	–	Metabonomics
Erel et al. (2005)	Standard	–	Assembly line balancing
Beraldi and Ruszczynski (2005)	Filtered	–	Stochastic programming
Zhou et al. (2005)	Standard	–	Tool management
Blum (2005)	Hybridized	ACO approach	Scheduling
Valente and Alves (2005)	Enhanced	Recovering phase	Scheduling
Ghirardi and Potts (2005)	Enhanced	Recovering phase	Scheduling
Zhou and Zhong (2005)	Standard	–	Scheduling
Lim et al. (2006)	Standard	–	Scheduling
Forshed et al. (2005)	Standard	–	Metabonomics
McMullen and Tarasewich (2005)	Standard	–	Assembly line sequencing
Zhou and Hansen (2004)	Enhanced	Backtracking	Automated planning

The MMAL sequencing problem is defined as determining a sequence of product models on the final assembly line to optimize some performance measures. In this study, we use the part usage variation criterion that maintains a constant rate of usage of all parts feeding the final assembly line. This objective requires that products be assembled at rates proportional to their demand, and that parts be pulled through the system at constant rates (Miltenburg and Sinnamon, 1992). Note that we consider variability only at the sub-assembly level, as suggested by Monden (1983).

The mathematical formulation of this problem is first given in Jin and Wu (2002). Their objective function is to minimize the sum of quadratic differences between the actual parts usage and desired parts usage at each stage (i.e. position). At any stage k, the total number of sequenced models must be equal to k, and the number of times model i is sequenced should increase by one or remain the same. In addition, the number of times that model i is sequenced at any stage k should not exceed the demand for this model. The problem is an integer non-linear problem and it is *NP-Hard* in any sense even if the objective is linearized (Jin and Wu, 2002).

The parts usage variation at any stage, i.e., level k is calculated as follows:

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

where $x_{i,k}$ is the total number of times model i sequenced in the first k positions for a specific sequence, $c_{j,i}$ is the number of part j required for model i, kr_j is the desired number of part j consumed in the first k positions for a specific sequence, N is number of different models to be produced, and

C is the number of different parts that can be used by a model.

The state-of-the-art heuristic to solve the MMAL sequencing problem is the two-step variance method developed by Jin and Wu (2002). The variance method is developed to eliminate the myopic feature of a well-known greedy heuristic, the goal chasing method (GCM) developed by Monden (1983). The GCM selects the model that yields the minimum parts usage variation at any stage, ignoring the effect of future sequences. This myopic feature of the GCM is reduced when the effect of the remaining composition is taken into account in selection of the model. In this vein, the variance method integrates the "composition variance" for the remaining composition as the opportunity cost into the total cost. The opportunity cost is multiplied by a discounting coefficient and the model with the minimum total cost is selected at each level. The two-step variance method positions two models for the two subsequent levels and compares alternatives with respect to the combined total variation. The combination of two feasible models with the least total variation is selected and only the first model is positioned in the sequence.

4. Proposed algorithm

This section is organized in two parts: general description of enhanced BS and application to the MMAL problem.

4.1. General description

The representation of a BS tree is shown in Fig. 1. We select the promising β nodes (beam nodes) by invoking local and global evaluations and proceed with the search through these selected nodes. Then we apply the algorithm to these nodes independently and generate one partial tree (i.e., beam) from each of them. After a filtering procedure and using the outcome of the global evaluation, one node is chosen from the descendants of each beam node. This becomes the beam node for the next level. In this way, the search progresses through β parallel beams.

The proposed algorithm is based on a BS in which each node corresponds to a solution state representing the partial sequence of products. The leaf nodes correspond to the full sequence of products (see Fig. 1). The potential promise of each node is determined by the global evaluation function, which typically estimates the minimum total cost of the best solution that can be obtained from the partial schedule represented by that node. In the proposed algorithm, a filtering procedure is also used to eliminate some nodes by a computationally fast method (i.e., local evaluation function), and only the remaining nodes (*filter width*) are globally evaluated. The value of the local evaluation function is the parts-usage variation. 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 the subsequent levels. Hence, it estimates the solution quality of a partial solution, instead of a full solution; this allows us to globally evaluate the candidate nodes quickly (this procedure is explained in Section 4.2.1).

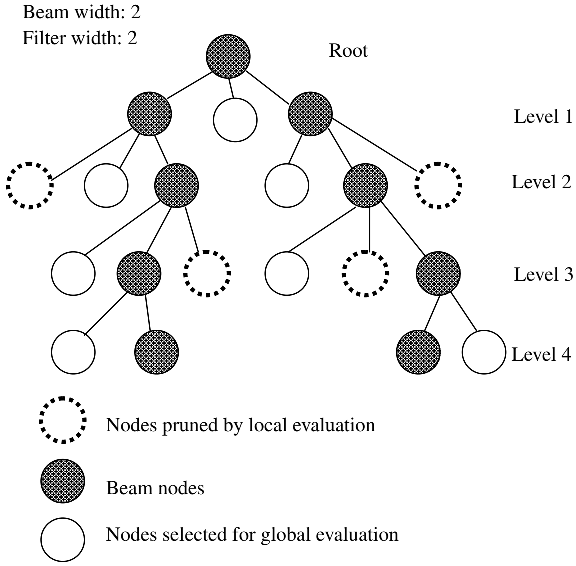


Fig. 1. Representation of a BS tree.

The proposed algorithm incorporates two new enhancement tools, backtracking and information exchange, to improve the performance of BS. *Backtracking* is the process of revisiting previous solution states in the search tree with the expectation of obtaining better solutions. The motivation for 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 solution states at earlier levels.

The equivalence theorem enables to determine if the current beam at a stage is equivalent to another beam in terms of the number of products sequenced up to that stage and identify the inferior beam.

Search is resumed on the superior beam, but it is backtracked to the previous stage on the inferior beam in a different direction.

The second enhancement tool is the exchange-of-information (EOI) by which part of a solution from one beam is transferred to another beam hoping that the resulting beam will lead to better solutions. EOI is carried out in such a way that a *partial solution* consisting of the product sequence between the first and the last appearance of a product is transferred to all other beams. All these enhancement procedures will be explained in detail within the context of the MMAL sequencing problem in the next section. The basic steps of the proposed algorithm are as follows.

Notation

BL	beginning level for EOI
I	interval for EOI
k	indicator for EOI
s	total number of stages (e.g., total number of products to be sequenced)
l	current level in the search tree

Steps of the Proposed Algorithm:

Step 0. (Initialization)

Set $k = 0$ and $l = 0$.

Step 1. Generate descendant nodes.

Step 2. (Determining beam nodes)

Select the best β beam nodes using the *global evaluation function*, and set $l = l + 1$.

Step 3. (Search the beam nodes)

Step 3.1. For each beam:

Step 3.1.1. Using the *local evaluation function*, keep at most α nodes emanating from the current beam node.

Step 3.1.2. Using the *global evaluation function*, select the best node among w of them.

Step 3.2. Set $l = l + 1$.

Step 4. (Exchange of information)

Step 4.1. If $l = BL + I * k$ and $l \leqslant s$, then

Step 4.1.1. For each beam:

Step 4.1.1.1. Select the best beam among the alternative solutions generated by the EOI procedure.

Step 4.1.2. Set $k = k + 1$.

Step 5. (Backtracking)

Step 5.1. If $l = s$, stop the algorithm.

Step 5.2. If equivalency is observed, create an alternative beam for each inferior beam using the backtracking procedure.

Step 5.3. Go to Step 1.

4.2. Application of enhancement tools

4.2.1. Backtracking procedure

The backtracking procedure is applied whenever *equivalence* is observed following the selection of beam nodes at any level. Beams are considered *equivalent* at a level whenever each product has been sequenced the same number of times at that level. As an illustration, consider the following two beams in Fig. 2: The products A, B, B, C are sequenced in Beam 1 and the products C, A, B, B are sequenced in Beam 2 (see Fig. 2a). Since both of these beams have one A, two B's, and one C, they are considered equivalent.

The cumulative variation of equivalent beams at the current level (i.e., level $k < D_{\mathrm{T}}$) is calculated and inferior beams with larger cumulative variation are identified. Each of these inferior beams is backtracked by moving one level up, and generating the next best (NB) child node. The NB node is further sprouted by selecting the best node using the global evaluation function. The original node, however, is also sprouted by selecting the NB node. Finally, these two generated nodes are evaluated and the one having the least value of global evaluation function plus the variation at the current level is selected.

The backtracking procedure is shown by considering two equivalent beams as in Fig. 2a. After comparing the cumulative variation values of the two beams at level k, Beam 2 is found to be inferior. Then, the NB child node of Beam 2 at level k is further branched by choosing the best node at level $k + 1$; the original node (i.e., product B on Beam 2) at level k is further branched by selecting the NB node at level $k + 1$. Hence, at level $k + 1$ we have two alternative beam nodes; after evaluating these nodes, we continue the search procedure by selecting the superior one.

The backtracking procedure discussed above is based on the equivalence theorem that is stated and proved next.

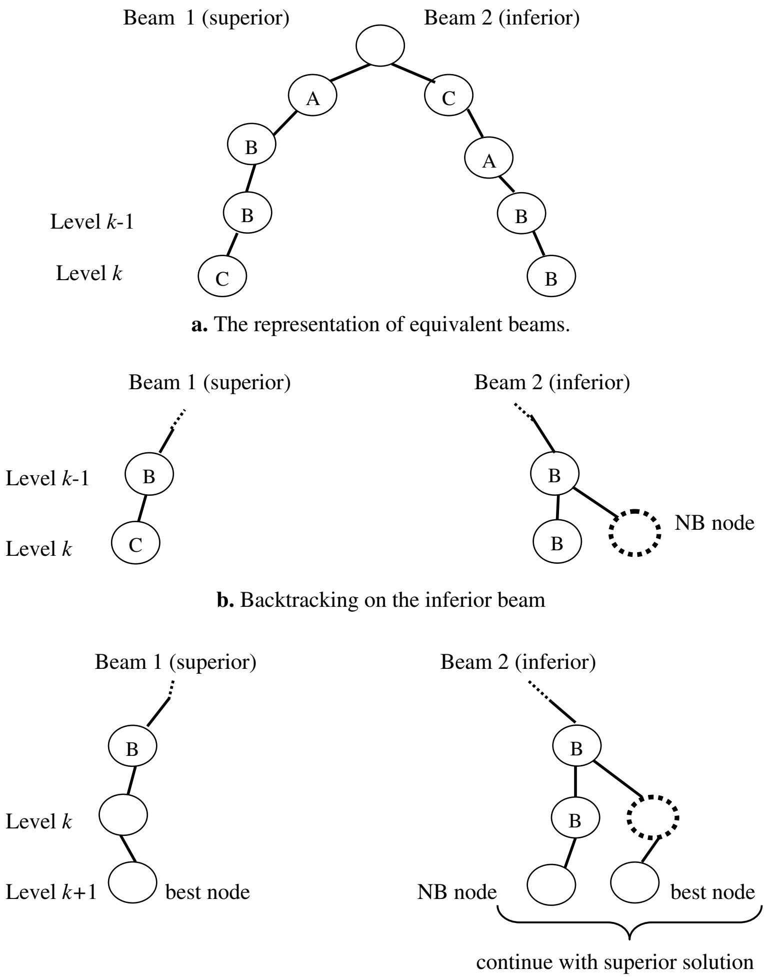
4.2.2. Equivalence theorem

We first present the notation and then the proof of the theorem.

σ_k^i: partial sequence at level k on beam i, $k = 1, \ldots, D_{\mathrm{T}} - 1$

$V(\sigma_k^i)$: parts usage variation for σ_k^i

CV (σ_k^i): cumulative parts usage variation for σ_k^i $(\mathrm{CV}(\sigma_k^i) = \sum_{j=1}^{k} V(\sigma_j^i))$



a. The representation of equivalent beams.

b. Backtracking on the inferior beam

c. The evaluation of the alternatives at the next level (k+1)

Fig. 2. A schematic view of the backtracking procedure.

$GE_j(\sigma_k^i)$: the value of global estimation obtained by completing σ_k^i up to level j, $j = k+1,\ldots,D_T$

Theorem. *Let σ_k^i and σ_k^j be two equivalent sequences belonging to beam i and beam j, respectively. If the result of global and local evaluation functions only depend on the remaining products at level $k-1$, and $CV(\sigma_k^i) < CV(\sigma_k^j)$, then the following inequality holds (as long as the same BS parameters are used in the remaining levels of the search tree): $CV(\sigma_{D_T}^i) < CV(\sigma_{D_T}^j)$.*

Proof. First consider the case in which only the global evaluation function is invoked. Since the remaining products to be scheduled for beam i and beam j are identical, during global evaluation the same nodes are considered at level $k+1$ for each beam. Let the products chosen for beam i and beam j at level $k+1$ be m and n, respectively, such that $m \neq n$; hence:

$$m = \operatorname{argmin}_s\{GE_t(\sigma_k^i \cup s), s : x_{sk} < d_k, t = k+1,\ldots,D_T\}, \tag{2}$$

where x_{sk}: number of times product s is sequenced up to level k,

d_s: demand for product s

The following inequality is obtained from (1):

$$GE_t(\sigma_k^i \cup m) < GE_t(\sigma_k^i \cup n). \tag{3}$$

After applying the same steps for beam j, the following inequality is obtained:

$$GE_t(\sigma_k^j \cup n) < GE_t(\sigma_k^i \cup m). \tag{4}$$

Since the values of global estimations are equal for the equivalent sequences (i.e., $\sigma_k^i \cup l$ and $\sigma_k^j \cup l$), the following equation is obtained:

$$\mathrm{GE}_t(\sigma_k^i \cup l) = \mathrm{GE}_t(\sigma_k^j \cup l), \quad l = m, n. \tag{5}$$

Hence, the inequalities (3) and (4) contradict each other, implying $m = n$. As a result, the same product is chosen for each beam at level $k + 1$. The selection of the same product at further levels for each beam is pursued since the beams are also equivalent at each of the remaining levels. Since variation at any level only depends on the number of times each product is sequenced up to that level, and the sequences of beam i and beam j are also equivalent at level $k + 1$; the following equality is obtained:

$$V(\sigma_k^i \cup m) = V(\sigma_k^j \cup m). \tag{6}$$

It is inferred from Eq. (6) that the cumulative variation for each beam is equally incremented at subsequent levels. Accordingly, if beam i is superior to beam j at level k, it is also superior at the last level (i.e., level D_{T}), which implies:

$$\mathrm{CV}(\sigma_{D_T}^i) < \mathrm{CV}(\sigma_{D_T}^j). \qquad \square$$

This theorem also holds with the filtering procedure. This is due to the fact that the same candidate nodes are filtered out for beam i and beam j, since the local evaluation function depends on the remaining products at level $k - 1$. As a result, the same nodes are considered at further levels for each beam during the global evaluation, which also implies that cumulative variation for each beam is equally incremented at subsequent levels.

4.2.3. Exchange-of-information (EOI) procedure

EOI is performed in the following way. First, the last product (i.e., product i at level k in Beam 2 in Fig. 3a) of a beam is chosen. Then, a *partial solution* consisting of the product sequence between the first and the last appearance of that product, i.e., product i is transferred to all other beams (see Fig. 3b). This transfer is carried out as follows.

First, we insert the *partial solution* to a new beam at the level where product i appears first in the sequence. If the level to which the new beam extends is smaller than the current level, then we repeat this insertion process at the next level where i appears next. Otherwise, we truncate the *partial solution* at level k. Note that in either case, we repeat this insertion process until a feasible solution is obtained.

The new beams are compared against the original beams; the beams with the smallest value of the global evaluation function plus cumulative variation at the previous level (i.e., one level before the current level) are retained to continue the search procedure (see Fig. 3b). The procedure is repeated for subsequent levels. When EOI and backtracking procedures are used at the same level, EOI is invoked before backtracking. Moreover, if a new beam is chosen as a result of EOI and there exits equivalency between the new beam and another beam where the new beam is the inferior beam, the backtracking procedure is skipped. Instead, the NB is selected for the next level.

4.2.4. Global evaluation

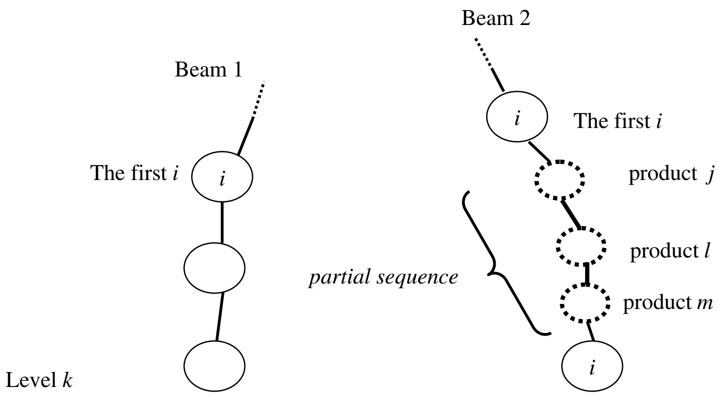
The global evaluation function used in the proposed BS calculates the sum of variation at the current level, i.e., level k and subsequent three levels. The mathematical expression of this function has been given in Eq. (1).

The global evaluation function uses a heuristic procedure to determine the quality of a partial sequence for the next three levels. The procedure first selects the product with the minimum variation at the next level, i.e., level $k + 1$ for a sequence at level k. For the last two levels, it calculates the combined variations (i.e., the variations at levels $k + 1$ and $k + 2$) with each of the alternative product pairs. Then, it selects the first product pair with the lowest combined variation.

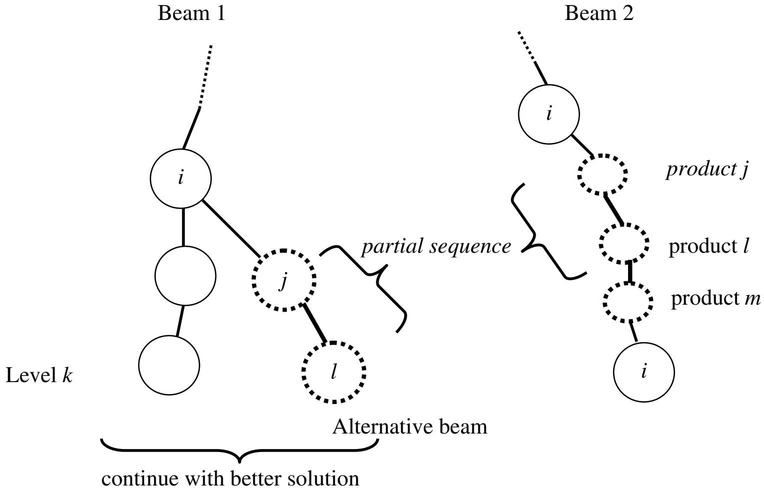
Since minimization of the sum of the variation at each level is the objective function, it is expected that an optimal/near-optimal sequence will yield little variation at each level. This implies that the amount of actual usage is very close to desired usage for each part at a particular level. Hence, if the variation at level $k + 1$ is ignored, some of the alternative pairs can be eliminated without considering the sequence at level $k + 1$. A detailed explanation of the methodology for selecting the last two products is given below:

First, all of the feasible 3-level sequences starting with the product at level $k + 1$ are created. Then, the total variation for each of them is calculated by summing the variations at level 2 and level 3, and the variation at level 2 for a sequence with the last two products. As an example, for a sequence of A (the last product of the sequence at level $k + 1$), B, and D, the total variation is calculated as follows:

$$TV(A,B,D) = V(A,B) + V(A,B,D) + V(B,D).$$



a. Constructing a partial solution from Beam 2.



b. The comparison of the original beam and new beam

Fig. 3. The schematic view of EOI.

This equation implies that if the $TV(A,B,D)$ is sufficiently small, the products B and D are suitable to be sequenced after A, and D is suitable to be chosen after B. If $TV\ (A,B,D)$ is significantly large, products B and D should not be appended to any partial solution that ends with A. This is because of the fact that at the last three levels in any near-optimal solution that ends with A, B, and D, the total variation most probably increases dramatically.

After calculating the value of total variation for each 3-level sequence, the best w solutions (i.e., the ones that have the minimum total variation) of at most n^2 alternatives is considered for global estimation. Then w solutions are created by adding the last two products of the filtered 3-level sequences to the current solution (i.e., the sequence at level $k+1$) and the pair that yields the minimum combined variation is selected. Finally, the first product of the best pair is chosen for level $k+2$. Similarly, the product for level $k+3$ is selected using the same procedure.

The enhancement tools are illustrated on an example in the Appendix.

4.3. Different versions of the proposed method

Up to now, we assume that beams progress independent of each other. However, the proposed 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. In this section, we consider this new version, expecting to obtain better solutions. Note that the filtering procedure is also invoked for each beam independently in this version.

In order to observe the effect of backtracking and EOI on the performance of the procedure, we develop the following six versions of the proposed algorithm.

BS-1: A BS technique in which beams progress independently.
BS-2: A BS technique in which beams progress independently, and backtracking procedure is invoked.
BS-3: A BS technique in which beams progress independently, and EOI procedure is invoked.
BS-4: A BS technique in which beams progress independently, and backtracking and EOI procedures are invoked.
BS-5: A BS technique with dependent beams.
BS-6: A BS technique in which beams progress dependently, and EOI procedure is invoked.

Note that we do not apply the backtracking procedure for the BS method with dependent beams as backtracking requires that beams progress independently.

5. Computational results

In this section, we first present the results of experiments that compare the proposed method with the 2-step Variance method. Then, we examine the effects of the backtracking and EOI procedures in Section 5.2. We further study the effects of EOI at different positions in Section 5.3.

5.1. The evaluation of the proposed algorithm

All the six versions of BS (i.e, BS-1, ..., BS-6) are compared with the 2-step Variance Method. In the implementation process, we tune-up the parameters of the algorithms, including filter width, EOI beginning level, and number of stages to invoke EOI.

5.1.1. 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 used for various part-product combinations and demand patterns. The value of the objective function (Z_{avg}) is the average of the variations obtained for demand patterns.

The results indicate that the performance of BS-6 is generally better than the other versions. Specifically, BS-6 is statistically better than the 2-step Variance method for all structures, except Structure 3 and Structure 5 (see Table 2). BS-4 and BS-6 are numerically the best performers (even though statistically insignificant) in structures 3 and 5, respectively. In addition, BS-4 is superior to the 2-step Variance method for structures 2, 6.1, 6.2, and 6.3. Moreover, we observe no significant difference between BS-5 and BS-6, implying that EOI is not very effective in improving the solution quality of BS for dependent beams.

We compare the computation times of the proposed algorithms and the heuristics in the literature. We observe that all of the proposed algorithms require larger computational effort relative to the heuristics in the literature. To be more specific, the CPU times of the proposed algorithms and the heuristics for Structure 6.2 are depicted in Table 3. The larger computational effort of the proposed algorithms is due to the existence of the enhancement tools and the global evaluation function, whereas most of the heuristics in the literature are greedy in nature. Although the proposed algorithms require relatively higher computation times than the heuristics, they are still fast for practical purposes. For example, average CPU time of the BS-6 for the largest problem size is only about 220 milliseconds.

Computational experiments are also conducted on new data sets that employ the following factors: (1) number of products, (2) quantity per assembly, and (3) degree of commonality. Nine different demand patterns are generated for each configuration as discussed in Ding and Cheng (1993). In each experimental condition, 10 independent replications are made for statistical accuracy.

As can be seen in Table 4, BS-6 statistically outperforms the 2-step Variance method in all of the structures except Structures 4 and 7 for which BS-4 displays better performance. However, BS-6 is still statistically better than BS-4 for other structures.

In order to test the efficiency of the backtracking procedure, we compare the performances of BS-1 and BS-2. As explained before, beams in BS-1 progress independently and none of the enhancement tools are invoked; however in BS-2, beams progress independently and only the backtracking procedure is performed. To measure the effects of the EOI procedure, we examine the relative performances of BS-1 and BS-3. The computational results (Table 2)

Table 2
The results obtained by the data sets given in the literature

Structure	N	D	# of demand pattern	β	Heuristic	Z_{avg}	p-Value
1	4	20	45	–	2-step/var.	61.973	0.003
				4	BS-1	62.24	
				4	BS-2	61.742	
				4	BS-3	61.795	
				4	BS-4	60.818	
				4	BS-5	60.356	
				4	BS-6	60.124	
2	4	20	45	–	2-step/var.	138.376	0.0005
				4	BS-1	139.541	
				4	BS-2	137.852	
				4	BS-3	137.545	
				4	BS-4	134.849	
				4	BS-5	134.089	
				4	BS-6	133.529	
3	4	20	45	–	2-step/var.	137.984	0.478
				4	BS-1	141.642	
				4	BS-2	140.078	
				4	BS-3	140.82	
				4	BS-4	137.598	
				4	BS-5	137.620	
				4	BS-6	137.309	
4	4	20	45	–	2-step/var.	15.732	0.032
				4	BS-1	15.821	
				4	BS-2	15.803	
				4	BS-3	15.812	
				4	BS-4	15.714	
				4	BS-5	15.661	
				4	BS-6	15.652	
5	4	20	45	–	2-step/var.	157.505	0.243
				4	BS-1	163.232	
				4	BS-2	158.456	
				4	BS-3	161.241	
				4	BS-4	154.827	
				4	BS-5	156.705	
				4	BS-6	155.367	
6.1	5	20	45	–	2-step/var.	48.216	0.0001
				4	BS-1	48.18	
				4	BS-2	47.891	
				4	BS-3	47.562	
				4	BS-4	47.287	
				4	BS-5	46.58	
				4	BS-6	46.402	
6.2	5	48	1	–	2-step/var.s	138.040	–
				5	BS-1	137.04	
				5	BS-2	137.04	
				5	BS-3	125.708	
				5	BS-4	130.208	
				5	BS-5	126.708	
				5	BS-6	125.708	
6.3	5	280	1	–	2-step/var.	574.018	–
				5	BS-1	599.660	
				5	BS-2	599.660	
				5	BS-3	552.732	
				5	BS-4	560.660	
				5	BS-5	587.875	
				5	BS-6	549.446	

Table 3
The comparison of the algorithms in terms of computational effort

Structure	β	Heuristic	Z	CPU time (in milliseconds)
6.2	–	GC	226.875	10
	–	2-step	153.040	30
	–	Variance	146.040	10
	–	2-step/var.s	138.040	20
	5	BS(Leu)	136.042	30
	5	BS-1	137.04	261
	5	BS-2	137.04	270
	5	BS-3	125.708	330
	5	BS-4	130.208	331
	5	BS-5	126.708	171
	5	BS-6	125.708	220

indicate that the EOI procedure improves the solution quality in all structures except Structure 4.

5.2. The effects of backtracking and EOI

We also observe that the performance improvement due to EOI gets larger as demand increases. This arises because the number of alternative solutions created during the search procedure increases as demand increases.

We further test the combined effect of backtracking and EOI on solution quality, by comparing BS-1 and BS-4 (see Table 2). The results indicate that BS-4 statistically outperforms BS-1 in all structures ($p < 0.044$). This implies that the enhancement tools are generally more effective when they are used together.

5.3. The effect of EOI at different positions

The results obtained up to now indicate that EOI generally improves the solution quality. However, a further analysis is needed to determine the interval for which EOI is more effective (i.e., frequency of

Table 4
The computational results obtained by the newly generated data sets

Configuration	N	D	QPA	DOC (%)	β	Heuristic	Z_{avg}	p-Value
1	5	20	1–10	0–20	–	2-step/var.	1043.8	<0.0001
					3	BS-1	1041.7	
					3	BS-4	1038.4	
					3	BS-5	1033.1	
					3	BS-6	1032.4	
2	5	20	1–10	60–80	–	2-step/var.	1290.8	<0.0001
					3	BS-1	1282.9	
					3	BS-4	1279.5	
					3	BS-5	1274	
					3	BS-6	1273.2	
3	5	20	1–20	0–20	–	2-step/var.	3840	0.002
					3	BS-1	3826.3	
					3	BS-4	3822.7	
					3	BS-5	3797.07	
					3	BS-6	3796.5	
					3	BS-1	4589.4	
4	5	20	1–20	60–80	–	2-step/var	4627.7	<0.0001
					3	BS-1	4589.4	
					3	BS-4	4584.7	
					3	BS-5	4565.5	
					3	BS-6	4564.2	
5	20	40	1–10	0–20	–	2-step/var.	11093.5	<0.0001
					10	BS-1	11035.6	
					10	BS-4	**10899.2**	
					10	BS-5	10984.6	
					10	BS-6	10983.7	
6	20	40	1–10	60–80	–	2-step/var.	29329.6	<0.0001
					10	BS-1	28839.2	
					10	BS-4	28734.8	
					10	BS-5	28391.7	
					10	BS-6	28389.1	
7	20	40	1–20	0–20	–	2-step/var.	36902.4	0.0002
					10	BS-1	36796.8	
					10	BS-4	36443.9	
					10	BS-5	36562.3	
					10	BS-6	36564.2	
8	20	40	1–20	60–80	–	2-step/var.	112,033	<0.0001
					10	BS-1	109820.2	
					10	BS-4	109158.3	
					10	BS-5	108688.6	
					10	BS-6	108688.6	

EOI use). Hence, the EOI procedure is invoked at certain stages to observe its effect on the performance measure. For example, in a design with 20 levels, EOI is invoked at the following intervals: 1–4, 5–8, 9–12, 13–16, 17–20. Note that this analysis is performed on BS-3 since beams progress independently in this version. The experiments are conducted for both small (i.e., demand of 20) and large (demand of 260) problems.

The results indicate that if the number of levels (i.e., total demand) is small, invoking the EOI procedure between certain intervals generally does not improve the objective function (see Fig. 4). For Structure 2, EOI statistically improves the solution quality if it is invoked at the middle levels. On large problems, EOI improves solution quality for Structures 1, 2, and 5. As can be seen in Fig. 5, EOI is more effective when invoked at the middle levels.

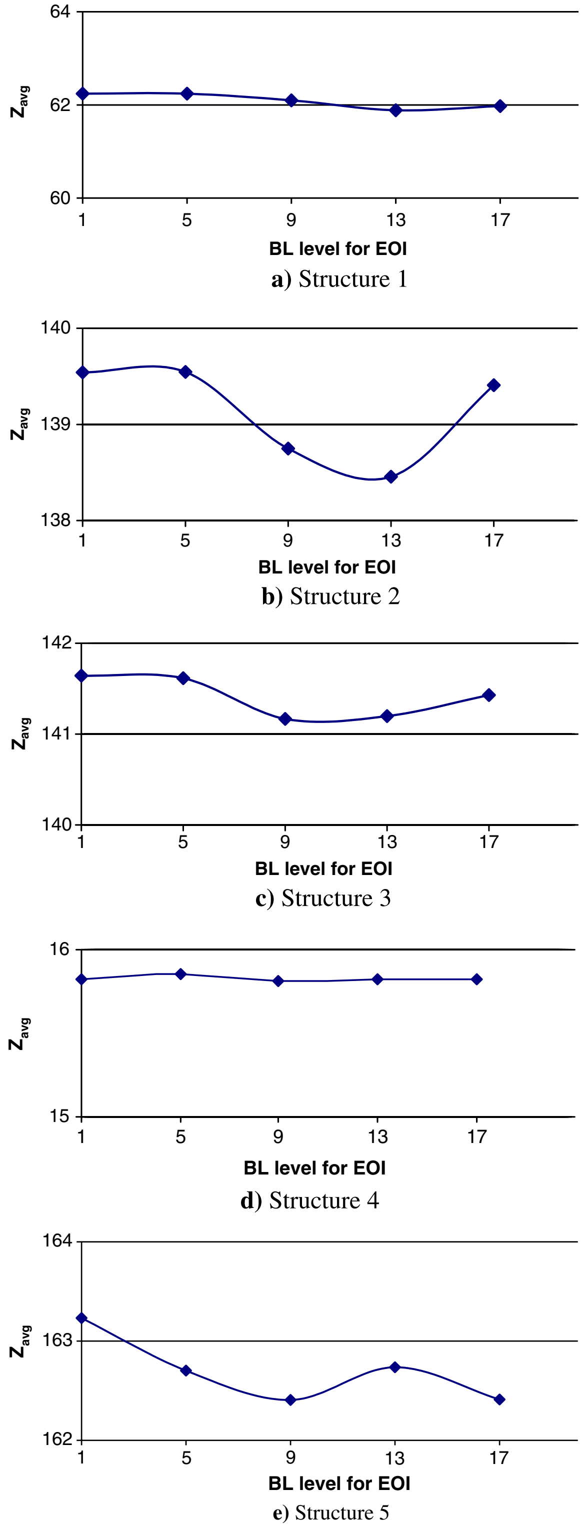


Fig. 4. The effect of EOI on the performance measure when number of levels is 20. BL for EOI refers to the level at which EOI is first invoked.

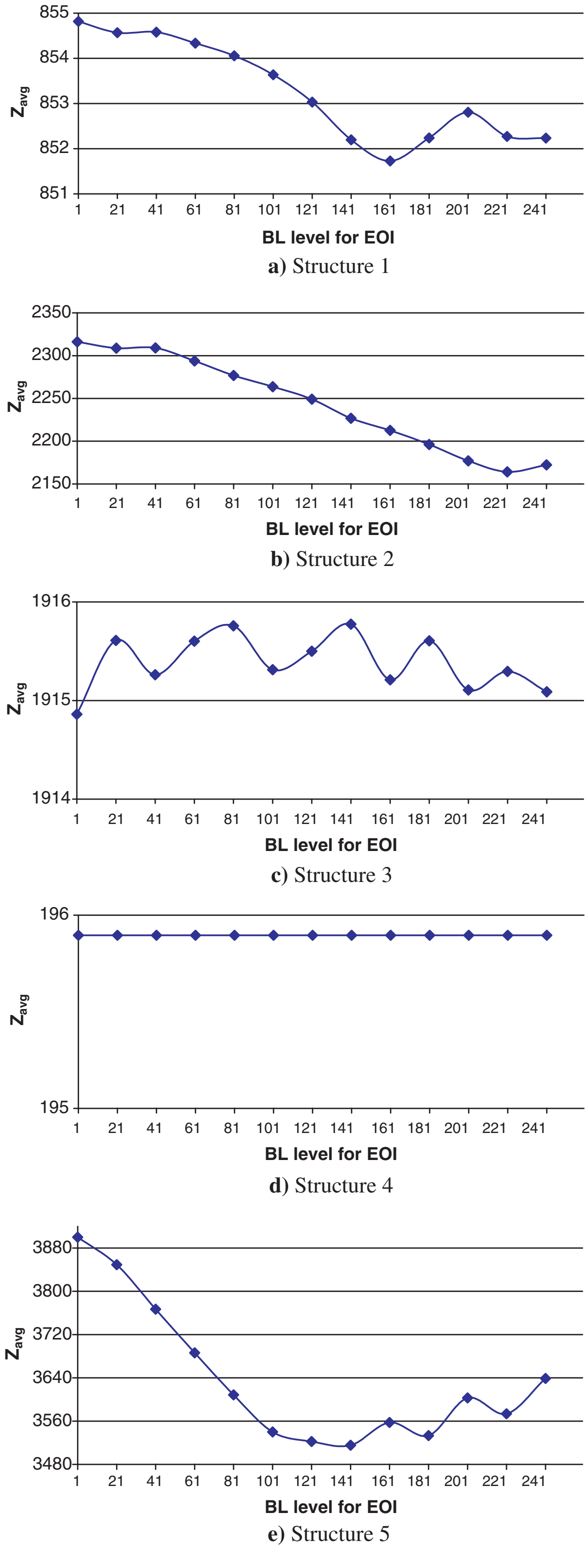


Fig. 5. The effect of information exchange on the performance measure when number of levels is increased to 260.

6. Conclusion

In this paper, we develop backtracking and EOI procedures to enhance the traditional beam search method. The backtracking procedure enables us to return to previous solution states in the search tree with the expectation of obtaining better solutions. The EOI procedure copies a part of the solution from one beam to another beam. The beam with this additional information is expected to yield improved solutions.

We apply the proposed algorithms to the mixed-model assembly line sequencing problem. The results indicate that the backtracking and EOI procedures generally improve solution quality. We also analyze the effect of EOI when it is invoked at different positions in the search tree. The results suggest that EOI is most effective when it is invoked at the middle levels in the search tree.

In this study, we show that the BS method with some appropriate enhancement tools can be used to solve difficult optimization problems. We also note that this enhanced version of BS offers research opportunities in other areas such as, scheduling, assembly line balancing, etc. New backtracking and EOI procedures can be developed to improve the efficiency of BS. Further studies could also identify the problem environments where these enhancements tools would be most effective.

Appendix. An example to illustrate the enhancement tools

The steps of the proposed algorithm are clarified with an example problem. Suppose that there are four different products to be assembled and four components that will be used for these products. The values of $c_{j,i}$, which are taken from Bautista et al. (1996), are presented in Table 5. The demand vector is (2,4,3,1), representing the demand for product 1 is 2, the demand for product 2 is 4, etc.

Table 5
The part structure used for the example problem (Structure 3 in Bautista et al. (1996))

Parts	Products			
	P1	P2	P3	P4
P1	0	0	0	5
P2	3	1	0	5
P3	3	3	5	0
P4	4	6	5	0

The values of the parameters used to implement the algorithm are as follows: the beam width β and filter width (α) are set to 2 and 3, respectively. EOI is invoked only at level 5. Moreover, the width for global evaluation function (w), called global width, is set to 5. Note that this version of the algorithm is BS-4.

In order to show the improvement obtained by BS-4 with respect to the traditional BS method, we first present the solution of the BS method (i.e., BS-1) for the example problem (see Fig. 6). The nodes given in Fig. 8 represent the beam nodes in the search tree. The resultant sequence of Beam 1 yields the CV of 71.8, while the CV of the sequence of Beam 2 is 72.6. Hence, the implementation of the BS method yields the cumulative variation of 71.8, with the following sequence:

2-1-3-1-2-3-4-3-2-2.

The proposed algorithm, however, first invokes the backtracking procedure at level 2, at which the equivalency is observed. As Beam 2 is found to be

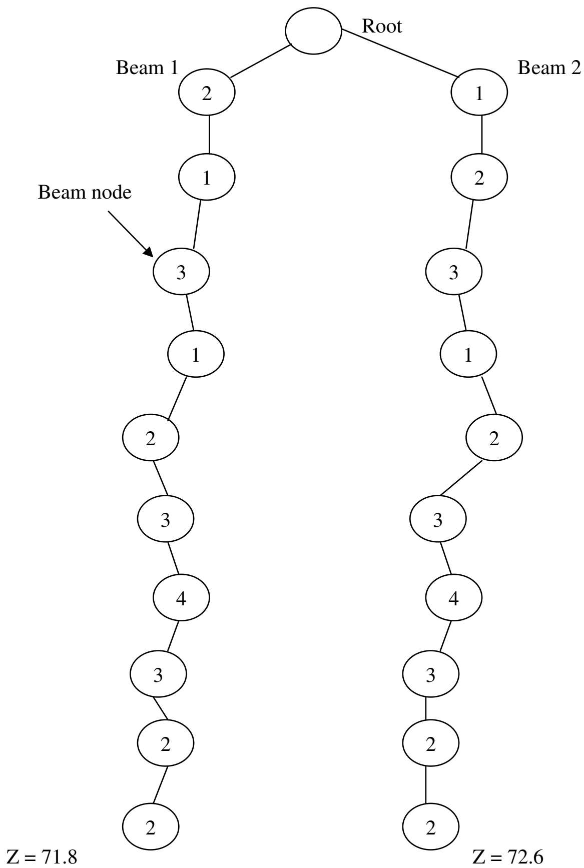
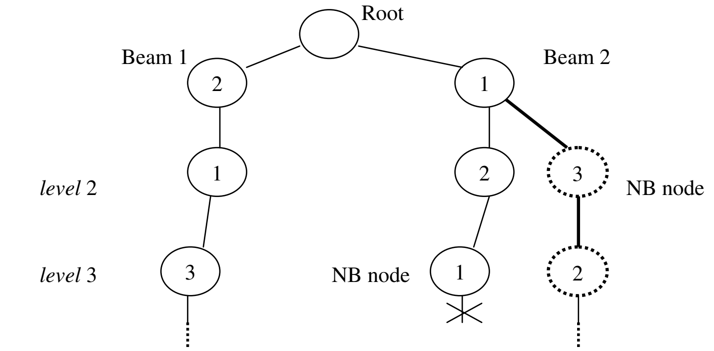
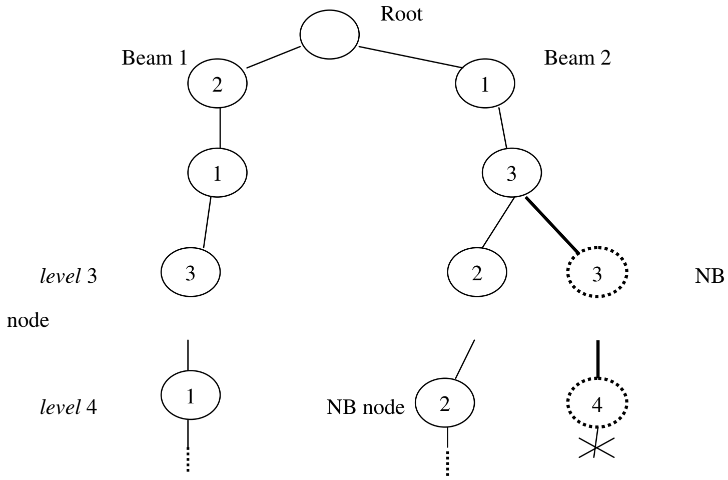


Fig. 6. The BS tree obtained by implementing BS-1. Z stands for the value of the objective function.



a. The schematic view of the backtracking procedure at *level* 2.



b. The schematic view of the backtracking procedure at *level* 3.

Fig. 7. The backtracking procedure invoked in the implementation of BS-4.

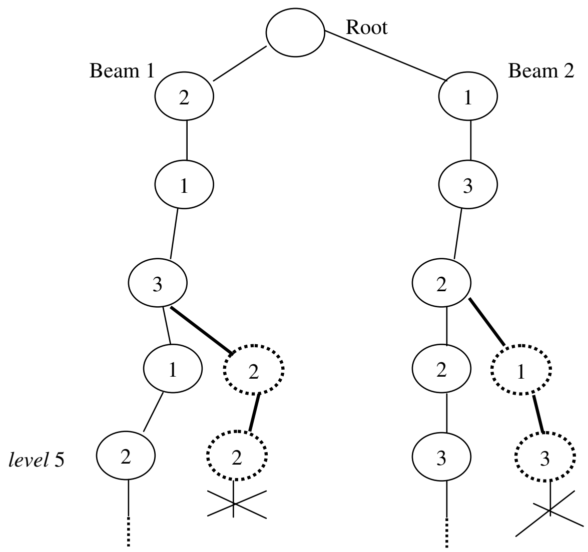


Fig. 8. The EOI procedure in the implementation of BS-4.

inferior at this level, it is backtracked by moving one level up, and generating the NB node, (i.e., product 3 in Fig. 7a). The NB node is further branched by selecting the best node (i.e., product 2 in Fig. 7a); whereas the original node is further sprouted by choosing the NB node at level 3. After the comparison of these two nodes, the newly generated node is found to be superior. Hence, Beam 2 progresses the search procedure with the new node. As the equivalency is observed at level 3, the same procedure is invoked for the inferior beam (i.e., Beam 2 in Fig. 7b) at this level.

In addition to the backtracking method, we apply the EOI procedure at a certain level which is level 5. We first choose the last products of Beam 1 and Beam 2, which are product 2 and product 3, respectively. Then we exchange the information between each beam in the following way: First we transfer the sequence of 2-2 from Beam 2 to Beam 1 since it is between the first and the last appearance of the last product (i.e., product 3) of Beam 2. This partial sequence is inserted to Beam 1 at level 3, at which the first appearance of product 3 is observed. Hence, a new solution with the sequence of 2-1-3-2-2 is obtained at level 5. This solution is compared with the original solution of Beam 1 that has the sequence of 2-1-3-1-2. Similarly, the sequence of 1-3-1 is transferred from Beam 1 to Beam 2, leading a new beam with a sequence of 1-3-2-1-3-1. Since the length of the new beam is greater than the current level, it is truncated, by which we obtain the sequence of 1-3-2-1-3. The result of the evaluation of the original beams and the newly generated beams reveals that the new beams are inferior. Hence, the procedure progresses with the original beams at level 5.

During the implementation of the algorithm, the equivalency is observed again at level 9. However it does not change the structure of the inferior beam (i.e., Beam 1) at this level. The resulting sequence is 1-3-2-2-3-4-2-3-1-2, with a CV value of 66.2. Hence, the CV is improved by 8.4%.

References

Abdou, S., Scordilis, M.S., 2004. Beam search pruning in speech recognition using a posterior probability-based confidence measure. Speech Communication 42, 409–428.

Alexouda, G., Paparrizos, K., 2001. A genetic algorithm approach to the product line design problem using the seller's return criterion: An extensive comparative computational study. European Journal of Operational Research 134, 165–178.

Bautista, J., Companys, R., Corominas, A., 1996. Heuristics and exact algorithms for solving the Monden problem. European Journal of Operational Research 88, 101–113.

Beraldi, P., Ruszczynski, A., 2005. Beam search heuristic to solve stochastic integer problems under probabilistic constraints. European Journal of Operational Research 167, 35–47.

Blum, C., 2005. Beam-ACO-hybridizing ant colony optimization with beam search: An application to open shop scheduling. Computers & Operations Research 32, 1565–1591.

Chang, Y., Matsuo, H., Sullivan, R.S., 1989. A bottleneck-based beam search for job scheduling in a flexible manufacturing system. International Journal of Production Research 27, 1949–1961.

Della Croce, F.D., T'kindt, V., 2002. A recovering beam search algorithm for the one-machine dynamic total completion time scheduling problem. Journal of The Operational Research Society 53, 1275–1280.

Della Croce, F.D., Ghirardi, M., Tadei, R., 2004. Recovering beam search: Enhancing the beam search approach for combinatorial optimization problems. Journal of Heuristics 10, 89–104.

Ding, F.Y., Cheng, L., 1993. An effective mixed-model assembly line sequencing heuristic for just-in-time production systems. Journal of Operations Management 11, 45–50.

Erel, E., Sabuncuoglu, I., Sekerci, H., 2005. Stochastic assembly line balancing using beam search. International Journal of Production Research 43, 1411–1426.

Esteve, B., Aubijoux, C., Chartier, A., T'kindt, V., 2006. A recovering beam search algorithm for the single machine just-in-time scheduling problem. European Journal of Operational Research 172, 798–813.

Forshed, J., Torgrip, R.J.O., Aberg, K.M., Karlberg, B., Lindberg, J., Jacobsson, S.P., 2005. A comparison of methods for alignment of NMR peaks in the context of cluster analysis. Journal of Pharmaceutical and Biomedical Analysis 38, 824–832.

Ghirardi, M., Potts, C.N., 2005. Makespan minimization for scheduling unrelated parallel machines: A recovering beam search approach. European Journal of Operational Research 165, 457–467.

Honda, N., Mohri, S., Ishii, H., 2003. Backtracking beam search applied to multi-objective scheduling problem. In: The Fifth Metaheuristics International Conference, Kyoto, pp. 1–6.

Jin, M., Wu, S.D., 2002. A new heuristic method for mixed model assembly line balancing problem. Computers and Industrial Engineering 44, 159–169.

Kim, K.H., Kim, K.Y., 1999. Routing straddle carriers for the loading operation of containers using a beam search algorithm. Computers and Industrial Engineering 36, 109–136.

Kim, K.H., Kang, J.S., Ryu, R.K., 2004. A beam search algorithm for the load sequencing of outbound containers in port container terminals. OR Spectrum 26, 93–116.

Lee, G.C., Woodruff, D.L., 2004. Beam search for peak alignment of NMR signals. Analytica Chimica Acta 513, 413–416.

Leu, Y., Huang, P.Y., Russell, R.S., 1997. Using beam search techniques for sequencing mixed-model assembly lines. Annals of Operations Research 70, 379–397.

Lim, A., Rodrigues, B., Zhang, X., 2006. Scheduling sports competitions at multiple venues-Revisited. European Journal of Operational Research 175, 171–186.

Lowerre, B.T., 1976. The HARPY speech recognition system, Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA.

Matsuda, T., Motoda, H., Yoshida, T., Washio, T., 2002. Mining patterns from structured data by beam-wise graph-based induction. In: Proceedings of the 5th International Conference on Discovery Science Discovery, vol. 2534, pp. 422–429.

McMullen, P.R., Tarasewich, P., 2005. A beam search heuristic method for mixed-model scheduling with setups. International Journal of Production Economics 96, 273–283.

Miltenburg, J., Sinnamon, G., 1992. Algorithms for scheduling multi-level just-in-time production systems. IIE Transactions 24, 121–130.

Monden, Y., 1983. Toyota Production System. Institute of Industrial Engineers, Norcross, GA, USA.

Ortmanns, S., Ney, H., 2000. Look-ahead techniques for fast beam search. Computer Speech and Language 14, 15–32.

Ow, S.P., Morton, T.E., 1988. Filtered beam search in scheduling. International Journal of Production Research 26, 35–62.

Pacciarelli, D., Pranzo, M., 2004. Production scheduling in a steelmaking-continuous casting plant. Computers and Chemical Engineering 28, 2823–2835.

Sabuncuoglu, I., Karabuk, S., 1998. A beam search-based algorithm and evaluation of scheduling approaches for flexible manufacturing systems. IIE Transactions 30, 179–191.

Sabuncuoglu, I., Bayiz, M., 1999. Job shop scheduling with beam search. European Journal of Operational Research 118, 390–412.

Sabuncuoglu, I., Bayiz, M., 2000. Analysis of reactive scheduling problems in a job shop environment. European Journal of Operational Research 126, 567–586.

Shahookar, K., Khamisani, W., Mazumder, P., Reddy, S.M., 1993. Genetic beam search for gate matrix layout. In: Proceedings of the 6th International Conference on VLSI Design, pp. 208–213.

Shayan, E., Al-Hakim, L., 2002. Beam search for sequencing point operations in flat plate manufacturing. Computers and Industrial Engineering 42, 309–315.

Tillmann, C., Ney, H., 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. Computational Linguistics 29, 97–133.

Valente, J.M.S., Alves, R.A.F.S., 2005. Filtered and recovering beam search algorithms for the early/tardy scheduling problem with no idle time. Computers and Industrial Engineering 48, 363–375.

Wang, J., 2002. A fuzzy project scheduling approach to minimize schedule risk for product development. Fuzzy Sets and Systems 127, 99–116.

Zeng, X., Martinez, T.R., 2002. Optimization by varied beam search in hopfield networks. In: Proceedings of the IEEE International Joint Conference on Neural Networks, pp. 913–918.

Zhou, R., Hansen, E.A., 2004. Breadth-first heuristic search. In: Proceedings of the 14th International Conference on Automated Planning and Scheduling.

Zhou, B., Xi, L., Cao, Y., 2005. A beam-search-based algorithm for the tool switching problem on a flexible machine. International Journal of Advanced Manufacturing Technology 25, 876–882.

Zhou, X., Zhong, M., 2005. Bicriteria train scheduling for high-speed passenger railroad planning applications. European Journal of Operational Research 167, 752–771.