

Kimms, Alf

**Working Paper — Digitized Version**

## Minimal investment budgets for flow line configuration

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 470

**Provided in Cooperation with:**

Christian-Albrechts-University of Kiel, Institute of Business Administration

*Suggested Citation:* Kimms, Alf (1998) : Minimal investment budgets for flow line configuration, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 470, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/149068>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*

Manuskripte  
aus den  
Instituten für Betriebswirtschaftslehre  
der Universität Kiel

No. 470

**Minimal Investment Budgets  
for Flow Line Configuration**

A. Kimms



Manuskripte  
aus den  
Instituten für Betriebswirtschaftslehre  
der Universität Kiel

No. 470

**Minimal Investment Budgets  
for Flow Line Configuration**

A. Kimms

March 1998

Alf Kimms

Lehrstuhl für Produktion und Logistik, Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel, Olshausenstr. 40, 24118 Kiel, Germany

email: [Kimms@bwl.uni-kiel.de](mailto:Kimms@bwl.uni-kiel.de)

URL: <http://www.wiso.uni-kiel.de/bwlinstitute/Prod>

<ftp://ftp.wiso.uni-kiel.de/pub/operations-research>

## Abstract

A flow line consists of a sequence of work places (or stations) through which one or more products (or models) move one-way in order to be processed. Each model requires specific operations which must be performed in a predefined order. To be able to do so, the stations must be equipped with machines, robots, and workers having a certain skill such that it is guaranteed that each model passing through the system can be completely processed. The number of stations and the equipment of these stations is called the configuration of the flow line. In this paper we deal with the  $\mathcal{NP}$ -hard problem of finding a configuration such that the net present value of cash outflows for installing and maintaining the flow line is minimized. As a special case, minimizing the number of stations is treated as well. Lower bounds are derived using column generation. Also, two heuristics are presented. One heuristic is based on the result of the column generation procedure while the other is adapted from the so-called majority merge heuristic. A computational study proves that the feasible solution obtained on the basis of column generation requires a decidedly lower investment budget.

**Keywords:** Flow line configuration, investment budget, net present value, shortest common supersequence, column generation

## 1 Flow Line Configuration

Finding a flow line configuration is a long-term decision problem which usually defines the production capabilities for several years. For small to medium sized firms, the financial burden of installing a flow line is immense. The investment not only includes the price for the acquisition of men and machines, but also covers future operating and maintaining costs (e.g., expenses for wages and repair).

To discuss the problem, assume that we have  $E$  different types of station equipments. For each equipment  $e \in \{1, \dots, E\}$  we have an estimate  $c_e > 0$  for the net present value (NPV) of the investment (cash outflow) per equipment. For example, a particular type of equipment could be “2 workers with a certain skill”. Furthermore, we plan to manufacture  $J$  different models on the flow line. Without loss of generality, let us suppose that we have to perform  $L$  operations for each model  $j \in \{1, \dots, J\}$ . Let  $e_{ji}$  denote the equipment that is needed to perform the  $i$ -th operation of model  $j$ .

A small example will help to illustrate the problem. Consider an instance with  $J = 3$  models. Each model requires  $L = 5$  operations. Furthermore, let us assume to have  $E = 3$  types of station equipment. Table 1 shows which operations need what equipment. For example, the third operation of model 2 needs equipment of type 1.

Three feasible solutions for this instance are given in Table 2. While configuration 1 consists of 10 stations, configurations 2 and 3 consist of 8 stations only which is, by the way, the minimal length. To install configuration 1 one must invest in 5 equipments of type 1, 3 of type 2, and 2 of type 3. Configuration 1 is dominated by configuration 2 in terms of the length of the flow line, and it is dominated by configuration 3 in terms of the investment to make. The question whether to prefer configuration 2 or 3 depends on the NPV's of the investment alternatives.

In summary we find that two objectives may determine what is a desirable flow line configuration. First, one may wish to keep the length of the flow line as short as possible. Given the speed with which models move through the system, this minimizes the lead time

$e_{ji}$	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
$j = 1$	1	2	1	3	2
$j = 2$	2	3	1	2	1
$j = 3$	3	1	2	3	1

Table 1: Equipment Requirements

Stations:	1	2	3	4	5	6	7	8	9	10
Configuration 1:	1	2	1	3	2	1	2	1	3	1
Model 1:	•	•	•	•	•					
Model 2:		•		•		•	•	•		
Model 3:				•		•	•		•	•
Configuration 2:	3	1	2	3	1	3	2	1		
Model 1:		•	•		•	•	•			
Model 2:			•	•	•		•	•		
Model 3:	•	•	•	•	•					
Configuration 3:	1	2	3	1	2	3	1	2		
Model 1:	•	•		•		•		•		
Model 2:		•	•	•	•		•			
Model 3:			•	•	•	•	•			

Table 2: Feasible Configurations

(defined as the time between entering and leaving the flow line). Such objective may be important in a make-to-order environment where short lead time helps to keep inventory levels low which in turn reduces the opportunity cost of capital, because customer orders can be fulfilled fast without large inventories. Second, one may wish to minimize the investment to make. This is important especially for small and medium sized firms which usually face low budgets. It is generally important for all firms, because low production costs allow low product prices which is an important marketing instrument in a competitive environment. Note, shorter flow lines do not necessarily mean lower investments. Compare the configurations 1 and 2 in the example above. If equipment of type 3 is very expensive and equipment of type 1 is fairly cheap, the configuration 1 may dominate configuration 2 in terms of the investment to make though specifying a longer flow line.

In this paper we will deal with the problem of minimizing the investment to make. Remarkable to note, this covers the problem of minimizing the length of the flow line as well, because choosing  $c_e = 1$  for all  $e \in \{1, \dots, E\}$  reveals the minimal length problem to be a special case of the minimal investment problem. As a solution procedure we propose column generation where the subproblem reveals to be a shortest path problem in an acyclic graph.

The existing literature is restricted to the special case  $c_e = 1$  for all  $e \in \{1, \dots, E\}$  and refers to it as the shortest common supersequence problem. It has been proven that this problem is  $\mathcal{NP}$ -hard [11, 13, 14, 19] even for  $E = 2$ . Thus, the minimal investment problem must be  $\mathcal{NP}$ -hard, too. Heuristics are presented in [2, 4, 6, 5, 9]. Optimal

procedures are described in [5].

Somehow related to our problem is the assembly line balancing problem (see, for instance, [7, 8, 10, 17, 18, 20]). In both problems operations are to be assigned to stations and a common goal is to minimize the number of stations again. In the assembly line balancing problem, however, the notion of a station means a working area rather than a particular type of equipment. Operations are to be assigned to these working areas subject to some side constraints (e.g., the sum of processing times of the operations to be executed at a station must not exceed the cycle time of the assembly line). The question of what equipment is needed to perform certain operations plays no role in assembly line balancing. Cost and profit oriented objectives have almost not been considered for assembly line balancing so far. A very few exceptions exist for the single-model case (see [12, 15, 16]). A multi-model case is described in [3]. Investment decisions have not been treated at all in the context of assembly line balancing.

We proceed as follows: Next, in Section 2 we present a combinatorial programming model to give a precise description of the problem to be attacked. Section 3 is devoted to find lower bounds by column generation. In Section 4 we derive upper bounds. A computational study in Section 5 tests the performance of the presented methods. Section 6 finishes the paper with some concluding remarks.

## 2 Model Formulation

To couch the minimal investment flow line configuration problem mathematically, we need to introduce some notation first. The parameters are:

- $E$  : the number of equipment types to be considered;
- $J$  : the number of models to be produced;
- $L$  : the number of operations per model;
- $P$  : an upper bound on the number of stations (=positions), e.g.,  $P = J \cdot L$ ;
- $c_e$  : NPV of the investment to make for installing equipment  $e$  once;
- $M_e$  : the set of operations which require equipment of type  $e$ .

To identify the operations we use  $(j, i)$  to denote the  $i$ -th operation of model  $j$ . That is, operations are considered to be distinct although they may require the same type of equipment. Furthermore, we assume that  $M_e \cap M_{e'} = \{\}$  for  $e \neq e'$ , i.e. operations require exactly one type of equipment. This implies that  $\sum_{e=1}^E |M_e| = J \cdot L$ . Once more we like to point out that  $c_e = 1$  for  $e \in \{1, \dots, E\}$  covers the case of minimizing the length of the flow line.

To describe the decision we have to make, we use the following binary decision variables:

- $x_{jip}$  : 1, if operation  $(j, i)$  is performed at position  $p$  (0, otherwise);
- $z_{ep}$  : 1, if equipment of type  $e$  is installed at position  $p$  (0, otherwise).

The problem can now be described formally.

$$\min \sum_{e=1}^E \sum_{p=1}^P c_e z_{ep} \tag{1}$$

subject to

$$\sum_{p=1}^P x_{jip} = 1 \quad \begin{array}{l} j = 1, \dots, J \\ i = 1, \dots, L \end{array} \quad (2)$$

$$\sum_{p=1}^P p(x_{j(i+1)p} - x_{jip}) \geq 1 \quad \begin{array}{l} j = 1, \dots, J \\ i = 1, \dots, L-1 \end{array} \quad (3)$$

$$\sum_{(j,i) \in M_e} x_{jip} \leq |M_e| \cdot z_{ep} \quad \begin{array}{l} e = 1, \dots, E \\ p = 1, \dots, P \end{array} \quad (4)$$

$$\sum_{e=1}^E z_{ep} \leq 1 \quad p = 1, \dots, P \quad (5)$$

$$x_{jip} \in \{0, 1\} \quad \begin{array}{l} j = 1, \dots, J \\ i = 1, \dots, L \\ p = 1, \dots, P \end{array} \quad (6)$$

$$z_{ep} \in \{0, 1\} \quad \begin{array}{l} e = 1, \dots, E \\ p = 1, \dots, P \end{array} \quad (7)$$

The objective (1) is to minimize the total investment budget for the flow line configuration. Equation (2) makes sure that each operation is assigned to exactly one station (= position) in the flow line. Due to (3), the precedence relations among the operations are guaranteed. Restriction (4) ensures that an operation can only be performed at a certain position in the flow line, if proper equipment is installed at that position. At most one type of equipment can be installed at a certain position which is formulated in (5). (6) and (7) are the binary constraints for the decision variables. Note, feasible solutions may contain “empty” positions. We have not added a restriction  $\sum_{e=1}^E (P-p)z_{ep} \geq \sum_{e=1}^E \sum_{t=p+1}^P z_{et}$  for  $p \in \{1, \dots, P-1\}$  to the model, because this is technical overhead. Even without it, the solution has a unique interpretation: Positions  $p$  with  $\sum_{e=1}^E z_{ep} = 0$  can simply be ignored. Stations are represented by those positions for which this expression has a positive value.

### 3 Lower Bounds

Because the problem formulated in the previous section is  $\mathcal{NP}$ -hard, it is very unlikely to find an optimal solution procedure that is efficient. For  $P \geq J \cdot L$ , the feasibility problem is solvable in  $O(J \cdot L)$  time, but it becomes  $\mathcal{NP}$ -complete, if arbitrary values of  $P$  are used. However, in many cases  $P = J \cdot L$  may be used and the construction of heuristics which guarantee to find a feasible solution is simple. What remains is to evaluate these heuristics. Optimal solutions are not available and using the best known feasible solution as a point of reference does not give any insight into the deviation from the optimum result. Hence, lower bounds are needed. A simple lower bound is

$$LB_0 = \sum_{e=1}^E \left( c_e \cdot \max_{j \in \{1, \dots, J\}} \{ |M_e \cap \{(j, i) \mid i \in \{1, \dots, L\}\}| \} \right) \quad (8)$$

which is based on the fact that every equipment is needed at least as often as operations of a single model require that equipment. A more sophisticated approach that we present here is based on column generation.

### 3.1 The Master Problem

In Section 2 we identified stations with positions to which operations are assigned to. To formulate a master problem, we now use the idea that each solution can be described by a one-to-one mapping of the operations to positions  $p \in \{1, \dots, P\}$  where  $P = J \cdot L$ . Loosely speaking, a station is then represented by a sequence of subsequent positions such that all operations in this subsequence require the same type of equipment.

A column in the master problem describes an assignment of operations to positions. To be more precise, we need additional parameters:

- $b_{jips}$  : 1, if operation  $(j, i)$  is assigned to position  $p$  in column  $s$  (0, otherwise);
- $l_s$  : the NPV of the investment to make according to column  $s$ ;
- $S$  : the number of columns.

From this point of view, the decision to make is to select one or more columns. The new decision variable is:

- $y_s$  : 1, if column  $s$  is selected (0, otherwise).

This renders it possible to reformulate the problem as follows:

$$\min \sum_{s=1}^S l_s y_s \quad (9)$$

subject to

$$\sum_{s=1}^S \sum_{p=1}^P b_{jips} y_s = 1 \quad \begin{matrix} j = 1, \dots, J \\ i = 1, \dots, L \end{matrix} \quad (10)$$

$$\sum_{s=1}^S \sum_{j=1}^J \sum_{i=1}^L b_{jips} y_s = 1 \quad p = 1, \dots, P \quad (11)$$

$$\sum_{s=1}^S \sum_{p=1}^P p(b_{j(i+1)ps} - b_{jips}) y_s \geq 1 \quad \begin{matrix} j = 1, \dots, J \\ i = 1, \dots, L-1 \end{matrix} \quad (12)$$

$$y_s \in \{0, 1\} \quad s = 1, \dots, S \quad (13)$$

The objective (9) is to minimize the investment to make. Due to (10), columns must be selected such that every operation is assigned to exactly one position. (11) makes sure that operations are assigned to distinct positions. Both together define a one-to-one mapping of operations to positions. Precedence constraints among the operations come in via (12). Finally, (13) defines the decision variable to be binary valued.

Given all possible columns, this model is equivalent to the one presented in Section 2. Solving the LP-relaxation of this model yields a lower bound. The problem with it is that the number of columns grows exponentially with  $J$  and  $L$  (and  $P$ ). This is where column generation comes in. Starting with a single column ( $S = 1$ ) which defines a feasible solution, we solve the LP-relaxation of the model.

In our implementation the first column is defined by

$$b_{jip1} = \begin{cases} 1, & \text{if } p = (j-1)L + i \\ 0, & \text{otherwise} \end{cases} \quad (14)$$



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Equipment:	1	2	1	3	2	2	3	1	2	1	3	1	2	3	1
Model 1:	•	•	•	•	•										
Model 2:						•	•	•	•	•					
Model 3:											•	•	•	•	•
Station:	×	×	×	×	×		×	×	×	×	×	×	×	×	×

Table 3: An Initial Column

where  $j \in \{1, \dots, J\}$ ,  $i \in \{1, \dots, L\}$ , and  $p \in \{1, \dots, P\}$ . Table 3 illustrates this definition by using the example from Section 1 again.

The investment  $l_1$  that corresponds to this configuration can be computed as follows. For  $p \in \{1, \dots, P\}$ , let

$$m(p) = \sum_{j=1}^J \sum_{i=1}^L j \cdot b_{jip1} \quad (15)$$

be the model assigned to position  $p$ , and

$$o(p) = \sum_{j=1}^J \sum_{i=1}^L i \cdot b_{jip1} \quad (16)$$

be the operation within that model. Furthermore, let  $w_p$  a binary indicator variable which is one if the operation at position  $p$  cannot be performed at the same station as the operation at position  $p-1$ , and which is zero, otherwise. More formally, we have  $w_1 = 1$  and

$$w_p = \begin{cases} 1, & \text{if } m(p) = m(p-1) \\ 1, & \text{if } m(p) \neq m(p-1) \text{ and } e_{m(p),o(p)} \neq e_{m(p-1),o(p-1)} \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

The last line in Table 3 indicates via “ $\times$ ” which positions define a new station ( $w_p = 1$ ). Putting all this together, we can compute

$$l_1 = \sum_{p=1}^P \sum_{e=1}^E \sum_{(j,i) \in M_e} c_e b_{jip1} w_p, \quad (18)$$

the objective function coefficient for the first column.

Once we have solved the LP-Relaxation optimally, the question arises whether or not other columns can exist which, when added to the master problem, could lead to a reduction of the objective function value. To answer this question, we consider the dual of the model. The dual variables  $\mu_{ji}$ ,  $\lambda_p$ , and  $\pi_{ji}$  correspond to the constraints (10), (11), and (12), respectively. The dual then reads as follows:

$$\max \sum_{j=1}^J \sum_{i=1}^L \mu_{ji} + \sum_{p=1}^P \lambda_p + \sum_{j=1}^J \sum_{i=1}^{L-1} \pi_{ji} \quad (19)$$

subject to

$$\begin{aligned}
& \sum_{j=1}^J \sum_{i=1}^L \sum_{p=1}^P b_{jip}( \mu_{ji} + \lambda_p ) \\
& + \sum_{j=1}^J \sum_{i=1}^{L-1} \sum_{p=1}^P p(b_{j(i+1)p} - b_{jip}) \pi_{ji} \leq l_s \quad s = 1, \dots, S
\end{aligned} \tag{20}$$

$$\mu_{ji} \in \mathbb{R} \quad \begin{array}{l} j = 1, \dots, J \\ i = 1, \dots, L \end{array} \tag{21}$$

$$\lambda_p \in \mathbb{R} \quad p = 1, \dots, P \tag{22}$$

$$\pi_{ji} \geq 0 \quad \begin{array}{l} j = 1, \dots, J \\ i = 1, \dots, L-1 \end{array} \tag{23}$$

From the strong duality theorem we know that the optimal objective function value of the master problem's LP-relaxation is equal to the optimal objective function value of its dual. Looking for a column  $S+1$  which may reduce the objective function value of the master problem is thus equivalent to look for a row  $S+1$  which, when added to the dual, may reduce the optimum objective function value of the dual. Such a row must violate restriction (20). In other words, we have to search for values  $b_{jip(S+1)}$  (and a corresponding  $l_{S+1}$  value) such that

$$\sum_{j=1}^J \sum_{i=1}^L \sum_{p=1}^P b_{jip(S+1)}(\mu_{ji} + \lambda_p) + \sum_{j=1}^J \sum_{i=1}^{L-1} \sum_{p=1}^P p(b_{j(i+1)p(S+1)} - b_{jip(S+1)})\pi_{ji} > l_{S+1} \tag{24}$$

holds where the  $\mu_{ji}$ 's,  $\lambda_p$ 's, and  $\pi_{ji}$ 's are parameters now. How to do so efficiently, is the subject of the next subsection. If such values can be found, we add a new column to the master problem, solve its LP-relaxation, and start all over again. This procedure iterates until (24) cannot be fulfilled ("pricing out"). After termination, the lower bound can be strengthened a bit when all parameters  $c_e$  are integral values (which is true especially when we consider the problem of minimizing the length of the flow line). In such cases, the smallest integer value which is greater than or equal to the bound obtained is also a lower bound.

### 3.2 The Subproblem

We will now present a strongly polynomial time algorithm which minimizes

$$l_{S+1} - \sum_{j=1}^J \sum_{i=1}^L \sum_{p=1}^P b_{jip(S+1)}(\mu_{ji} + \lambda_p) - \sum_{j=1}^J \sum_{i=1}^{L-1} \sum_{p=1}^P p(b_{j(i+1)p(S+1)} - b_{jip(S+1)})\pi_{ji} \tag{25}$$

given some values for  $\mu_{ji}$ ,  $\lambda_p$ , and  $\pi_{ji}$ . Clearly, if the minimum of this expression is negative, we have found a solution such that (24) holds, and thus we have found a new column for the master problem. If the minimum is non-negative, we have priced out the problem and the column generation procedure terminates.

The problem of minimizing (25) is equivalent to finding a shortest path from a source to a sink node in an acyclic digraph. To understand this, we need to define that graph, of course.

The underlying idea for constructing a graph which helps us to minimize (25), is based on the fact that for each position  $p \in \{1, \dots, P\}$  we need to specify an operation to fill

in that position. So, what we will work out in more detail just now, is a graph where nodes are represented by a triple  $[j, i, p]$  indicating that operation  $(j, i)$  may be assigned to position  $p$ .

More formally, let  $\Gamma = (N, A, v)$  be the graph to be defined where  $N$  is the set of nodes,  $A$  is the set of arcs, and  $v$  is the set of arc weights. Before we define  $N$ , we note that an operation  $(j, i)$  may never be assigned to a position  $p < i$ , because the precedence relations among the operations require that the  $i - 1$  operations  $(j, 1), \dots, (j, i - 1)$  are assigned to positions prior to the one of operation  $(j, i)$ . Also, operation  $(j, i)$  may never be assigned to a position  $p > P - (L - i)$ , because the precedence relations among the operations require that the  $L - i$  operations  $(j, i + 1), \dots, (j, L)$  are assigned to positions after the one operation  $(j, i)$  is assigned to. Thus, we define

$$N = \{[j, i, p] \mid j \in \{1, \dots, J\} \text{ and } i \in \{1, \dots, L\} \text{ and } p \in \{i, \dots, P - (L - i)\}\} \quad (26)$$

$$\cup \{[0, 0, 0], [J + 1, 0, P + 1]\}$$

where  $[0, 0, 0]$  is the dummy source node and  $[J + 1, 0, P + 1]$  is the dummy sink node.

Let us turn to define the set of arcs  $A$  and the corresponding arc weights  $v$  now. Roughly speaking, we have an arc from node  $[j, i, p]$  to node  $[h, k, t]$ , only if the assignment of operation  $(h, k)$  to period  $t$  is not in conflict with the assignment of operation  $(j, i)$  to period  $p$ . Since we want to assign an operation to every position, it is sufficient to consider arcs with  $t = p + 1$  only. To give a systematic presentation of all arcs, we discriminate them with respect to the type of nodes they connect.

First, there are arcs emanating from the dummy source node and pointing to some of those nodes which correspond to position one. It would be correct to have a connection between the dummy source node and every node with  $p = 1$ . However, we can use the following insight to keep the number of arcs small.

*Insight 1:* If there are two different models  $j$  and  $h$  with  $j < h$  such that their first operations require the same equipment, i.e.  $e_{j1} = e_{h1}$ , then it is sufficient to have an arc from the dummy source node to node  $[j, 1, 1]$ , but not to node  $[h, 1, 1]$ .

*Proof:* Consider an assignment of operations to positions where  $(h, 1)$  is assigned to position 1 and operation  $(j, 1)$  to  $p' \geq 2$ . Note, a solution with  $p' > 2$  is dominated by the assignment which emerges from the one above by shifting all operations assigned to positions 2 thru  $p' - 1$  one position further, and assigning operation  $(j, 1)$  to position 2, because  $e_{h1} = e_{j1}$ . Hence, it is sufficient to consider  $p' = 2$ . Now, keeping all other assignments as they were, the assignments  $(j, 1)$  to 1 and  $(h, 1)$  to 2 represent the same solution. All we have to make sure is that the arc  $([j, 1, 1], [h, 1, 2])$  is included in the graph. The definition of the arc set  $A_4$  will do so (see Insight 4 below).  $\square$

Thus, arcs emanating from the dummy source node can be defined as follows:

$$A_1 = \{([0, 0, 0], [j, 1, 1]) \in N \times N \mid \forall h \in \{1, \dots, j - 1\} : e_{h1} \neq e_{j1}\} \quad (27)$$

It is clear that  $|A_1| \leq J$ . For  $a \in A_1$  where  $a$  is of the form  $([0, 0, 0], [j, 1, 1])$ , we define

$$v_a = c_{e_{j1}} - \mu_{j1} - \lambda_1 + \pi_{j1}. \quad (28)$$

Second, there are arcs entering the dummy sink node from nodes corresponding to position  $P$ . Similar to the definition of  $A_1$  there are cases in which we can reduce the number of arcs using the following insight.

$e_{ji}$	$i = 1$	$i = 2$
$j = 1$	1	2
$j = 2$	2	1

Table 4: Equipment Requirements for the Small Example

*Insight 2:* If there are two different models  $j$  and  $h$  with  $j > h$  such that their last operations require the same equipment, i.e.  $e_{jL} = e_{hL}$ , then it is sufficient to have an arc from the node  $[j, L, P]$  to the dummy sink node, but not from the node  $[h, L, P]$ .

*Proof:* Consider an assignment of operations to positions where  $(h, L)$  is assigned to position  $P$  and operation  $(j, L)$  to  $p' \leq P-1$ . Note, a solution with  $p' < P-1$  is dominated by the assignment which emerges from the one above by shifting all operations assigned to positions  $p'+1$  thru  $P-1$  one position to the left, and assigning operation  $(j, L)$  to position  $P-1$ , because  $e_{hL} = e_{jL}$ . Hence, it is sufficient to consider  $p' = P-1$ . Now, keeping all other assignments as they were, the assignments  $(j, L)$  to  $P$  and  $(h, L)$  to  $P-1$  represent the same solution. All we have to make sure is that the arc  $([h, L, P-1], [j, L, P])$  is included in the graph. The definition of the arc set  $A_4$  will do so (see Insight 4 below).  $\square$

Thus, arcs pointing to the dummy sink node can be defined as follows:

$$A_2 = \{([j, L, P], [J+1, 0, P+1]) \in N \times N \mid \forall h \in \{j+1, \dots, J\} : e_{hL} \neq e_{jL}\} \quad (29)$$

Again, we have  $|A_2| \leq J$ . For  $a \in A_2$  where  $a$  is of the form  $([j, L, P], [J+1, 0, P+1])$ , we define

$$v_a = 0. \quad (30)$$

Third, there are arcs which represent the precedence relations among the operations, i.e.

$$A_3 = \{([j, i, p], [j, i+1, p+1]) \mid [j, i, p] \in N \text{ and } [j, i+1, p+1] \in N\}. \quad (31)$$

It is easy to verify that  $|A_3| = J \cdot (L-1) \cdot (P-L+1)$ . For  $a \in A_3$  where  $a$  is of the form  $([j, i, p], [j, i+1, p+1])$ , we define

$$v_a = \begin{cases} c_{e_{j(i+1)}} - \mu_{j(i+1)} - \lambda_{(p+1)} - (p+1) \cdot (\pi_{ji} - \pi_{j(i+1)}) & , \text{ if } i+1 < L \\ c_{e_{j(i+1)}} - \mu_{j(i+1)} - \lambda_{(p+1)} - (p+1) \cdot \pi_{ji} & , \text{ otherwise} \end{cases} \quad (32)$$

To illustrate the definitions given so far, consider a small example with  $J = L = E = 2$ , i.e.  $P = 4$ , and the equipment requirements given in Table 4. Figure 1 illustrates the node set  $N$ , and the arc sets  $A_1$ ,  $A_2$ , and  $A_3$  (rows correspond to operations and columns correspond to positions).

To complete the definition of the arc set  $A$ , we now consider those arcs which connect nodes belonging to different models. The following insights help to keep the number of arcs small.

*Insight 3:* An arc from node  $[j, i, p]$  to  $[h, k, p+1]$  with  $j \neq h$  can only exist, if  $i+k \leq p+1$  and  $P - (L-i) - (L-k) \geq p+1$  holds.

*Proof:* Due to the precedence relations among the operations, the operations  $(j, 1), \dots, (j, i-1)$  and  $(h, 1), \dots, (h, k-1)$  must be assigned to positions prior to  $p$ . This proves the former condition. Also, we have to take into account that operations  $(j, i+1), \dots, (j, L)$

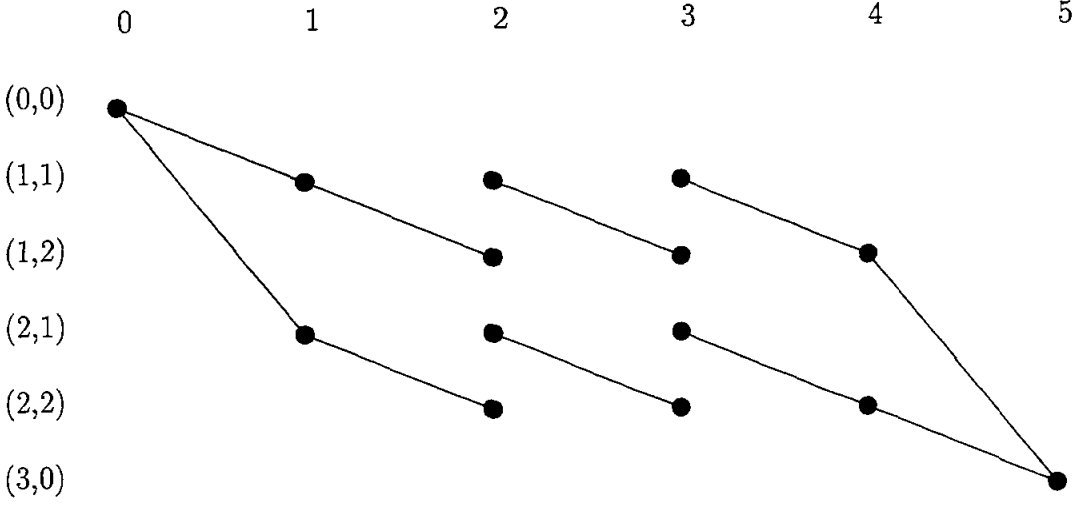


Figure 1: An Illustration of  $N$ ,  $A_1$ ,  $A_2$ , and  $A_3$

and  $(h, k + 1), \dots, (h, L)$  must fill in some positions which is the reason for the latter condition.  $\square$

*Insight 4:* If the operations  $(j, i)$  and  $(h, k)$  require identical equipment, i.e.  $e_{ji} = e_{hk}$ , then we can assume without loss of generality that, if both operations are assigned to subsequent positions  $p$  and  $p + 1$ , the operation with lower model number is assigned to position  $p$ . That is, if  $j < h$  holds, we want to have the arc  $([j, i, p], [h, k, p + 1])$  but not the arc  $([h, k, p], [j, i, p + 1])$ .

*Proof:* The proof is trivial.  $\square$

*Insight 5:* If  $i < L$ ,  $j < h$ , and  $e_{ji} \neq e_{j(i+1)} = e_{hk}$  then we can disregard without loss of generality the arc  $([j, i, p], [h, k, p + 1])$ .

*Proof:* Consider an assignment of operations to positions where  $(j, i)$  is assigned to  $p$ ,  $(h, k)$  to  $p + 1$ , and  $(j, i + 1)$  to  $p' \geq p + 2$ . Note, a solution with  $p' > p + 2$  is dominated by the assignment which emerges from the one above by shifting all operations assigned to positions  $p + 2$  thru  $p' - 1$  one position further, and assigning operation  $(j, i + 1)$  to position  $p + 2$ , because  $e_{hk} = e_{j(i+1)}$ . Hence, it is sufficient to consider  $p' = p + 2$ . Now, keeping all other assignments as they were, the assignments  $(j, i)$  to  $p$ ,  $(j, i + 1)$  to  $p + 1$ , and  $(h, k)$  to  $p + 2$  represent the same solution and is indeed covered by the graph, because there exists an arc from  $[j, i + 1, p + 1]$  to  $[h, k, p + 2]$  by construction.  $\square$

*Insight 6:* If  $k > 1$ ,  $j < h$ , and  $e_{hk} \neq e_{h(k-1)} = e_{ji}$  then we can disregard without loss of generality the arc  $([j, i, p], [h, k, p + 1])$ .

*Proof:* The proof is similar to the one for Insight 5. Consider an assignment of operations to positions where  $(h, k - 1)$  is assigned to  $p' \leq p - 1$ ,  $(j, i)$  to  $p$ , and  $(h, k)$  to  $p + 1$ . Note, a solution with  $p' < p - 1$  is dominated by the assignment which emerges from the one above by shifting all operations assigned to positions  $p' + 1$  thru  $p - 1$  one position further to the left, and assigning operation  $(h, k - 1)$  to position  $p - 1$ , because  $e_{h(k-1)} = e_{ji}$ . Hence, it is sufficient to consider  $p' = p - 1$ . Now, keeping all other assignments as they were, the assignments  $(j, i)$  to  $p - 1$ ,  $(h, k - 1)$  to  $p$ , and  $(h, k)$  to  $p + 1$  represent the same solution and is indeed covered by the graph, because there exists an arc from  $[j, i, p - 1]$  to  $[h, k - 1, p]$  by construction.  $\square$

*Insight 7:* Consider a node  $[j, i, p]$ . It is clear from the interpretation of the nodes that the operations  $(j, 1), \dots, (j, i-1)$  must have been assigned to positions prior to  $p$  then. Consider now the node  $[h, k, p+1]$  where  $h \neq j$ . The sequence of operations  $(h, 1), \dots, (h, k)$  is a subsequence of the sequence of operations  $(j, 1), \dots, (j, i-1)$ , if and only if there exists a function  $\sigma : \{1, \dots, k\} \rightarrow \{1, \dots, i-1\}$  which is strictly increasing and  $e_{hg} = e_{j, \sigma(g)}$  holds for every  $g \in \{1, \dots, k\}$ . This situation will in short-hand notation be stated as  $\langle j, i-1 \rangle \succeq \langle h, k \rangle$ . The insight is that if such situation occurs, it is sufficient to consider the case where  $(h, 1), \dots, (h, k)$  is assigned to positions prior to  $p$ . This is to say, the arc  $([j, i, p], [h, k, p+1])$  need not exist.

Proof: Obvious. A solution where ceteribus paribus  $(h, k)$  is assigned to a position right next to  $(j, \sigma(k))$  cannot be worse.  $\square$

Formally, all this can be stated by

$$\begin{aligned}
A_4 = & \{([j, i, p], [h, k, p+1]) \in N \times N \mid 1 \leq j, h \leq J \text{ and } j \neq h \text{ and} \\
& i + k \leq p + 1 \leq P - (L - i) - (L - k)\} \\
& \cap \{([j, i, p], [h, k, p+1]) \in N \times N \mid 1 \leq j, h \leq J \text{ and } j \neq h \text{ and} \\
& (e_{ji} \neq e_{hk} \text{ or } j < h)\} \\
& \cap \{([j, i, p], [h, k, p+1]) \in N \times N \mid 1 \leq j, h \leq J \text{ and } j \neq h \text{ and} \\
& (i = L \text{ or } j > h \text{ or } e_{ji} = e_{hk} \text{ or } e_{j(i+1)} \neq e_{hk})\} \\
& \cap \{([j, i, p], [h, k, p+1]) \in N \times N \mid 1 \leq j, h \leq J \text{ and } j \neq h \text{ and} \\
& (k = 1 \text{ or } j > h \text{ or } e_{ji} = e_{hk} \text{ or } e_{h(k-1)} \neq e_{ji})\} \\
& \cap \{([j, i, p], [h, k, p+1]) \in N \times N \mid 1 \leq j, h \leq J \text{ and } j \neq h \text{ and} \\
& \langle j, i-1 \rangle \not\succeq \langle h, k \rangle\}.
\end{aligned} \tag{33}$$

Note,  $|A_4| \leq J \cdot (J-1) \cdot L^2 \cdot (P-1)$ . For  $a \in A_4$  where  $a$  is of the form  $([j, i, p], [h, k, p+1])$ , we define:

$$v_a = \begin{cases} c_{e_{hk}} - \mu_{hk} - \lambda_{(p+1)} - (p+1)(\pi_{h(k-1)} - \pi_{hk}) & , \text{ if } e_{ji} \neq e_{hk} \text{ and } 1 < k < L \\ c_{e_{hk}} - \mu_{hk} - \lambda_{(p+1)} + (p+1) \cdot \pi_{hk} & , \text{ if } e_{ji} \neq e_{hk} \text{ and } k = 1 \\ c_{e_{hk}} - \mu_{hk} - \lambda_{(p+1)} - (p+1) \cdot \pi_{h(k-1)} & , \text{ if } e_{ji} \neq e_{hk} \text{ and } k = L \\ -\mu_{hk} - \lambda_{(p+1)} - (p+1)(\pi_{h(k-1)} - \pi_{hk}) & , \text{ if } e_{ji} = e_{hk} \text{ and } 1 < k < L \\ -\mu_{hk} - \lambda_{(p+1)} + (p+1) \cdot \pi_{hk} & , \text{ if } e_{ji} = e_{hk} \text{ and } k = 1 \\ -\mu_{hk} - \lambda_{(p+1)} - (p+1) \cdot \pi_{h(k-1)} & , \text{ if } e_{ji} = e_{hk} \text{ and } k = L \end{cases} \tag{34}$$

Putting all together, we have

$$A = A_1 \cup A_2 \cup A_3 \cup A_4. \tag{35}$$

Figure 2 illustrates the node set  $N$  and the complete arc set  $A$  for the small example above.

A shortest path from the dummy source node to the dummy sink node in the graph  $\Gamma$  defines a new column, if the problem is not priced out. We simply set  $b_{jip(S+1)} = 1$ , if node  $[j, i, p]$  lies on the shortest path, and  $b_{jip(S+1)} = 0$ , otherwise. The objective function coefficient  $l_{S+1}$  is determined by adding the  $c_e$  values associated with the arcs which belong

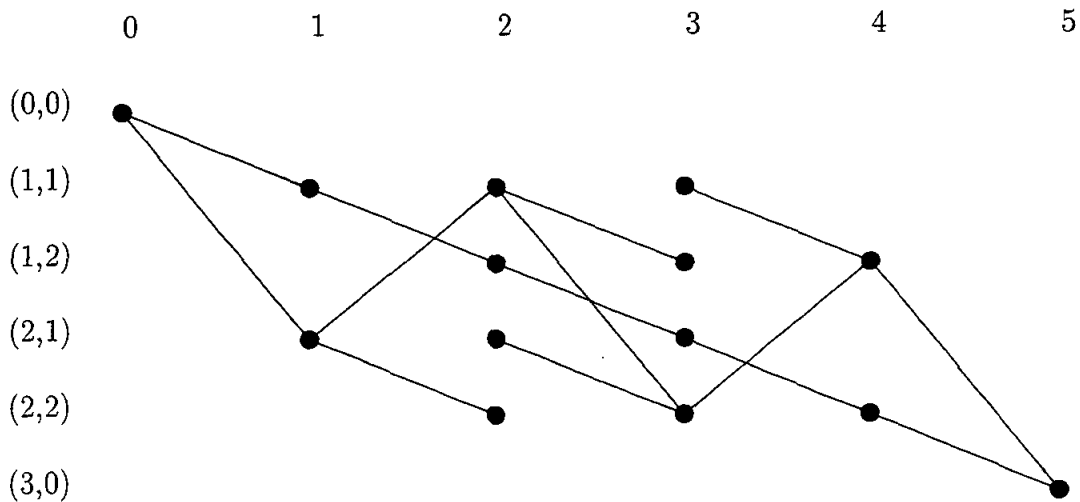


Figure 2: An Illustration of  $N$  and  $A$

to the shortest path. An illustration of the column generation procedure is given in the appendix.

In summary, it has been shown that the subproblem can be solved efficiently by computing a shortest path in an acyclic digraph. The time complexity of such kind of shortest path problems is determined by the number of arcs (see e.g. [1]), because nodes are in topological order. In our case, this means that we can solve the subproblem in  $O(J^3 \cdot L^3)$  time.

We finish this subsection with some important remarks: In our small example every path from the dummy source node to the dummy sink node represents a feasible solution. In general, it may happen that paths exist which do not represent feasible solutions, because some operations  $(j, i)$  occur more than once on such a path (i.e. they occur at several positions) while other operations occur not. Also, Figure 2 shows a symmetric graph. This won't be the case in general. Furthermore, as one can see in Figure 2, there may be nodes which cannot be reached from the dummy source node (see nodes  $[1, 1, 3]$  and  $[2, 1, 2]$ ) as well as there may be paths which are a dead end (see the paths to nodes  $[2, 2, 2]$  and  $[1, 2, 3]$ ). Hence, the graph can be further reduced in a preprocessing phase (i.e., before the iteration of the column generation procedure starts) by deleting arcs. The next subsection is devoted to discuss this in more detail.

### 3.3 Preprocessing

Consider the set of non-dummy nodes in the graph for the subproblem. If there is a node  $n$  to which no arc is pointing to, we cannot reach node  $n$  from the dummy source node. Hence, we can delete all arcs emanating from  $n$ . Similarly, if there is a node  $n$  from which no arc is emanating, we cannot reach the dummy sink node from  $n$ . Hence, we can also delete all arcs pointing to  $n$ . To ease the notation, we use  $d^-(n)$  to denote the in-degree of node  $n$ , i.e. the number of arcs pointing to node  $n$ . Analogously,  $d^+(n)$  denotes the out-degree of node  $n$ , i.e. the number of arcs emanating from node  $n$ . Preprocessing then works as follows:

*Phase 1 (Forward):*

- Step 1.1:* Set  $j = 1$ ,  $i = 1$ , and  $p = 1$ .
- Step 1.2:* If  $p > P$  then goto *Phase 2*.
- Step 1.3:* If  $j > J$  then set  $j = 1$ , increase  $p$  by one, and goto *Step 1.2*.
- Step 1.4:* Compute  $d^-([j, i, p])$ .  
If  $d^-([j, i, p]) = 0$  then delete all arcs emanating from  $[j, i, p]$ .
- Step 1.5:* If  $i < L$  then increase  $i$  by one and goto *Step 1.4*.  
Otherwise, set  $i = 1$ , increase  $j$  by one, and goto *Step 1.3*.

*Phase 2 (Backward):*

- Step 2.1:* Set  $j = 1$ ,  $i = 1$ , and  $p = P$ .
- Step 2.2:* If  $p < 1$  then STOP.
- Step 2.3:* If  $j > J$  then set  $j = 1$ , decrease  $p$  by one, and goto *Step 2.2*.
- Step 2.4:* Compute  $d^+([j, i, p])$ .  
If  $d^+([j, i, p]) = 0$  then delete all arcs pointing to  $[j, i, p]$ .
- Step 2.5:* If  $i < L$  then increase  $i$  by one and goto *Step 2.4*.  
Otherwise, set  $i = 1$ , increase  $j$  by one, and goto *Step 2.3*.

## 4 Upper Bounds

Up to here, we tackled the problem of finding a lower bound for the minimal investment flow line configuration problem. For practice, however, a lower bound for the investment to make is of little help. What is needed is a feasible solution which defines an upper bound. In this section we present two approaches. One makes use of the column generation result while the other is a straightforward construction scheme.

### 4.1 A Column Generation Based Solution

Upon termination of the column generation procedure, we can easily compute

$$\phi_{jip} = \sum_{s=1}^S \sum_{t=1}^p b_{jits} y_s \quad (36)$$

for  $j \in \{1, \dots, J\}$ ,  $i \in \{1, \dots, L\}$ , and  $p \in \{1, \dots, P\}$ . This value can be interpreted as being a priority value for assigning operation  $(j, i)$  to position  $p$ . Using this interpretation, we can derive a construction scheme to find a feasible solution. The symbols used to specify the construction scheme are:  $p$  the current position,  $l$  is the investment to make,  $w_p = 1$  signals the beginning of a new station,  $D$  is the set of operations which are valid candidates to be assigned next,  $F$  is the set of models which are assigned to the current station,  $m(p)$  is the model assigned to position  $p$ , and  $o(p)$  is the operation within that model.



- Step 1:* Set  $p = 1$ ,  $l = 0$ , and  $w_p = 0$  for  $p \in \{1, \dots, P\}$ .  
Set  $D = \{(j, 1) \mid 1 \leq j \leq J\}$ .
- Step 2:* Determine  $(j, i) \in D$  such that  $\phi_{jip} = \max_{(h,k) \in D} \phi_{hkp}$ .
- Step 3:* Set  $w_p = 1$ ,  $F = \{\}$ , and  $l = l + c_{eji}$ .
- Step 4:* Set  $m(p) = j$ ,  $o(p) = i$ ,  $F = F \cup \{j\}$ ,  $D = D \setminus \{(j, i)\}$ , and  $p = p + 1$ .
- Step 5:* If  $i < L$  then set  $D = D \cup \{(j, i + 1)\}$ .
- Step 6:* Search for  $(j, i) \in D$  where  $e_{ji} = e_{m(p-1), o(p-1)}$  and  $j \notin F$ .  
If such  $(j, i)$  can be found then goto *Step 4*.
- Step 7:* If  $D \neq \{\}$  then goto *Step 2*.  
Otherwise, STOP.

As a tie break rule in Step 2, we use the minimum model number  $j$ . Upon termination of this construction scheme,  $l$  is an upper bound for the optimum objective function value. The number of stations can be determined by evaluating  $\sum_{p=1}^P w_p$ . Operation  $o(p)$  of model  $m(p)$  at position  $p$  is performed at station  $\sum_{t=1}^p w_t$ .

## 4.2 A Simple Construction Scheme

Since feasibility is no problem, it is very easy to specify ad hoc heuristics which do not make use of the result of the column generation procedure. The heuristic described above is a priority rule based method. Hence, we can derive a simple variant from it by replacing the definition of the priority value  $\phi_{jip}$ . For instance, we can modify Step 2 and use a dynamic rule (tie breaker: lowest equipment number, lowest model number):

- Step 2:* Compute  $\psi_{ji} = \sum_{(h,k) \in D \cap M_{e_{ji}}} \frac{1}{c_{e_{ji}}}$   
and determine  $(j, i) \in D$  such that  $\psi_{ji} = \max_{(h,k) \in D} \psi_{hk}$ .

The idea of the rule  $\psi_{ji}$  is to assign an operation  $(j, i)$  to the current position  $p$  such that many other operations may use the same equipment while there is a tendency to delay those operations which require high investments. For the special case  $c_e = 1$  for  $e \in \{1, \dots, E\}$ , this procedure is known as “majority merge” (see, e.g., [2, 9]).

## 5 Computational Study

To test the performance of the presented procedures, we have implemented them in GNU C using the CPLEX callable library to solve the LP-relaxation of the master problem. The programs were executed on a Pentium computer with 120 MHz running a LINUX operating system.

### 5.1 The Test-Bed

We defined a fractional experimental design to study the impact of different parameter levels of  $L$ ,  $J$ , and  $E$  on the performance. For each parameter level combination we generated 10 instances randomly. Throughout, we provide the average results only.

To study the impact of  $L$ , i.e. the number of operations per model, we set  $J = 3$  and  $E = 3$ . Table 5 shows the number of arcs in the subproblem and the time measured in

	$L = 5$	$L = 10$	$L = 15$
#Arcs:	540.6	4109.9	12614.2
Initializing time:	0.11	4.58	45.28

Table 5: Instance Characteristics Depending on  $L$

	$J = 3$	$J = 5$	$J = 8$
#Arcs:	540.6	4900.2	26091.4
Initializing time:	0.11	5.98	182.31

Table 6: Instance Characteristics Depending on  $J$

CPU-seconds to initialize the column generation procedure where initializing consists of constructing the graph and performing the preprocessing phase.

In a similar manner, we varied  $J$ , the number of models. In this test, we set  $L = 5$  and  $E = 3$ . Table 6 provides the instance characteristics depending on  $J$ .

Also, we considered three different levels of  $E$ , the number of equipment types. Here, we set  $J = 3$  and  $L = 10$ . Table 7 shows if and how the size of the subproblem depends on  $E$ .

## 5.2 Minimizing the Investment

In a first experiment, we chose the net present values  $c_e$  randomly from the interval  $[1, 100]$  with uniform distribution. For each instance, a new set of  $c_e$  values was drawn. The performance measures of interest are:

- The total number of columns generated.
- The run-time in CPU-seconds of the column generation procedure (without initializing) for finding a lower bound.
- The improvement of the lower bound obtained with respect to the lower bound defined by (8). The improvement is defined by

$$DEV_{LB} = \frac{LB_{CG} - LB_0}{LB_0} \cdot 100 \quad (37)$$

where  $LB_{CG}$  is the lower bound obtained by column generation and  $LB_0$  is the lower bound due to (8).

	$E = 3$	$E = 5$	$E = 8$
#Arcs:	4109.9	5157.5	5625.8
Initializing time:	4.58	6.88	8.09

Table 7: Instance Characteristics Depending on  $E$

	$L = 5$	$L = 10$	$L = 15$
#Columns:	95.1	281.7	833.1
Run-time:	1.25	25.19	358.58
$DEV_{LB}$ :	6.20	5.81	4.15
$DEV_{UB}^{CG}$ :	5.31	22.73	30.55
$DEV_{UB}^{MM}$ :	15.77	35.05	48.46

Table 8: Performance Depending on  $L$

- The deviation of the upper bounds from the lower bound. The deviations are defined as

$$DEV_{UB}^{CG} = \frac{UB_{CG} - LB_{CG}}{LB_{CG}} \cdot 100 \quad (38)$$

for the upper bound  $UB_{CG}$  derived from the column generation result, and

$$DEV_{UB}^{MM} = \frac{UB_{MM} - LB_{CG}}{LB_{CG}} \cdot 100 \quad (39)$$

for the upper bound  $UB_{MM}$  obtained with the majority merge like heuristic. Again,  $LB_{CG}$  is the lower bound given by column generation.

The run-time performance of the heuristics is not reported in more detail, because it turned out that all instances can be solved within less than one second. The majority merge like heuristic works remarkably fast which is due to its simplicity. Its run-time is less than 0.1 seconds for each instance in our test-bed.

Table 8 provides the results depending on  $L$ . It turns out that the lower bound which is computed by column generation improves the simple bound. But, more important, the feasible solution derived from the column generation result is decidedly better than the solution which is computed by applying the majority merge heuristic. Since differences between  $LB_{CG}$  and  $LB_0$  are not dramatic, we conjecture that large gaps between upper and lower bound are due to weak lower bounds.

The impact of  $J$  on the performance is reported in Table 9. The result accords to the one above. The improvement of the simple lower bound  $LB_0$  again is not large. However, the feasible solution derived from the column generation result is extremely better than the one computed with majority merge. For  $J = 8$ , for instance, the deviation of the majority merge result from the lower bound is about five times higher than the column generation based feasible solution.

Table 10 gives some insight into how  $E$  affects the performance. While the results are not as impressive as before, the tendency is the same. The lower bound  $LB_0$  can only be improved slightly. But, the feasible solution gained via column generation is decidedly better than the one obtained with majority merge. The deviations of the upper bounds from the lower bound are more or less the same no matter what value  $E$  is chosen. Thus, it turns out that the main impact of  $E$  is on the run-time performance.

### 5.3 Minimizing the Length

In a second experiment, we used the same instances as before, but this time, we chose  $c_e = 1$  for all  $e \in \{1, \dots, E\}$ . That is, we considered the problem of minimizing the length

	$J = 3$	$J = 5$	$J = 8$
#Columns:	95.1	260.2	524.6
Run-time:	1.25	16.91	158.91
$DEV_{LB}$ :	6.20	5.06	3.22
$DEV_{UB}^{CG}$ :	5.31	20.89	27.55
$DEV_{UB}^{MM}$ :	15.77	75.21	136.43

Table 9: Performance Depending on  $J$

	$E = 3$	$E = 5$	$E = 8$
#Columns:	281.7	329.8	492.7
Run-time:	25.19	34.36	62.41
$DEV_{LB}$ :	5.81	2.12	5.43
$DEV_{UB}^{CG}$ :	22.73	25.92	27.06
$DEV_{UB}^{MM}$ :	35.05	32.21	38.39

Table 10: Performance Depending on  $E$

of the flow line.

In Table 11 we can see what happens of  $L$  is varied. Compared to the minimal investment problem, we have a greater improvement of the lower bound  $LB_0$  now. Again, the feasible solutions derived from the column generation result are much better than the majority merge results. This is remarkable to note, because the column generation procedure was developed to solve a more general problem while the majority merge heuristic is tailored to solve the minimal length problem.

The impact of  $J$  on the performance is revealed in Table 12. Again, the result of the majority merge heuristic is disastrous if  $J$  grows while the column generation based feasible solution is reasonably well. Once more, improvements of the lower bound  $LB_0$  are stronger than for the minimal investment problem.

It remains to study the impact of  $E$  on the performance. Table 13 gives the results and allows to make the same observations as before.

	$L = 5$	$L = 10$	$L = 15$
#Columns:	85.7	281.9	688.9
Run-time:	1.09	26.95	272.75
$DEV_{LB}$ :	14.17	8.48	7.17
$DEV_{UB}^{CG}$ :	4.90	18.00	27.13
$DEV_{UB}^{MM}$ :	19.52	24.41	33.94

Table 11: Performance Depending on  $L$

	$J = 3$	$J = 5$	$J = 8$
#Columns:	85.7	191.4	500.4
Run-time:	1.09	11.6	155.21
$DEV_{LB}$ :	14.17	10.15	7.12
$DEV_{UB}^{CG}$ :	4.90	17.51	21.11
$DEV_{UB}^{MM}$ :	19.52	64.51	126.33

Table 12: Performance Depending on  $J$

	$E = 3$	$E = 5$	$E = 8$
#Columns:	281.9	286.9	459.4
Run-time:	26.95	28.88	56.64
$DEV_{LB}$ :	8.48	4.97	8.58
$DEV_{UB}^{CG}$ :	18.00	18.30	24.44
$DEV_{UB}^{MM}$ :	24.41	32.74	36.55

Table 13: Performance Depending on  $E$

## 6 Conclusion

In this paper we have discussed the  $\mathcal{NP}$ -hard problem of finding a flow line configuration such that the net present value of the cash outflows for installing and maintaining the flow line is minimized. We have shown that this problem can be seen as a generalization of the so-called shortest common supersequence problem and thus covers the special case of minimizing the length of the flow line. A mathematical programming model formulation is given to describe the problem precisely. To compute lower bounds a column generation procedure is developed. The subproblem turns out to be a shortest path problem in an acyclic network which can be solved efficiently. The number of arcs in this network can be reduced by making use of several problem specific insights. Additionally, two heuristics are presented. The first one uses the column generation result and constructs a feasible solution. The second one is a generalization of the so-called majority merge heuristic which has been proposed in the literature to solve the shortest common supersequence problem. A computational study is performed to test the performance of the presented methods. The lower bound obtained by column generation is better than a simple analytic bound. But, even more important is the fact that the feasible solution derived from the column generation result is extremely better than the majority merge solution. Impressive improvements are gained for both, the minimal investment problem as well as the minimal length problem. Since we conjecture that large gaps between upper and lower bounds are due to poor lower bounds, future work should be devoted to lower bounds in order to prove us right.

Position:	1	2	3	4
Operation:	(1, 1)	(1, 2)	(2, 1)	(2, 2)

Table 14: The Initial Column

Position:	1	2	3	4
Operation:	(2, 1)	(1, 1)	(2, 2)	(1, 2)

Table 15: The Second Column

## Acknowledgement

We are grateful to Andreas Drexl for providing a stimulating environment.

## A An Example

To illustrate the column generation procedure, we provide a protocol for the small example with  $J = 2$ ,  $L = 2$ , and  $E = 2$  that has already been used in Section 3. Table 4 gives the equipment requirements. Let us assume  $c_1 = 20$  and  $c_2 = 10$ . The initial column as defined by (14) is illustrated in Table 14. The corresponding objective function coefficient is  $l_1 = 50$ .

Solving the LP-relaxation of the master problem optimally gives  $\mu_{11} = 50$ . All other dual variables are assigned to zero. By solving the subproblem, we then find the shortest path  $[0, 0, 0] \rightarrow [2, 1, 1] \rightarrow [1, 1, 2] \rightarrow [2, 2, 3] \rightarrow [1, 2, 4] \rightarrow [3, 0, 5]$ . The length of that path is -10. From this we can derive the second column for the master problem (see Table 15) and the objective function coefficient  $l_2 = 40$ .

Solving the master problem a second time, gives  $\mu_{11} = 50$  again. All other dual variables are assigned to zero. And again, we solve the subproblem which gives the same shortest path as before. The length still is -10. Thus, the third column is identical to the second, and we have  $l_3 = 40$ , of course. Solving the master problem one more time yields  $\mu_{11} = 40$  and all other dual variables are assigned to zero. Now, the subproblem gives the same shortest path as before, but this time the length is zero. Hence, the problem is priced out and the procedure terminates. The solution of the last master problem defines the lower bound. We get  $y_1 = y_3 = 0$ , and  $y_2 = 1$ . The corresponding objective function value is 40 which is not only a lower bound for the investment to make, but it is the minimal investment, because the solution of the LP-relaxation is an integral solution.

## References

- [1] AHUJA, R.K., MAGNANTI, T.L., ORLIN, J.B., (1993), Network Flows — Theory, Algorithms, and Applications, Englewood Cliffs, Prentice Hall

- [2] BRANKE, J., MIDDENDORF, M., SCHNEIDER, F., (1998), Improved Heuristics and a Genetic Algorithm for Finding Short Supersequences, *OR Spektrum*, Vol. 20, pp. 39–45
- [3] CHANDRASEKARAN, R., ANEJA, Y.P., NAIR, K.P.K., (1984), Production Planning in Assembly Line Systems, *Management Science*, Vol. 30, pp. 713–719
- [4] FOULSER, D.E., YANG, Q., LI, M., (1992), Theory and Algorithms for Plan Merging, *Artificial Intelligence*, Vol. 57, pp. 143–181
- [5] FRASER, C.B., (1995), Subsequences and Supersequences, Ph. D. Dissertation, University of Glasgow
- [6] FRASER, C.B., IRVING, R., (1995), Approximation Algorithms for the Shortest Common Supersequence, *Nordic Journal on Computing*, Vol. 2, pp. 303–325
- [7] GUTJAHR, A.L., NEMHAUSER, G.L., (1964), An Algorithm for the Line Balancing Problem, *Management Science*, Vol. 11, pp. 308–315
- [8] HOFFMANN, T.R., (1992), EUREKA: A Hybrid System for Assembly Line Balancing, *Management Science*, Vol. 38, pp. 39–47
- [9] JIANG, T., LI, M., (1995), On the Approximation of Shortest Common Supersequences and Longest Common Subsequences, *SIAM Journal on Computing*, Vol. 24, pp. 1122–1139
- [10] JOHNSON, R.V., (1988), Optimally Balancing Large Assembly Lines with “FABLE”, *Management Science*, Vol. 34, pp. 240–253
- [11] MAIER, D., (1978), The Complexity of Some Problems on Subsequences and Supersequences, *Journal of the ACM*, Vol. 25, pp. 322–336
- [12] MARTIN, G.E., (1994), Optimal Design of Production Lines, *International Journal of Production Research*, Vol. 32, pp. 989–1000
- [13] MIDDENDORF, M., (1994), More on the Complexity of Common Superstring and Supersequence Problems, *Theoretical Computer Science*, Vol. 125, pp. 205–228
- [14] RÄIHÄ, K.H., UKKONEN, E., (1981), The Shortest Common Supersequence Problem Over Binary Alphabet is NP-Complete, *Theoretical Computer Science*, Vol. 16, pp. 187–198
- [15] ROSENBERG, O., ZIEGLER, H., (1992), A Comparison of Heuristic Algorithms for Cost-Oriented Assembly Line Balancing, *Zeitschrift für Operations Research*, Vol. 36, pp. 477–495
- [16] ROSENBLATT, M.J., CARLSON, R.C., (1985), Designing a Production Line to Maximize Profit, *IIE Transactions*, Vol. 17, pp. 117–121
- [17] SCHOLL, A., (1995), Balancing and Sequencing of Assembly Lines, Heidelberg, Physica

- [18] THOMOPOULOS, N.T., (1967), Line Balancing-Sequencing for Mixed-Model Assembly, *Management Science*, Vol. 14, pp. B59-B75
- [19] TIMKOVSKY, V.G., (1990), Complexity of Common Subsequence and Supersequence Problems and Related Problems, *Cybernetics*, Vol. 25, pp. 565-580
- [20] VAN ZANTE-DE FOKKERT, J.I., DE KOK, T.G., (1997), The Mixed and Multi Model Line Balancing Problem: A Comparison, *European Journal of Operational Research*, Vol. 100, pp. 399-412