Jena Research Papers in Business and Economics

On the Part Inventory Model Sequencing Problem: Complexity and Beam Search Heuristic

Malte Fliedner, Nils Boysen, Armin Scholl

20/2007

Jenaer Schriften zur Wirtschaftswissenschaft

Working and Discussion Paper Series
School of Economics and Business Administration
Friedrich-Schiller-University Jena

ISSN 1864-3108

Publisher:

Wirtschaftswissenschaftliche Fakultät
Friedrich-Schiller-Universität Jena
Carl-Zeiß-Str. 3, D-07743 Jena

www.jbe.uni-jena.de

Editor:

Prof. Dr. Hans-Walter Lorenz
h.w.lorenz@wiwi.uni-jena.de

Prof. Dr. Armin Scholl
armin.scholl@wiwi.uni-jena.de

www.jbe.uni-jena.de

On the Part Inventory Model Sequencing Problem: Complexity and Beam Search Heuristic

Malte Fliedner[a], Nils Boysen[a,*], Armin Scholl[b]

[a] *Universität Hamburg, Institut für Industrielles Management, Von-Melle-Park 5, D-20146 Hamburg,* {fliedner,boysen}@econ.uni-hamburg.de

[*] *Corresponding author, phone +49 40 42838-4640.*

[b] *Friedrich-Schiller-Universität Jena, Lehrstuhl für Betriebswirtschaftliche Entscheidungsanalyse, Carl-Zeiß-Straße 3, D-07743 Jena,* a.scholl@wiwi.uni-jena.de

Abstract

In many industries mixed-model assembly systems are increasingly supplied out of third-party consignment stock. This novel trend gives rise to a new short-term sequencing problem which decides on the succession of models launched down the line and aims at minimizing the cost of in-process inventory held by the manufacturer. In this work, we investigate the mathematical structure of this part oriented mixed-model sequencing problem and prove that general instances of the problem are NP-hard in the strong sense. Moreover, we develop a new Beam Search heuristic, which clearly outperforms existing solution procedures.

Keywords: Mixed-model assembly line; Sequencing; Consignment stock; Complexity proof; Beam Search

1 Introduction

The problem of optimally sequencing mixed-model assembly systems has been the subject of extensive research for more than four decades. Various exact and heuristic solution approaches have been developed for several well-known sequencing approaches like mixed-model sequencing (Thomopoulos, 1967; Tsai, 1995), car sequencing (Parello et al., 1986; Gagné et al., 2006) and level scheduling (Kubiak, 1993; Monden, 1998). An in-depth overview on model sequencing is provided by Boysen et al. (2007a). However, these

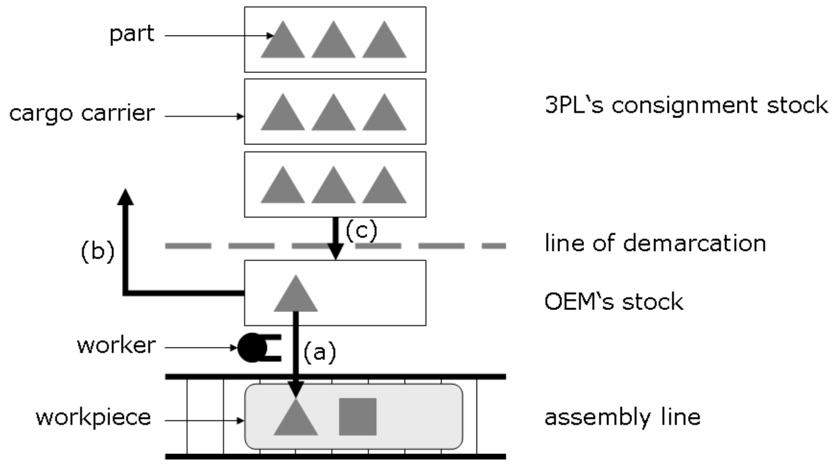

Figure 1: Schematic representation of the assembly line

traditional sequencing approaches are not sufficient to cover the recent industry trend of consignment stocks:

Nowadays, Original Equipment Manufacturers (OEMs) more and more reorganize their material supply by relying on third-party consignment stock, which serves the assembly line with required material. In such a setting, the structure of the sequencing problem is much different from traditional model sequencing approaches. The material is supplied just-in-time by cargo carriers of a fixed size from a consignment stock operated by a third-party logistics provider (3PL) adjacent to the line. In order to reduce the inventory held in possession by the OEM and to free valuable maneuvering space, a manufacturer in principle seeks to follow three simple policies, which are depicted in the schematic representation of the line in Figure 1:

(a) There is only a single cargo carrier per part in the possession of the OEM at a time from which a worker removes the required material part by part and assembles it into a workpiece which requires the respective product feature.

(b) Once the cargo carrier is emptied out by the worker, it is instantaneously removed from the station to free manoeuvering space. With regard to today's trend of decreasing vertical integration and, thus, an ever increasing number of parts to be assembled per station, the space at the assembly line is typically very scarce (see Klaempfl, 2006; Boysen et al., 2007c).

(c) A new cargo carrier is issued as late as possible, that is, only if the current inventory of a part is zero and the respective part is required again by a model in the production sequence. Once a new cargo carrier crosses the line of demarcation, which typically separates the OEM's inventory from the consignment stock, the parts contained therein are automatically charged to the OEM by an online billing system.

Obviously, the model sequence heavily influences the demand pattern of parts, so that model sequencing in a consignment stock setting aims at a demand pattern, which

P	set of parts (index p)
T	number of production cycles (index t)
M	set of models (index m)
b_{pm}	integer demand coefficients for part p and model m
d_m	demand of model m
c_p	inventory holding cost for storing a unit of part p during a cycle
G_p	capacity of the cargo carrier for part p
S_p	quantity of part p initially stored in OEM's stock
y_{pt}	number of cargo carriers for part p taken from consignment stock up to cycle t
x_{mt}	number of scheduled copies of model m up to cycle t
l_{pt}	quantity of part p in the OEM's stock stored during cycle t
r_{pt}	quantity of part p removed from actual cargo carrier up to cycle t
z_{pt}	total quantity of part p demanded by the model sequence up to cycle t

Table 1: Notation

minimizes OEM's inventory costs. For this purpose, Boysen et al. (2007b) introduce the so called part inventory model sequencing problem (PIMSP), whose mathematical structure is presented in Section 2. The paper on hand broadens the work of Boysen et al. (2007b) in two directions. First (Section 3), we prove NP-hardness (in the strong sense) for general instances of PIMSP, which was only conjectured in the aforementioned paper. For the solution of PIMSP the preceding paper proposes an exact Bounded Dynamic Programming approach suited for small instances and two heuristic approaches (Goal Chasing and an Ant Colony approach) for instances of real world size. The paper on hand describes a heuristic Beam Search procedure (Sections 4.1 and 4.2), which clearly outperforms the existing procedures (Section 4.3). Finally, Section 5 concludes the paper.

2 Problem statement

In a consignment stock setting, model sequencing seeks to reduce the in-process inventory of parts, where the number of units l_{pt} stored for part p during a cycle t can be calculated by determining the total number y_{pt} of issued cargo carriers of part p up to cycle t and the cumulative part usage, which in turn depends on the total number of scheduled model copies x_{mt} over all types m assigned up to cycle t. With the help of the notation summarized in Table 1, the PIMSP model contains of objective function (1) and constraints (2)-(8):

$$\text{(PIMSP) Minimize } C(X,Y,L) = \sum_{p \in P} (c_p \cdot \sum_{t=1}^{T} l_{pt}) \tag{1}$$

$$\sum_{m \in M} x_{mt} = t \qquad \forall\, t = 1, \ldots, T \tag{2}$$

$$x_{mT} = d_m \qquad \forall\, m \in M \tag{3}$$

$$\sum_{m \in M} x_{mt} \cdot b_{pm} + l_{pt} = y_{pt} \cdot G_p + S_p \qquad \forall\, p \in P;\, t = 1, \ldots, T \tag{4}$$

$$0 \leq x_{mt} - x_{mt-1} \leq 1 \qquad \forall\, m \in M;\, t = 2, \ldots, T \tag{5}$$

$$0 \leq y_{pt} - y_{pt-1} \leq 1 \qquad \forall\, p \in P;\, t = 2, \ldots, T \tag{6}$$

$$y_{pt} \in \mathbb{N}^0;\, l_{pt} \geq 0 \qquad \forall\, p \in P;\, t = 1, \ldots, T \tag{7}$$

$$x_{mt} \in \mathbb{N}^0 \qquad \forall\, m \in M;\, t = 1, \ldots, T \tag{8}$$

The objective function (1) minimizes the total cost of inventory summing up the quantities l_{pt} of all parts p stored in all cycles t each of which is weighted with the part-specific inventory holding cost factor c_p. Constraints (2) and (5) ensure that in each cycle t exactly one model copy is produced, whereas equations (3) enforce that the demand d_m of each model m is met at the end of the planning horizon. The balance equations (4) define the quantity l_{pt} stored per part p and cycle t as the difference between the overall number of issued units (number of issued carriers y_{pt} times carrier size G_p) plus initial stock S_p and the cumulative consumption of the part by previously scheduled model copies. Constraints (6) enforce the integer variables y_{pt} to monotonically increase over time.

3 Complexity of PIMSP

3.1 Restatement of PIMSP

First, we will equivalently restate the original problem formulation of PIMSP to ease the formalization of the proof. Note that the number of part units r_{pt} of part p which have been removed from the actual cargo carrier at time t can be calculated as follows:

$$r_{pt} = z_{pt} + G_p - S_p - \left\lfloor \frac{z_{pt} + G_p - S_p}{G_p} \right\rfloor \cdot G_p = (z_{pt} + G_p - S_p)\ mod\ G_p \tag{9}$$

where z_{pt} denotes the total number of part units consumed by the model sequence up to cycle t, i.e., $z_{pt} = \sum_{m \in M} x_{mt} \cdot b_{pm}$ and '*mod*' refers to the modulo division. If for a part p initial stock is positive ($S_p > 0$), it follows that there is a container for p at the beginning of the planning horizon from which $G_p - S_p$ units have already been removed. It holds for all t subsequent cycles within the horizon, that demands z_{pt} are additionally removed from this and all further containers, each of which has size G_p. The modulo operation thus yields the exact number of units removed from the current container in t, as all prior containers were emptied out completely, i.e., G_p units were removed respectively.

It follows that the actual number of stored units of part p at cycle t amounts to:

$$l_{pt} = (G_p - r_{pt}) \; mod \; G_p \tag{10}$$

where the additional modulo division ensures that whenever no part unit is required from the next cargo carrier in cycle t ($r_{pt} = 0$), its issuance is postponed, so that the current inventory is zero.

Equations (9) and (10) can now be used to rewrite the model formulation as follows.

$$\text{Minimize } C(X) = \sum_{t=1}^{T} \sum_{p \in P} c_p \cdot (G_p - r_{pt}) \; mod \; G_p \tag{11}$$

$$\text{s.t. } (2), (3), (5), (8)$$

$$r_{pt} = (z_{pt} + G_p - S_p) \; mod \; G_p \qquad \forall \, p \in P; \, t = 1, \ldots, T \tag{12}$$

$$z_{pt} = \sum_{m \in M} x_{mt} \cdot b_{pm} \qquad \forall \, p \in P; \, t = 1, \ldots, T \tag{13}$$

The rewritten objective function (11) is still minimizing part inventory cost, merely the current inventory is now calculated differently. Note, that r_{pt} and z_{pt} along with restrictions (12) and (13) are introduced to ease the presentation, so that the only variables really required in the restated formulation are the model assignments x_{mt}. The exact cycles in which new cargo carriers have to be issued (y_{pt}) can be easily determined for any given sequence by retrieving the actual cycles in which the number of removed units r_{pt} just exceeds a new multiple of G_p for any part p.

3.2 Proof of NP-hardness for PIMSP

On the basis of the restated model formulation, we will proof NP-hardness for the general version of PIMSP. For this purpose we show how to transform instances of the 3-Partition Problem to PIMSP. The 3-Partition Problem is well known to be NP-hard in the strong sense (see Garey and Johnson, 1979) and can be summarized as follows.

3-Partition Problem: Given $3q$ positive integers a_t ($t = 1, \ldots, 3q$) and a positive integer B with $B/4 < a_t < B/2$ and $\sum_{t=1}^{3q} a_t = qB$, does there exist a partition of the set $\{1, 2, \ldots, 3q\}$ into q sets $\{A_1, A_2, \ldots, A_q\}$ such that $\sum_{t \in A_i} a_t = B \quad \forall i = 1, \ldots, q$?

Transformation of 3-Partition to PIMSP: Consider an instance of PIMSP with two parts $P = \{1, 2\}$, $T = 3q$ production cycles and $\overline{M}$ models in the set $M = \{1, 2, \ldots, \overline{M}\}$ with demands $d_m \geq 1 \quad \forall m \in M$ and $\sum_{m \in M} d_m = 3q$. Let inventory holding cost equal one and initial inventories be zero ($c_p = 1, S_p = 0$ for $p = 1, 2$). The demand coefficients are

$$\begin{aligned} b_{1m} &= \alpha_m \\ b_{2m} &= B - \alpha_m \end{aligned} \qquad \forall m \in M, \tag{14}$$

where α_m are positive integer values such that $B/4 < \alpha_m < B/2$ and $\sum_{m \in M} \alpha_m \cdot d_m = q \cdot B$ and B is a positive integer.

Such a PIMSP instance can be derived in polynomial time from any instance of 3-Partition by grouping the set $\{1, 2, \ldots, 3q\}$ into $\overline{M}$ subsets $\{A_1^*, A_2^*, \ldots, A_{\overline{M}}^*\}$, so that all integers in such a subset are of the same size. That is, for each pair $t, j \in A_m^*$ of each subset $m \in M$, we have $a_t = a_j$, while $a_t \neq a_j$ is true for all pairs t, j from different subsets A_m^* and A_n^* with $m \neq n$. The demand coefficients for parts are then determined via $\alpha_m = a_t$ for all $t \in A_m^*, m \in M$ and model demands are equal to $d_m = |A_m^*|$ for all $m \in M$.

Let the size of the two cargo carriers further be $G_1 = B$ and $G_2 = 2B$. When replacing r_{pt} in (11) by the expression defined in (12) and considering the assumptions on c_p and S_p, we can rewrite the objective function as follows (notice that the equivalence $(x + y)\ mod\ y \equiv x\ mod\ y$ holds for integers x and y):

$$\begin{aligned} C &= \sum_{t=1}^{T} (B - z_{1t}\ mod\ B)\ mod\ B + \sum_{t=1}^{T} [2B - z_{2t}\ mod\ (2B)]\ \ mod\ (2B) \\ &= \sum_{t=1}^{T} (B - z_{1t}\ mod\ B)\ mod\ B + [2B - z_{2t}\ mod\ (2B)]\ \ mod\ (2B) \qquad (15) \end{aligned}$$

We will now show that the instance of 3-Partition is a YES-instance if and only if there exists a solution to the respective PIMSP instance with $C \leq 3qB$.

The rewritten objective function in (15) can be rearranged to $C = \sum_{t=1}^{3q} C_t$ with $C_t = (B - z_{1t}\ mod\ B)\ mod\ B + [2B - z_{2t}\ mod\ (2B)]\ \ mod\ (2B)$. Due to the structure of demand coefficients it further holds that in any cycle t the cumulated requirement of both parts together is a integer multiple of B, i.e., we get

$$z_{1t} + z_{2t} = t \cdot B \quad \text{for } t = 1, \ldots, T, \qquad (16)$$

which follows directly from the fact that $b_{1m} + b_{2m} = \alpha_m + B - \alpha_m = B$ for each $m \in M$. As further $b_{pm} > 0$ is true for all $p \in \{1, 2\}$ and $m \in M$, part consumption is strictly monotonically increasing:

$$z_{p,t+1} > z_{pt} \quad \text{for } p \in \{1, 2\} \text{ and } t = 1, \ldots, T - 1 \qquad (17)$$

In dependence of the actual size of z_{2t} it can be shown that:

$$C_t = \begin{cases} 0 & \text{, if } z_{2t}\ mod\ (2B) = 0 \\ 2B & \text{, if } 0 < z_{2t}\ mod\ (2B) < B \\ B & \text{, otherwise} \end{cases} \quad \forall t = 1, \ldots, T \qquad (18)$$

In the following, these three cases are treated separately.

Case 1 ($z_{2t}\ mod\ (2B) = 0$): Since z_{2t} is a multiple of $2B$ it follows from (16) that z_{1t} is a multiple of B, too, and consequently $z_{1t}\ mod\ B = 0$. It directly follows that

$C_t = B \ mod \ B + 2B \ mod \ (2B) = 0$.

Case 2 ($0 < z_{2t} \ mod \ (2B) < B$): Due to $z_{2t} \ mod \ (2B) < B$, the following equivalence holds: $z_{2t} \ mod \ (2B) \equiv z_{2t} \ mod \ B$. So, we get $[2B - z_{2t} \ mod \ (2B)] \ mod \ (2B) \equiv [2B - z_{2t} \ mod \ B] \ mod \ (2B) \equiv 2B - z_{2t} \ mod \ B$. As at the same time the condition of case 2 means that z_{2t} is no multiple of B then due to (16) also z_{1t} is no multiple of B. From these preconditions it follows that $C_t = B - z_{1t} \ mod \ B + 2B - z_{2t} \ mod \ B = 3B - (z_{1t} \ mod \ B + z_{2t} \ mod \ B) = 3B - B = 2B$.

Case 3 ($B \leq z_{2t} \ mod \ (2B) < 2B$): As $z_{2t} \ mod \ (2B)$ is bounded from above by $2B$ this is the only case which remains to be investigated. Due to $z_{2t} \ mod \ (2B) \geq B$ we get $z_{2t} \ mod \ (2B) \equiv B + z_{2t} \ mod \ B$ and $(2B - B - z_{2t} \ mod \ B) \ mod \ (2B) \equiv B - z_{2t} \ mod \ B$, so that $C_t = (B - z_{1t} \ mod \ B) \ mod \ B + B - z_{2t} \ mod \ B$. Now, we distinguish between two sub-cases:

(a) If $z_{2t} \ mod \ (2B) = B$ then z_{2t} is a multiple of B and due to (16), also z_{1t} is a multiple of B so that $C_t = (B - z_{1t} \ mod \ B) \ mod \ B + B - z_{2t} \ mod \ B = B \ mod \ B + B - 0 = B$.

(b) If z_{2t} is not a multiple of B then due to (16) also z_{1t} is not a multiple of B, so that $0 < z_{1t} \ mod \ B < B$ and $C_t = B - z_{1t} \ mod \ B + B - z_{2t} \ mod \ B = 2B - (z_{1t} \ mod \ B + z_{2t} \ mod \ B) = 2B - B = B$.

It can further be shown that any solution to such an instance will at least have q slots where $0 < z_{2t} \ mod \ (2B) < B$ and at maximum q slots where $z_{2t} \ mod \ (2B) = 0$. The latter follows directly from the fact that $z_{2T} = 2qB$ and further (17) holds, so that only q multiples of $2B$ can be met. The former is due to the fact that because of (14) all demand coefficient of part 2 are smaller than B. It follows that in the first cycle and subsequently in any cycle where z_{2t} just exceeds a new multiple of $2B$ the difference between this multiple and z_{2i} has to be smaller than B so that $z_{2t} \ mod \ (2B) < B$. This has to occur q times to reach $z_{2T} = 2qB$. As further due to (18) the remaining q cycles show at least a cost of B, it follows that $C_{LB} = 3qB$ is a lower bound on the objective value.

We can transform the solution of any YES-instance of 3-Partition to a solution of PIMSP by simply arranging the sets A_i in an arbitrary order and replacing each element $j \in A_i$ by the number of the model m to which it is assigned in the PIMSP-instance, i.e., $j \in A_m{}^*$. Irrespective of the internal order within sets A_i, the restrictions imposed on α_m in (14) will lead to a production sequence where $z_{11} < B/2$ and $B/2 < z_{21} < B$. Applying (18) we get $C_1 = 2B$. Furthermore, $B/2 < z_{12} < B$ and thus $B < z_{22} < 2B$ so that $C_2 = B$. Finally, due to the definition of the partition we have $z_{13} = B$, so that $z_{23} = 2B$ and $C_3 = 0$, which sums up to inventory cost of $3B$ for the first three cycles. Note that after cycle 3 the total consumption of parts equals B and $2B$ for products 1 and 2, respectively, and the inventories are thus $l_{13} = l_{23} = 0$. As the same has to hold for all subsequent triplets, the argumentation can be continued exactly q times, resulting to a total objective value of $C = 3qB$.

Conversely, let us consider that a solution with an objective value $C \leq 3qB$ actually exists. As was argued above, such an objective value can only be realized if z_{2t} becomes a multiple of $2B$ exactly q times, which due to (16) means that z_{1t} is also a multiple of B for these slots. Now let us consider the part consumption at the beginning of such a sequence. Because of (14) it holds that $0 < z_{12} < B$ and $z_{14} < 2B$, so that at the same time due to (16) $B < z_{22} < 2B$ and $z_{24} > 2B$. In other words, the cumulated consumption of part 2 is always strictly lower than $2B$ up to slot 2, but strictly larger than $2B$ after slot 4. If the consumption of part 2 is thus not equal to $2B$ at slot 3, then due to (17) the first multiple of $2B$ cannot be met anymore and because of (18) the objective value needs to increase by at least B, so that $C \geq (3q+1)B$. It follows that for a solution with $C \leq 3qB$ it has to hold that $z_{23} = 2B$ and thus $z_{13} = B$. As this means that $l_{13} = l_{23} = 0$, the argumentation can be continued in the same fashion for all subsequent triplets, so that $z_{1,3 \cdot i} = i \cdot B$ for all $i = 1, \ldots, q$, which immediately yields the required partition.

The answer to the question of whether there exists a solution to an instance of 3-Partition is thus YES, if and only if there exists a solution with $C \leq 3qB$ for the corresponding instance of PIMSP. As 3-Partition is NP-hard in the strong sense, so is the general version of PIMSP.

4 A novel Beam Search procedure

4.1 General description

As the problem was shown to be NP-hard in the strong sense, heuristic solution approaches are required to solve problem instances of real-world size. Beam Search is a truncated breadth-first tree search heuristic and was first applied to speech recognition systems by Lowerre (1976). Ow and Morton (1988) systematically study the performance of Beam Search compared to other well-known heuristics for two scheduling problems. Since then, Beam Search was utilized within multiple fields of application and many extensions have been developed, e.g., stochastic node choice (Wang and Lim, 2007) or hybridization with other meta-heuristics (Blum, 2005), so that Beam Search turns out to be a powerful meta-heuristic applicable to many real-world optimization problems. A review on these developments is provided by Sabuncuoglu et al. (2008).

Like other breadth-first search procedures, Beam Search relies on a tree representation of the solution space. Unlike a breadth-first version of Branch&Bound, Beam Search restricts the number of nodes per stage to be further branched to a promising subset, which is determined by heuristic choices in a multi-stage filtering process. A schematic representation of Beam Search is depicted in Figure 2.

Starting with the root node of stage 0, all nodes (set V^1) of stage 1 are constructed and form the set B^1 of branched nodes. Then, the multi-stage filtering process of Beam Search starts to identify promising nodes of stage 1. First, a rough and computational inexpensive measure is applied within *rough filtering*. This measure, i.e. a priority rule or lower bound on the remaining path from the current node to the final stage, assigns a priority value to each node within set B^1, so that the first FW nodes with regard to this

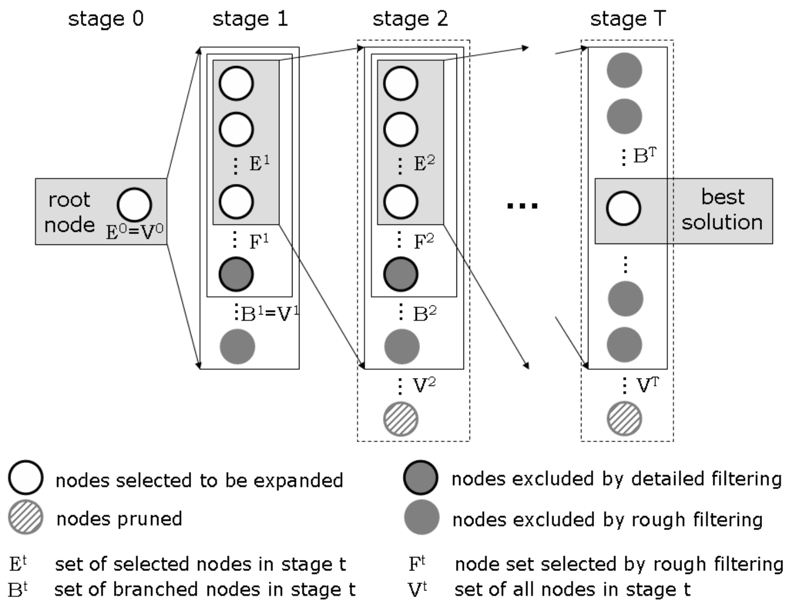

Figure 2: Schematic representation of Beam Search

priority value are chosen to form the set F^1, where $FW = |F^1|$ is a control parameter called the *filtered beam width*. In the following step, so called *detailed filtering* is applied to further reduce node set F^1 to set E^1, which contains all nodes to be further branched in the succeeding stage. To choose the respective number of $|E^1| = BW$ nodes, where BW is a control parameter called *beam width*, a more detailed and time-consuming inspection of nodes is applied. Typically, a more sophisticated lower bound procedure is utilized or even upper bound solutions are constructed by completing partial solutions (represented by the respective node) with a simple myopic priority rule based heuristic (e.g. Ow and Morton, 1988). Only the selected nodes contained in set E^1 are branched to build the set B^2 of branched nodes in stage 2, which is only a small subset of all possible nodes V^2. These steps are repeated until the final stage T is reached, where the best solution out of the set B^T of constructed nodes is returned as the result of the Beam Search procedure.

4.2 A Beam Search procedure for PIMSP

To apply the general procedure of Beam Search in a specific domain the following three components must be specified with regard to the respective problem: (i) the graph structure and the measures for (ii) rough filtering as well as (iii) detailed filtering. In the following, we describe these specifications for PIMSP:

Graph structure: The most obvious graph structure to be applied for sequencing problems like PIMSP is to let nodes of a stage t represent partial sequences π of models up to sequence position t. We call the graph structure resulting from such an explicit enumeration scheme a *sequence based graph*. Thus, at every node of the resulting tree at

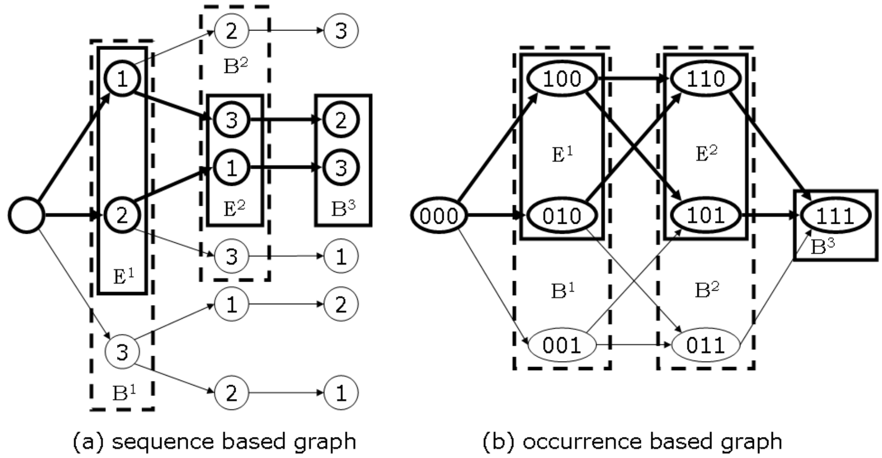

Figure 3: Alternative graph structures

most $|M|$ nodes are to be branched, one for every model which still needs to be scheduled. However, there exists a more compact graph representation of PIMSP, where a node represents the number of occurrences of each single model up to stage t. In such an implicit enumeration scheme, a node of stage t represents multiple subsequences of models, i.e. all subsequences (filled up to sequence position t) which share the same number of model occurrences irrespective of their exact order. Consequently, after branching into at most $|M|$ additional nodes from each node of a stage t (like above) the resulting node set B^{t+1} is to be consolidated by deleting duplicate nodes with identical model occurrences. In such an *occurrence based graph* subsequences of models are represented by the different paths in the graph leading to a node. Figure 3 depicts both alternative graph structures for an example with 3 models with a demand of one copy each.

An occurrence based graph can be applied whenever the contribution to the objective value of a partial solution represented by a node at a current stage t, exclusively depends on the occurrence of models instead of their exact partial sequence. This is the case for PIMSP because inventory cost of cycle t only depends on the given initial inventories S_p of part p, the cumulated production quantities X_{tim}, where X_{tim} denotes the number of occurrences of model m of a node i in stage t and the size of the cargo carrier G_p. The more compact occurrence based graph can be advantageous since the same node implicitly represents multiple subsequences, which are all evaluated in the process. In the sequence based graph each node merely represents a unique subsequence, so that with identical beam width BW less sequences are evaluated. This relationship becomes obvious in the example of Figure 3, where filtered beam width FW and beam width BW are assumed to be 2. If the sequence based graph (a) is applied only 2 sequences are evaluated, whereas the occurrence based graph (b) allows for an evaluation of 3 sequences. The evaluation of the advantage when applying the occurrence based graph is part of our computational study in Section 4.3.

Rough filtering measure: To select a number of FW (filtered beam width) nodes out of the set B^t into node set F^t, we simply calculate the actual contribution of a partial solution to the objective value. This contribution for an actual node (t, i) with cumulated production quantities X_{tim} per model m can be calculated as follows:

The produced quantities of all models up to cycle t in a state (t, i) directly determine the cumulative demands D_{tip} for all parts p:

$$D_{tip} = \sum_{m \in M} X_{tim} \cdot b_{pm} \qquad \forall\, p \in P \tag{19}$$

The inventories I_{tip} of the parts $p \in P$ during a cycle t in state (t, i) are easily derived by (20), because they are either units from initial stock S_p not consumed by cumulated demand D_{tip} or residual units out of newly issued cargo carriers of size G_p. The special case $I_{tip} = 0$ arises when the carrier has been emptied at the beginning of t or was already empty and no unit of p has been required in cycle t.

$$I_{tip} = \begin{cases} S_p - D_{tip}, & \text{if } S_p \geq D_{tip} \\ 0, & \text{else if } (D_{tip} - S_p) \bmod G_p = 0 \\ G_p - (D_{tip} - S_p) \bmod G_p, & \text{otherwise} \end{cases} \qquad \forall\, p \in P \tag{20}$$

Because the state (t, i) directly determines the quantities stored for each part $p \in P$, the corresponding node can be assigned with a unique partial objective value of R_{ti} equal to the inventory holding cost at cycle t as follows:

$$R_{ti} = \sum_{p \in P} c_p \cdot I_{tip} \qquad \forall\, t = 0, \ldots, T;\ i \in V_t \tag{21}$$

Our rough filtering selects the best FW nodes with regard to the partial objective values R_{ti} which form the set F^t and are further evaluated by detailed filtering.

Detailed filtering measure: Our detailed filtering procedure chooses BW (beam width) nodes out of node set F^t. Only these BW remaining nodes are stored in set E^t and are considered for further branching. As a measure to prioritize single nodes (t, i) we complete the partial solution represented by the respective node of stage t and determine an upper bound solution. To do so, a simple myopic priority rule based approach is applied.

At each remaining decision point $\tau = t + 1, \ldots, T$ only the set of possible alternatives POS_τ is relevant, which covers all models m whose demand is not satisfied by the partial solution of node (t, i) and preceding sequencing decisions between $t + 1$ and $\tau - 1$. Let $D_p(\tau, m) = \sum_{m' \in M} X_{tim} \cdot b_{pm'} + \sum_{t'=t+1}^{\tau-1} b_{p\pi_{t'}} + b_{pm}$ denote the cumulative demand for units of type p provided that model $m \in POS_\tau$ is assigned to the current decision point τ Then, for each model $m \in POS_\tau$ a priority value $f(\tau, m)$ has to be determined (see (20)):

$$f(\tau,m)=\sum_{p\in P} c_p\cdot\begin{cases}(S_p-D_p(\tau,m)), & \text{if } S_p\geq D_p(\tau,m)\\ 0, & \text{else if } (D_p(\tau,m)-S_p) \bmod G_p=0\\ (G_p-(D_p(\tau,m)-S_p)) \bmod G_p, & \text{otherwise}\end{cases} \tag{22}$$

Finally, with these priority values on hand, a greedy choice assigns the best model available to the sequencing position τ:

$$\pi_\tau = \text{argmin}_{m\in POS_\tau}\{f(\tau,m)\} \tag{23}$$

Then, τ is incremented and choices are repeated until model vector π is completely filled. In such a manner, all nodes of set F^t are completed to feasible solutions, and the best BW nodes are further branched in the next stage.

Finally, when the last stage is reached the best solution value is returned as the solution of our Beam Search approach for PIMSP.

4.3 Computational study

For our computational study we apply the 1458 test instances generated by Boysen et al. (2007b), which are downloadable under `www.assembly-line-balancing.de`. The overall test bed is subdivided into small (486 instances with a number of production cycles ranging between 10 and 20), medium (486 instances with 25-35 cycles) and large (486 instances with 100-300 cycles) instances. The methods described above have been implemented in Visual Basic.NET (Visual Studio 2003) and run on a Pentium IV, 1,800MHz PC, with 512MB of memory, which is the same configuration applied by Boysen et al. (2007b). Our computational study ought to answer the following three questions:

(i) Which parameter setting constitutes a reasonable compromise between solution quality and solution time for the Beam Search approach?

(ii) Does the Beam Search approach which relies on the occurrence based graph (BS^o) outperform its counterpart relying on the sequence based graph (BS^s)?

(iii) And finally, which performance does the Beam Search procedure show compared to the heuristic approaches provided by Boysen et al. (2007b)?

First, question (i) on a reasonable setting of control parameters FW (filtered beam width) and BW (beam width) is investigated. For this purpose, we vary these parameters in the following ranges: $FW \in \{10, 15, 20, 25, 30, 35, 40\}$ and $BW \in \{5, 10, 15, 20, 25\}$. For any feasible combination of control parameters (note that $FW \geq BW$ has to hold) we solve the set of small problem instances and relate the solution performance obtained to the respective control parameter values. These results are depicted in Figure 4.

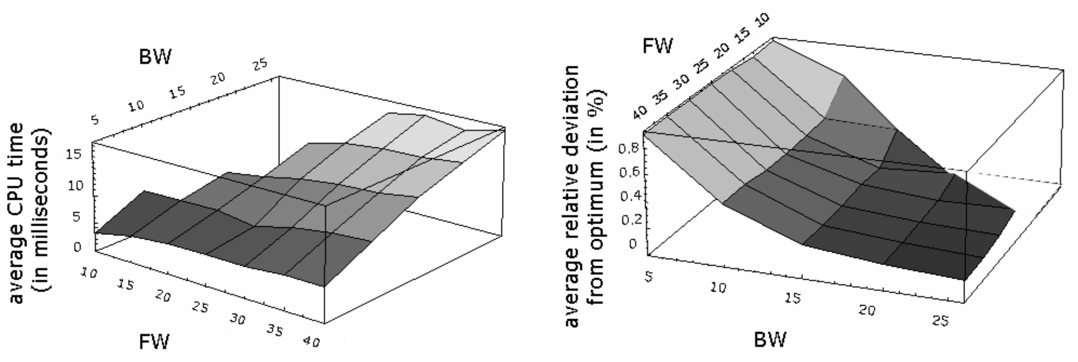

Figure 4: Solution time and quality in dependence of control parameters

Figure 4 displays that the solution time roughly increases in a linear manner if both BW and FW rise. However, this increase is smaller with regard to a rising FW than BW, since the completion of FW partial solutions by the myopic priority rule approach during detailed filtering is computational inexpensive when compared to the more complex branching process of BW nodes and the subsequent consolidation of duplicate nodes. On the other hand, with rising FW and BW the solution quality increases. This result is not astounding, because with increasing control parameter values a larger part of the complete solution graph is explored. In view of these results, we apply a parameter setting of $FW = 35$ and $BW = 20$ for the following computational tests, which turns out to be a reasonable choice to level the trade-off between solution quality and solution time.

To answer questions (ii) and (iii), we solve all 1458 instances with our novel Beam Search approaches BS^s basing on the sequence based graph and BS^o (occurrence based graph) and compare their results with the so-called Goal Chasing (GC) procedure, which is a simple myopic priority rule based heuristic, and the Ant Colony approach (ANT) provided by Boysen et al. (2007b). Table 2 displays the aggregated results over all small instances, where the results are compared in relation to optimal solution values. For the medium and large test instances these optimal solutions are unknown, so that Table 3 lists the aggregated results for these two instance sets compared to the results of a lower bound procedure (LB) introduced by Boysen et al. (2007b).

measure	GC	ANT	BS^s	BS^o
number of optimal solutions	89	323	340	435
average relative deviation from optimum in %	13.03	1.03	1.10	0.18
maximum relative deviation from optimum in %	123.08	11.2	30.0	5.42
average absolute deviation from optimum	17.3	1.9	1.59	0.36
maximum absolute deviation from optimum	246	44	33	22
average CPU-seconds	<0.1	0.53	0.009	0.014

Table 2: Results aggregated over all 486 small instances

measure	GC	ANT	BS^s	BS^o
average relative deviation from LB in %	22.72/26.68	9.37/16.28	7.61/9.30	5.21/6.90
maximum relative deviation from LB in %	120.3/148.7	70.4/69.4	85.0/60.37	55.6/50.7
average absolute deviation from LB	52.3/852.3	22.9/588.2	19.9/322.6	13.8/242.6
maximum absolute deviation from LB	528/8123	279/5613	263/3216	221/2554
average CPU-seconds	<0.1/<0.1	1.09/40.92	0.01/0.73	0.03/3.36

Legend: medium/big instances

Table 3: Results aggregated over all medium and large instances

These results reveal that BS^o easily outperforms all other approaches by far. For instance, BS^o solves 435 small instances to optimality with an average relative deviation from the optimum of merely 0.18%. This is considerably better than the alternative Beam Search approach BS^s (1.1%) and the heuristic approaches provided by Boysen et al. (2007b) (GC=13.03% and ANT=1.03%). Moreover, BS^o is also considerably faster than ANT. Analogously, BS^o outperforms all other approaches when solving medium and large test instances (see Table 3). Only with regard to the solution time BS^s is slightly superior, which can be explained by the fact that the sequence based graph does not require a check for duplicate nodes. This consolidation step at each stage of the graph, which is inevitable for the occurrence based graph, slows down BS^o. However, BS^o is about 10 times faster than ANT and yields a considerably better solution quality.

5 Conclusion and Future Research

In this work, it was shown that general instances of PIMSP are NP-hard in the strong sense. Although from a practical point of view, average algorithmic performance is more conclusive with regard to the applicability of particular solution methods, the theoretic computational complexity of a problem provides valuable insights with regard to the choice of suited algorithms and constitutes a decisive step in understanding the problem's structure. For a problem which is NP-hard in the strong sense, the existence of an exact algorithm with even pseudo-polynomial time complexity is highly unlikely, so that the development of specialized heuristic approaches seems meaningful in order to solve problem instances of real-world size. An appropriate Beam Search heuristic was additionally developed. A comprehensive computational study showed that this approach clearly outperforms existing solution procedures.

Future research could investigate the question of whether special PIMSP-instances, consisting exclusively of zero-one demand coefficients ($b_{pm} \in \{0,1\}$), are also NP-hard in the strong sense. Such a special structure of bills of material exists in some industrial cases (see Cakir and Inman, 1993; Boysen et al., 2007b), so that a respective proof would be a valuable contribution. Moreover other exact and heuristic algorithms could be developed to further enhance the solution performance.

References

[1] Blum C (2005) Beam-ACO – Hybridizing ant colony optimization with beam serach: An application to open shop scheduling, Computers & Operations Research 32: 1565–1591.

[2] Boysen N, Fliedner M, Scholl A (2007a) Sequencing mixed-model assembly lines: Survey, classification and model critique, European Journal of Operational Research (to appear).

[3] Boysen N, Fliedner M, Scholl A (2007b) Sequencing mixed-model assembly lines to minimize part inventory cost, OR Spectrum (to appear).

[4] Boysen N, Fliedner M, Scholl A (2007c) Level scheduling of mixed-model assembly lines under storage constraints, International Journal of Production Research (to appear).

[5] Cakir A, Inman RR (1993) Modified goal chasing for products with non-zero/one bills of material, International Journal of Production Research 31: 107–115.

[6] Garey MR, Johnson DS (1979) Computers and intractability: A guide to the theory of NP-completeness, Freeman, New York.

[7] Lowerre BT (1976) The HARPY speech recognition system, Ph.D. thesis, Carnegie-Mellon University, U.S.A., April.

[8] Ow PS, Morton TE (1988) Filtered beam search in scheduling, International Journal of Production Research 26: 297–307.

[9] Sabuncuoglu I, Gocgun Y, Erel E (2008) Backtracking and exchange of information: Methods to enhance a beam search algorithm for assembly line scheduling, European Journal of Operational Research 186: 915–930.

[10] Wang F, Lim A (2007) A stochastic beam search for the berth allocation problem, Decision Support Systems 42: 2186–2196.