



Optimizing Modular Machining Line Design Problem with Mixed Activation Mode of Machining Units

Sana Belmokhtar*, Alexandre Dolgui**,
Xavier Delorme**, Ivan Ignatenko***

Abstract. A modular transfer line designing problem is investigated. The problem is to find the best subset of modules (machining units) from a given set and to assign them to different stations so that technological constraints and cycle upper limit are respected and the line cost is minimal. The investigated lines have a mixed activation mode for the machining units of each station, *i.e.* the units of each station are arranged into a series of stages such that each stage is composed of several units activated in parallel. A mixed integer program approach is proposed to model and solve the corresponding design problem. Improvements are suggested in order to reduce the model size and speed up the computations.

Keywords: modular machining line, cost optimization, mixed integer programming

Mathematics Subject Classification: 90C09, 90C05, 90C57

Received/Revised: 27 February 2007/22 May 2007

1. INTRODUCTION

The automotive industry market is one of the most unpredictable because of the global competition. Manufacturers companies have to face a high fluctuation in product demand and mix. Modular systems seem to be the most suitable to shorten the lead time and fasten the reconfiguration of the machining lines because it becomes possible to integrate or remove some modules without affecting the whole system. It increases feasibility of product/component change, simplifies the introduction of new product or the use of new technology process. Although these benefits, a formal approach for modularity is still lacking (Kusiak, 1999).

In this paper, an approach based on linear programming is suggested to configure/reconfigure modular machining lines from a set of available modules (machining units for the lines) with a mixed activation mode for modules of stations (noted

* CRAN – Research Center for Automatic Control of Nancy, France.

E-mail: Sana.Belmokhtar@eustib.uhp-nancy.fr.

** Scientific Methods for Industrial Management Dept. – (MSGI), Industrial Engineering and Computer Science Division (G2I), Ecole des Mines de Saint Etienne, France.

E-mail: {dolgui,delorme}@emse.fr.

*** Grodno State University, Belarus. E-mail: ignatenko@grsu.by.

TLBP-B/M for *Transfer Line Balancing Problem with known Blocks and mixed activation mode*).

The study deals with completely automatic machining lines using multi-spindle units as modules such that each multi-spindle can perform one or several operations. Such machining line consists in a series of stations disposed linearly without buffers in between. The machining process for each part is complete once the part has visited all the stations while spending a cycle time at each station. The parts are transferred from one station to the following with a convoyer. Each station is a sequence of several stages and each stage is composed of several parallel units. This composition line is more complex than the parallel or serial one. For instance, when each station is composed of only one stage the TLBP-B/M is reduced to the TLBP-B/P studied in Belmokhtar et al. (2006).

These lines are synchronous such that a cycle time is determined by the station bottleneck. More precisely, a cycle time corresponds to the elapsed time between the starting of the activation of all spindle units of the line and their end. For each station, the first stage activates simultaneously its multi-spindle units (the corresponding subset of operations are done in parallel) when they terminate the units of the second stage are activated and so on until all the stages of the corresponding station are performed.

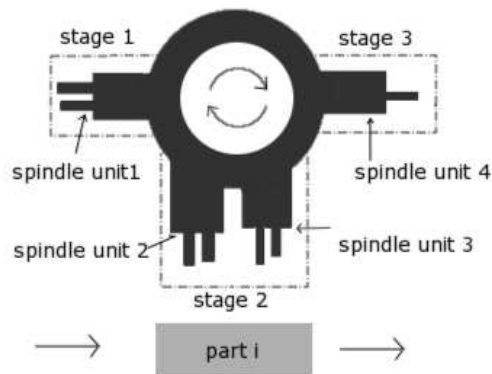


Fig. 1. A three-stage machining station with 4 spindle units

Figure 1 illustrates a station with three sequential stages. The cycle time begins when the first stage, materialized by its spindle unit, is positioned in front of the part. At this moment, the corresponding spindle unit is activated. Then, the equipment of the second stage is positioned by a rotation of the mechanism. By the same way the operations of the second stage are performed by a simultaneous activation of the two parallel spindle units. Finally, the third stage is positioned and the last operation is executed.

The design problem is defined as the selection of a subset of spindle units and their assignment to stages of stations while respecting cycle time and different other

constraints. The objective is to minimize the total cost which is composed of the fixed cost of stations and those of spindle units. This cost can be either an investment cost at preliminary design step or a reconfiguration time in case of redesign.

The rest of the paper is organized as follows: Section 2 presents a brief review on close problems in the literature. Section 3 gives a formal description of the problem as a Mixed Integer Program (MIP). Section 4 illustrates the problem by a small example. Experimental results are reported in Section 5. Finally, some perspectives and conclusion are given in Section 6.

2. RELATED WORKS

This section is devoted to a review of the closest works to the present study. The most often task assignment is addressed in the assembly environment. The Simple Assembly Line Balancing Problem (*SALBP*) is the most studied problem considering this issue. For this problem, a set of elementary operations has to be gathered in a minimal number of stations. Methods based on branch and bound Johnson, (1988) and dynamic programming approaches Held et al. (1963) have been proposed to solve it. Comprehensive surveys are presented in Baybars (1986) Erel, Sarin (1998) Rekiek et al. (2002) and Scholl (1999). If in the TLBP-B/M the equipment cost is neglected, all the stages are reduced to have one spindle unit and each spindle unit can execute just one operation then the TLBP-B/M is reduced to SALBP. However, in the machining process, the consideration of the equipment cost is necessary to estimate the whole line cost since the spindle units are extremely expensive.

An equipment selection coupled with a balancing assembly line problem has been studied in Bukchin, Tzur (2000) and Bukchin, Rubinowitz (2003). The works considers the cost induced by the equipment. Nevertheless, the authors considers only one type of equipment per station. Moreover, the operations are elementary as in the SALBP whereas in the TLBP-B/M the spindle units perform several operations in a parallel way.

In Sawik (2000) integer programming formulations and a heuristic solution procedure are proposed for allocation of the assembly tasks simultaneously with selection of the assembly plans and assembly routes for different products in a flexible assembly line. A top down approach is proposed considering bi-criterion problem then an improvement to a two-level solution is given. In Sawik (2002), the author presents a monolithic approach for balancing and scheduling a flexible assembly lines. In addition, a hierarchical approach is suggested to first balance the station workloads and then to precise the assembly schedule (tasks assignment and assembly routes).

A machining process optimization problem is studied in Szadkowski (1997) where the times of operations are depending on the cutting mode. The authors formulated the problem as a SALBP which is solved by a search of the shortest path in a particular graph.

In Masood (2006), the authors studies a balancing transfer line with automatic cylinder blocks. The aim is to reduce the cycle time and to increase the machines

utilization rate. The suggested approach is based on simulation for tools changing and operations sequencing.

Reconfigurable machining lines producing modules are studied in Yigit et al. (2002). The aim is to increase diversification of products which are obtained by combining instances of different modules. A line is dedicated for the production of one module and its reconfiguration allow to produce different instances of the corresponding module. an approach based on genetic algorithms is proposed in Tang et al. (2003) to simultaneously balancing multi-product lines and selecting machines with stocks in between stations. The objective considered is to maximize the line productivity.

A transfer line balancing problem (TLBP) have been studied in Dolgui et al. (2005) and Dolgui et al. (2006). The objective was to minimize the total number of built blocks of operations (for which spindle units should be provided) and stations with respect to several technological constraints. A Mixed Integer Program (MIP), heuristics and a decomposition approach have been suggested. These models define the blocks of operations and arrange them into several stations minimizing the total number of blocks and stations. The approaches for the TLBP can be seen as complementary to those for the TLBP-B/M studied in this paper. For instance, the aim of the TLBP-B/M is to optimize the total line cost by considering the cost of each spindle unit since they are known beforehand. Moreover, the TLBP-B/M has a more complex structure than the TLBP. Precisely, if each stage contains only one spindle unit the structure of the TLBP-B/M is reduced to the simple sequential mode of the TLBP (Dolgui et al. 2005; 2006).

The transfer line balancing problem considering the parallel activation mode for spindle units in stations (TLBP-B/P) is studied previously in Belmokhtar et al. (2002). The TLBP-B/M considered in this paper is more complex than the TLBP-B/P for the following reasons:

- In the TLBP-B/P, the stations are composed of parallel spindle units only. In the case of TLBP-B/M, the stations are composed of several sequential stages and each stage contains several spindle units which are activated in parallel. So, when there is one stage at each station the TLBP-B/M is reduced to the TLBP-B/P.
- In the TLBP-B/P, the cycle time constraint can be treated in pre-processing by eliminating the spindle units which do not respect it. For the TLBP-B/M, these constraints should be added explicitly in the model because of the sequential execution of stages in stations.
- In presence of several stages at the same station two types of incompatibilities for spindle units should be considered, *i.e.* the inability of spindle units to be assigned:
 - a) to the same stage,
 - b) to the same station.

The next section is devoted to the formal description of data and constraints of the problem.

3. FORMAL DESCRIPTION OF PROBLEM

The TLBP/B-M is defined by the following items:

- \mathbf{N} is the set of operations that have to be performed on each part (they represent drilling, milling or boring operations);
- \mathbf{B} is the set of available multi-spindle units (blocks);
- $\forall b \in \mathbf{B}, \mathcal{N}(b) \subset \mathbf{N}$ there exists a multi-spindle unit b to perform all the operations $\mathcal{N}(b)$;
- c_b is the cost of the unit b ;
- C is the cost for establishing a station;
- t_b is the execution time of the unit b
- m_0 and n_0 are respectively the maximum number of stations to establish and maximum number of assigned multi-spindle units at any station;
- q_0 is the maximal number of stages at any station;
- A minimal threshold on *productivity* is assured by imposing a maximal cycle time, noted T_0 , for each station. For such lines, the effective cycle time T is determined by the bottleneck station (the slowest one). Moreover, each station time is equal to the sum of the operating time of its stages (because of the sequential execution of stages). For each stage, the operating time is determined by its spindle unit having the highest execution time (because of the parallel execution of spindle units at each stage);
- Each operation have to be performed once at the line, there should neither be omitted nor duplicated operations;
- *Precedence constraints* describe a partial order on the set of operations \mathbf{N} . They are represented by an acyclic digraph $G^{or} = (N, D^{or})$. An arc $(i, j) \in \mathbf{N} \times \mathbf{N}$ belongs to the set D^{or} if the operation j must be executed after the operation i . If there exists a precedence constraint between any couple of operations (o, o') , then the unit chosen for o should be assigned in a stage strictly anterior to the stage of unit for machining o' (a stage s_j^k is anterior to s_i^l if $(k = l$ and $j < i)$ or if $(k < l)$).
- *Inclusion constraints* are imperative requirements for executing some operations to the same station. They are represented by a collection D^{in} of subsets $d \subset \mathbf{N}$ such that all operations from d must be performed at the same station. These constraints can also be represented by a graph G^{in} where each subset $d \subset \mathbf{N}$ modeling an inclusion constraint corresponds to a clique in this graph.
- Two different kind of *incompatibilities* for multi-spindle units are considered:
 - *exclusion constraints* represent incompatibilities between multi-spindle units which can not be assigned to the same station. They are described by a collection D^{ex} of subsets $e \subset \mathbf{B}$. It is not allowed to assign the whole of operations corresponding to any set e to the same station. This restriction does not concern any subset strictly included in e . The graph G^{ex} is used to represent the

exclusion constraints such as each arc represents an incompatibility between the corresponding nodes (multi-spindles units).

- *parallelism constraints* define the spindle units that can be assigned to the same stage. They are represented by a collection D^{pb} of subsets $p \subseteq \mathbf{B}$. The graph G^{bp} is used to represent the collection D^{pb} .

3.1. AN ILLUSTRATION EXAMPLE

A set of 5 operations has to be performed, i.e. $\mathbf{N} = \{1, \dots, 5\}$. The different spindle units b_1, \dots, b_8 which are available to perform the operations are described in Table 1.

Table 1. Spindle units description

b	$\mathcal{N}(b)$	t_b	c_b	b	$\mathcal{N}(b)$	t_b	c_b
b_1	$\{1\}$	5.4	14.5	b_5	$\{1,3\}$	8.6	30.6
b_2	$\{2\}$	7.8	16.9	b_6	$\{1,2,3\}$	8.6	41.4
b_3	$\{3\}$	12.8	18.7	b_7	$\{4\}$	12.5	11.4
b_4	$\{1,2\}$	8.6	25.4	b_8	$\{5\}$	6.8	16.5

The cost of a station is 250 and the cycle time has not to exceed 30. The limit on the number of stations is $m_0 = 3$ and maximal number of stages per station is $q_0 = 2$



Fig. 2. Precedence graph G^{or}

The precedence graph G^{or} given in Figure 2 represents the order relation between operations 2 and 3 and between operations 4 and 5. Operation 2 should be performed in a stage that is anterior to the stage performing operation 3 (the same for operations 4 and 5).

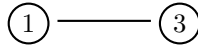


Fig. 3. Inclusion graph G^{in}

The inclusion graph G^{in} corresponds to the collection D^{in} (see Fig. 3). Operations 1 and 3 must be assigned at the same station.



Fig. 4. Exclusion graph G^{ex}

The incompatibility between blocks are represented by G^{ex} (see Fig. 4), where the blocks b_5 and b_8 should not be at the same station. Similarly for blocks b_8 and b_4 .

For this exemple, there are only couples of incompatibilities, so a graph can be used. Nevertheless, the model considers the general case where several blocks could be incompatible. In this case, these constraints can be represented by an hyper-graph.

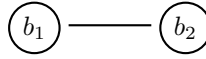


Fig. 5. Parallelism graph G^{bp}

The graph G^{bp} (see Fig. 5) provides the parallelism possibilities described previously by D^{pb} . For instance, the block b_1 can be with only b_2 in parallel at the same stage. In other words, b_1 should not be assigned to the same stage than other blocks b_3, \dots, b_8 .

A feasible solution for this problem is $L = \{s_1^1, s_2^1, s_1^2, s_2^2\}$ where block b_1 and b_2 are assigned to stage s_1^1 , block b_3 to s_2^1 , block b_7 to s_1^2 and block b_8 to s_2^2 . The cost of this solution corresponds to the cost of 2 stations: 2×250 which is added to the cost of all assigned blocks: c_1, c_2, c_3, c_7 and c_8 i.e: $14.5 + 16.9 + 18.7 + 11.4 + 16.5$. The cycle time of the line is defined by the slowest station: for the first station the execution time is the addition of the times of stage 1 and 2. The time of stage 1 is determined by the maximum operating time between t_1 and t_2 which is $t_2 = 7.8$. The time of stage 2 is $t_3 = 12.8$. Thus, the time of station 1 is equal to 20.6. The time of the second station is $12.5 + 6.8 = 19.3$. Therefore, the actual cycle time is 20.6.

3.2. DECISION VARIABLES

Each station $k = 1, \dots, m, m \leq m_0$ is represented by the sequence of its stages $\{s_1^k, \dots, s_j^k, \dots, s_{q_k}^k\}$ where q_k is the number of stages in the station k . Thus, a solution for the considered configuration problem is represented by:

$$L = \{s_1^1, \dots, s_{q_1}^1, \dots, s_j^k, \dots, s_1^m, \dots, s_{q_m}^m\}.$$

Furthermore, $Q(i) = \{b \in \mathbf{B} | i \in \mathcal{N}(b)\}$ represents the set of all blocks (the term block is used here to refer to the group of operations b performed by the corresponding spindle unit) from \mathbf{B} performing the operation $i \in \mathbf{N}$.

3.2.1. Assignment of units to stages

The variables x_{bjk} are introduced for each block b to decide whether it is assigned to the stage s_j^k or not.

$$x_{bjk} = \begin{cases} 1 & \text{if block } b \text{ is assigned to } s_j^k, \forall k = 1, \dots, m_0, \forall j = 1, \dots, q_0, \forall b \in \mathbf{B}; \\ 0 & \text{otherwise} \end{cases}$$

3.2.2. Opened stations

Decision concerning the opening of stations are represented by following variables:

$$y_k = \begin{cases} 1 & \text{if the } k^{\text{th}} \text{ station is opened,} \\ 0 & \text{otherwise.} \end{cases}, \forall k = m^* + 1, \dots, m_0;$$

where m^* is a lower bound on the number of stations to be established.

Thus, there are at least m^* stations which will be established in each feasible solution. Therefore, we can consider the variables y_k for the stations with $m^* + 1 \leq k \leq m_0$.

3.2.3. Stages operating time

To express cycle time constraints (each station working time should respect the maximum limit T_0), additional variables are considered. In particular, for each stage s_j^k the continuous variable t_j^k is used to determine the operating time of corresponding stage.

3.3. MIXED INTEGER FORMULATION

The objective function is expressed as follows:

$$\text{Minimize } \sum_{k=m^*+1}^{m_0} C y_k + \sum_{b \in \mathbf{B}} \sum_{k=1}^{m_0} \sum_{j=1}^{q_0} c_b x_{bjk} \quad (1)$$

The objective (1) did not consider the cost induced by the opening of the first m^* stations. This cost Cm^* is induced in each feasible solution, there is no need to integrate it in the objective function. The cost Cm^* should be added to the value given by the objective (1) to get the total cost of the line.

Note: Once a station is opened (when a block is assigned to one of its stage) the use of an additional stage does not induce any additional cost. This, means that many solutions with different stages composition will have the same cost.

Constraints (2) ensure for each operation from \mathbf{N} its execution in one and only one station:

$$\sum_{b \in Q(i)} \sum_{k=1}^{m_0} \sum_{j=1}^{q_0} x_{bjk} = 1, \quad \forall i \in \mathbf{N} \quad (2)$$

The precedence constraints are expressed to impose the assignment of predecessor operations to earlier stages than successor operations. These constraints are formulated in (3):

$$\sum_{b \in Q(o)} \sum_{h=1}^{k-1} \sum_{l=1}^{q_0} x_{blh} + \sum_{b \in Q(o)} \sum_{l=1}^{j-1} x_{blk} \geq \sum_{r \in Q(o')} x_{rjk}, \quad (3)$$

$$\forall (o, o') \in D^{or}, \forall j = 1, \dots, q_0, \forall k = 1, \dots, m_0$$

The constraints (4) express the inclusion restrictions:

$$\sum_{b \in Q(o)} \sum_{l=1}^{q_0} x_{slk} = \sum_{r \in Q(o')} \sum_{j=1}^{q_0} x_{rjk}, \quad \forall (o, o') \in d, \forall d \in D^{in}, \forall k = 1, \dots, m_0 \quad (4)$$

The exclusion constraints for blocks are respected by imposing the inequalities (5):

$$\sum_{b \in e} \sum_{j=1}^{q_0} x_{bjk} \leq |e| - 1, \quad \forall e \in D^{ex}, \forall k = 1, \dots, m_0 \quad (5)$$

The maximal number of blocks per station is respected using the following constraints:

$$\sum_{b \in \mathbf{B}} \sum_{j=1}^{q_0} x_{bjk} \leq n_0, \quad \forall k = 1, \dots, m_0 \quad (6)$$

The constraints of parallelism are formulated as follows:

$$\sum_{b \in p, b \neq s} x_{bjk} \leq 1 - x_{sjk}, \quad \forall s \in p, \forall p \in \overline{D^{bp}}, \forall k = 1, \dots, m_0, \forall j = 1, \dots, q_0 \quad (7)$$

such that $\overline{D^{bp}}$ is the collection correspondent to the complement graph $\overline{G^{bp}}$ of the graph G^{bp} .

Constraints (8) describe the operating time of stages. In particular, the operating time of each stage is determined by the largest execution time of its assigned blocks.

$$t_b x_{bjk} \leq t_j^k, \quad \forall b \in \mathbf{B}, \forall k = 1, \dots, m_0, \forall j = 1, \dots, q_0 \quad (8)$$

The working time of each station should be less or equal to the maximum cycle time T_0 . Thus, for each station, the sum of the operating time of its stages should respect T_0 . The corresponding constraints are given by (9).

$$\sum_{j=1}^{q_0} t_j^k \leq T_0, \quad \forall k = 1, \dots, m_0 \quad (9)$$

The constraints (10) are added in order to eliminate from the search solutions within empty stations in between.

$$y_{k-1} - y_k \geq 0, \quad \forall k = m^* + 2, \dots, m_0 \quad (10)$$

The constraints (11) are used to fix variable y_k once a block is assigned to station k. This constraints allow to consider the cost of the opened station in the objective function.

$$y_k \geq x_{bjk}, \quad \forall b \in \mathbf{B}, \forall k = m^* + 2, \dots, m_0, \forall j = 1, \dots, q_0 \quad (11)$$

3.4. REDUCTION VARIABLES

The idea is to reduce the number of variables x_{bjk} by exploiting the precedence and inclusion constraints in a preprocessing procedure. The present algorithm is a generalization of the algorithm for the TLBP-B/P proposed in Belmokhtar et al. (2006) for workstations indexes.

3.4.1. The reduction algorithm

An interval $[head_b, tail_b]$ is defined for each block b where $head_b = s_1^1, \dots, s_{q_0}^{m_0}$ is the earliest stage and $tail_b = s_1^1, \dots, s_{q_0}^{m_0}$ is the latest stage of block b . Initially, this interval is considered as equal to $[s_1^1, s_{q_0}^{m_0}]$ for all blocks whereas it could be tightened by increasing the earliest and decreasing the latest stage to which the block b could be assigned. In particular, the precedence constraints are first used to initialize the earliest and latest stages.

For example, a block which has a predecessor can not be assigned to the first stage, thus its earliest stage should be (at least) the second stage of the first station, *i.e.* $head_b \geq s_2^1$. Thus, the value of s_1^1 is eliminated from the interval. This way, variables corresponding to these values are omitted (for the last example, the variable x_{b11} is not considered since s_1^1 is not possible for b).

This principle is further applied to inclusion constraints of operations. The algorithm is reported below. To obtain the tails the precedence graph is reversed and the same algorithm is used to compute a transition value noted $transit_b$ for each block $b \in \mathbf{B}$. Then, for each block b , the latest stage is obtained by: $tail_b = m_0 + 1 - transit_b$. Let's define the following :

— $rank(i)$ is the rank of operation i in the precedence graph, it is computed as follows:

$$rank(j) = \begin{cases} \max\{rank(i) \mid i \in \mathbf{N}, (i, j) \in G^{or}\} + 1, & \text{if } j \text{ has predecessors,} \\ 1 & \text{otherwise,} \end{cases}$$

- $E(i)$ is the index of the earliest stage for each operation $i \in \mathbf{N}$,
- Cpt is a counter to check if the graph is acyclic,
- St_{max} is the index of the earliest station.

Algorithm for computation of $head_b$

Step 1: For all $i \in \mathbf{N}$

$$E(i) = rank(i)$$

$$Cpt = 0$$

Step 2: Set $imp_E = 0$

Step 3: For all $d \in D^{in}$

$$St_{max} = \max\{\lfloor E(i)/q_0 \rfloor \mid i \in d\}$$

For all $i \in d$

$$E(i) = \max(E(i), St_{max} \times q_0)$$

Step 4: For all $i \in \mathbf{N}$
 $E_{imp}(i) = \max\{E(i') + 1 \mid \forall i' \in \mathbf{N} - \{i\}, (i', i) \in D^{or}\}$
 If $(E_{imp}(i) > E(i))$ then
 $E(i) = E_{imp}(i)$
 $imp_E = imp_E + 1$
 Step 5: Set $C_{pt} = C_{pt} + 1$
 If $C_{pt} > |\mathbf{N}|$ then
 stop (there is no feasible solution)
 If $(imp_E > 0)$ then
 goto Step 2
 Step 6: Set $head_b = \max\{E(i) \mid i \in \mathcal{N}(b)\} \forall b \in \mathbf{B}$

3.4.2. A lower bound on the number of stations

Since all the operations have to be executed exactly once, thus the maximum earliest stage provides a lower bound on the number of stations to establish.

$$m^* = \max\{\lceil E(i)/q_0 \rceil \mid i \in \mathbf{N}\} \quad (12)$$

3.4.3. Example

The stage index limits for the blocks of the example shown in section 3.1 are reported in Table 2.

Table 2. Stage index limits

<i>blocks</i>	<i>head_b</i>	<i>tail_b</i>	<i>blocks</i>	<i>head_b</i>	<i>tail_b</i>
b_1	s_2^1	s_2^3	b_5	s_2^1	s_2^3
b_2	s_1^1	s_1^3	b_6	s_2^1	s_1^3
b_3	s_2^1	s_2^3	b_7	s_1^1	s_1^3
b_4	s_2^1	s_1^3	b_8	s_2^1	s_2^3

4. EXPERIMENTAL STUDY

In this section, some numerical results of computation studies are given. The model has been implemented with ILOG Cplex 9.0 which uses a *Branch & Cut* algorithm. The computations run on a Pentium 4 with 2.8 Ghz and 448 MB of RAM. The aim was to study the influence of some characteristics of the problem on the global performance of the algorithm. More precisely, the impact of the number of operations $|\mathbf{N}|$ and the density of the precedence graph on the running time is analyzed.

Table 3. Computational results

$ \mathbf{N} $	20	30*	50*	60*
$ \mathbf{B} $	40	60	100	120
minRun	6	111	15	216
maxRun	34	14 400	14 400	14 400
avgRun	12	3719	4359	4946
median	10	1186	1631	1641
avg nodes	70	1940	7758	14 758
avg init vars	2080	3200	9880	12 320
avg vars	120	509	1439	2055

* None of the instances could be optimally solved within 4 hours

The results are reported in Table 3. Each column corresponds to one family of ten instances generated randomly which are described by the number of operations $|\mathbf{N}|$ and number of blocks $|\mathbf{B}|$. The rows *minRun*, *maxRun*, *avgRun* and *median* provide respectively the minimum, maximum, average and median of the running time in seconds for the corresponding family. *min nodes*, *max nodes* and *avg nodes* provide respectively the minimal, maximal and average number of nodes developed in the *Branch & Cut* for each corresponding family. The row *avg init vars* gives the initial average number of variables, *i.e.* $Avg|\mathbf{B}| \times m_0 \times q_0$. The row *avg vars* provides the average number of variables in the families after the reduction process (applying the previous algorithm). From Table 3, it is observed that when the size of instances increases (20 to 30 operations) the size of the tree increases (in average it goes from 70 to 14 758 nodes) due to the growing of the number of variables. By the same way, the computation time increases from 0.5 to more than 14 400 seconds (for two instances with 50 and two with 60 operations the run was stopped after 4 hours).

Table 4. Influence of density of precedence graph $Density(G^{or})$

$Density(G^{or})$	$ \mathbf{N} = 20, Avg \mathbf{B} = 40, m_0 = 13, q_0 = 4$		
	$\in [0.05, 0.1]$	$\in [0.2, 0.5]$	≥ 0.55
minRun	1.35	0.9	0.16
maxRun	264.41	152.09	14.08
avgRun	83.69	22.05	3.1
min nodes	16	62	0
max nodes	8481	2248	435
avg nodes	3354.5	3221.7	103.1
avg init vars	2080	2080	2080
avg vars	634.3	391.8	235.4

The density is defined as the ratio between the number of edges in the digraph G^{or} and the number of edges in the corresponding complete graph (the graph with the same vertices and all possible edges).

Table 4 presents the computations results on three families of instances with the same cardinality of \mathbf{N} and \mathbf{B} (the chosen values are $|\mathbf{N}| = 20, Avg|\mathbf{B}| = 40$: *i.e.* the average number of blocks is 40) while varying the density of precedence graph. As

the density of precedence graph grows a decrease of the running time is noticed. This is explained by the great number of possible assignment for blocks to stations when there are few precedence constraints. Thus, the search tree size is smaller when the density is high as observed in the row with the number of nodes (see table 4).

5. CONCLUSION

In this paper, a new transfer line balancing problem in machining process environment (TLBP-B/M) is addressed. The aim is to find the best subset of spindle units from a given set and to assign them to stations while all the constraints are satisfied and the total line cost is minimal. A mixed integer program is suggested. The TLBP-B/M is a more complex problem than the TLBP previously studied. This is mainly due to the presence of stages at stations and serial/parallel execution of operations with additional exclusion and inclusion constraints.

Such approach is not limited to automatic machining transfer lines but can be easily extended to any flow line with mixed (sequential and parallel) execution of operations. Computation results obtained using Cplex are promising. Problems with up to 60 operations and 120 blocks have been solved to optimality (although the proof was not obtained for four instances of them). Nevertheless, an upper bound on the number of stations could reduce the search area. Furthermore, an improvement by penalizing some equivalent solutions could improve calculation time.

Acknowledgments

The authors would appreciate the support of the project INTAS N 03-51-5501.

REFERENCES

1. I. Baybars, *A survey of exact algorithms for the simple line balancing problem*, Management Science, 32 (1986), 909–932.
2. S. Belmokhtar, A. Dolgui, N. Guschinsky, G. Levin, *Integer programming models for logical layout design of modular machining lines*, Computers and Industrial Engineering, 2006.
3. J. Bukchin, J. Rubinovitz, *A weighted approach for assembly line design with station paralleling and equipment selection*, IIE Transactions, 35 (2003), 73–85.
4. J. Bukchin, M. Tzur. *Design of flexible assembly line to minimize equipment cost*, IIE Transactions, 32 (2000), 585–598.
5. A. Dolgui, B. Finel, N. Guschinsky, L. Levin, F. Vernadat, *An heuristic approach for transfer lines balancing*, Journal of Intelligent Manufacturing, 16 (2005), 2, 159–171.
6. A. Dolgui, B. Finel, N. Guschinsky, L. Levin, F. Vernadat, *MIP approach to balancing transfer lines with blocks of parallel operations*, IIE Transactions, 38 (2006), 869–882.

7. E. Erel, S. Sarin. *A survey of the assembly line balancing procedures*, Production Planning and Control, 9 (1998), 5, 414–434.
8. M. Held, R. Karp, R. Shareshian, *Assembly line balancing dynamic programming with precedence constraints*, Operations Research, 11 (1963), 442–459.
9. J. Johnson, *Optimally balancing large assembly lines with fable*, Management Science, 34 (1988), 240–253.
10. Kusiak, *Enginnering Design: Products, Processes, and Systems*, Academic Press, 1999.
11. S. Masood, *Line balancing and simulation of an automated production transfer line*, Assembly Automation, 26 (2006), 1, 69–74.
12. B. Rekiek, A. Dolgui, A. Dechambre, A. Bratcu, *State of art of assembly lines design optimisation*, Annual Reviews in Control, 26 (2002), 2, 163–174.
13. T. Sawik, *An LP-based approach for loading and routing in a flexible aseembly line*, International Journal of Production Economics, 64 (2000), 49–58.
14. T. Sawik, *Monolithic vs. hierarchical balancing ans cheduling of a flexible assembly line*, European Journal of Operational Research, 143 (2002), 115–124.
15. A. Scholl, *Balancing and Sequencing of Assembly Lines*. Physica-Verlag Heidelberg, 1999.
16. J. Szadkowski, *Critical path concept for multi-tool cutting processes optimization*, Manufacturing, Modeling, Management and Control IFAC Symposium, Elsevier, Vienna, Austria, 1997, 393–398.
17. L. Tang, D. Yip-Hoi, W. Wang, Y. Koren, *Concurrent line-balancing, equipement selection and throughput analysis for multi-part optimal line design*, CIRP 2nd International Conference on Reconfigurable Manufacring (Michigan), 2003.
18. A. Yigit, A. Ulsoy, A. Allahverdi, *Optimizing modular product design for reconfigurable manufacturing*, Journal of Intelligent Manufacturing, 13 (2002), 309–316.