Computers & Industrial Engineering 58 (2010) 393–400

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A MIP approach for balancing transfer line with complex industrial constraints

Mohamed Essafi, Xavier Delorme, Alexandre Dolgui *, Olga Guschinskaya

Centre for Industrial Engineering and Computer Science, Ecole des Mines de Saint Etienne, 158, cours Fauriel, Saint Etienne, France

A R T I C L E I N F O

Article history:
Available online 24 April 2009

Keywords:
Machining line balancing
Sequence-dependent setup time
Parallel machines
Accessibility constraints
MIP

A B S T R A C T

This paper deals with a novel line balancing problem for flexible transfer lines composed of identical CNC machines. The studied lines are paced and serial, i.e. a part to be machined passes through a sequence of workstations. At least one CNC machine is installed at each workstation. The objective is to assign a given set of operations required for the machining of the part to a sequence of workstations while minimizing the total number of machines used. This problem is subject to precedence, exclusion and inclusion constraints. In addition, accessibility has to be considered. Moreover, the workstation workload depends on the sequence in which the operations are assigned because of setup times related to the change and displacement of tools, rotation of the part, etc. It is a novel line balancing problem, and we highlight its particularities by reviewing the close problems existing in the literature. Then, a mathematical model as a mixed-integer program is suggested. A procedure for computing ranges for variables is given. Experimental computations with ILOG Cplex are reported.

1. Introduction

Manufacturers are increasingly interested in optimizing their production systems. In this paper, a special case of production systems is considered: machining transfer lines. Transfer lines are widely used in the automotive industry. In such a line, a repeatable set of operations is executed each cycle time. The line is composed of sequentially arranged workstations and a transport system which ensures a constant flow of parts along the workstations. These machining lines produce large series of identical or similar items.

A significant investment is needed to install a transfer line for a given product or product family. In addition, a long design period is required. Manufacturers have to invest heavily when installing these lines. This investment influences to a great extent the cost of the finished products over the lifetime of the line. Thus, profitability directly depends on the investment cost and production efficiency determined by the line design. Therefore, optimization is a crucial issue at the line design.

The design of transfer lines is composed of several steps:

- *Product analysis*: a complete description of the operations that have to be executed for the machining products.
- *Process planning*: the processes required to transform raw parts into finished products by considering technological constraints. For instance, during process planning, a partial order between operations, inclusion and exclusion constraints are established.
- *Line configuration*: defines the configuration design and resolves the balancing problem, i.e. the allocation of operations to workstations in order to obtain the necessary production rate while meeting demand and achieving the quality required. The line cycle time is calculated by considering the objective production rate. A security margin often has to be considered in order to take into account times to failure and times to repair of machines and also possible modifications on the initial product. It is imperative to consider all the constraints, particularly, those of precedence.
- *Dynamic flow analysis and transport system design*: products flow is studied via simulation to take into account random events and variability in production. The material handling system is selected and optimized.
- Detailed design and line implementation.

In this work, only the step "line configuration and balancing problem" will be studied. As a rule, the configuration of a machining line involves two principle phases: (i) choice of line type and (ii) logical synthesis of the manufacturing process which consists of grouping the operations into workstations (*line balancing*). Only the second phase is treated here; all parameters and input required for this problem are known and considered as deterministic values. The objective is to optimize the allocation of operations to the workstations.

A transfer line balancing problem with parallel workstations, sequence-dependent setup times and accessibility constraints is studied. A transfer line is equipped with a set of identical standard CNC machines tools in which different processing operations are executed. Each operation is characterized by an operational time; a set of operations which must be assigned before it (precedence

* Corresponding author. Tel.: +33 0 4 77 42 01 66; fax: +33 0 4 77 42 66 66.
E-mail addresses: essafi@emse.fr (M. Essafi), delorme@emse.fr (X. Delorme), dolgui@emse.fr (A. Dolgui), guschinskaya@emse.fr (O. Guschinskaya).

0360-8352/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2009.04.009

constraints); a set of operations which must be executed on the same workstation (inclusion constraints); a set of operations which cannot be executed on the same workstation (exclusion constraints). Our main objective is to minimize the total number of machines in the line for a given cycle time. This work was developed in collaboration with the enterprise PCI–SCEMM. The industrial case study has been presented and the problem has been defined in (Delorme, Dolgui, Essafi, Linxe, & Poyard, 2009).

The specificity of the studied problem consists in the necessity to take into account:

(i) Setup times between the operations, due to various causes, namely the change and displacement of the tool, rotation of the part, etc. They vary according to the sequence in which the operation is assigned.
(ii) Accessibility constraints are related to the position of the part on the machine. For every position there corresponds a set of operations which can be executed.
(iii) The utilization of parallel workstations, leads to a better balancing. In addition, when an operation has an operational time higher than the cycle time, it is still possible to respect the imposed cycle time of the line.

The paper consists of five sections. In Section 2, we present a survey of related works and pay particular attention to the problems with parallel machines and sequence-dependent setup times. In Section 3, we present the problem and specify its characteristics. In Section 4, a mixed-integer program is proposed for this problem as well as a procedure for computing ranges for the corresponding decision variables. In Section 5, experimental results are reported.

2. Related works

The line balancing (assignment of operation to workstations) is an important problem in the design of flow lines. Historically, the line balancing problem was first studied for assembly lines. As far as we know, the earliest publication on assembly line balancing was presented by Salveson (1955).

Baybars (1986) summarized the assumptions of the assembly line balancing problem and proposes a review of the developed methods to solve it. All of considered problems are deterministic and the resolution algorithms are exact. The author distinguished two traditional problems, namely, the simple assembly line balancing problem (SALBP) and the generalized assembly line balancing problem (GALBP). SALBP includes problems which corresponding to a set of predefined characteristics while GALBP includes problems with additional and specific characteristics, such as multi-product ALBP, equipments selection ALBP, etc. Even the SALBP is NP-hard, (see, for example Bhattachajee & Sahu, 1990), but, due to the many applications of this problem, an important body of literature has been published on this topic. A recent and comprehensive state of the art has been presented in a special issue (Dolgui, 2006). Several articles provide broad surveys of this problem, see, for example Rekiek, Dolgui, Delchambre, and Bratcu (2002), Boysen, Fliedner, and Scholl (2008).

Unlike the ALBP, the machining line balancing problem is rather recent. It was first mentioned in Szadkowski (1997). Then, in Dolgui, Guschinsky, and Levin (1999), it was defined for dedicated transfer lines and called the transfer line balancing problem (TLBP). In most of the works that consider these transfer line balancing problems, the lines are equipped by multi-spindle heads machines. Several exact and approximate methods for TLBP have been presented. The most significant methods for an exact resolution of the TLBP are (i) linear programming for the logical layout design of modular machining lines (Belmokhtar, Dolgui, Guschinsky, & Levin, 2006), a dynamic programming (Dolgui et al., 1999) and a mixed integer approach (Dolgui, Finel, Guschinsky, Levin, & Vernadat, 2006) to balancing machining lines with blocks of parallel operations, (ii) a Branch and Bound procedure for a design problem of transfer lines composed of workstations with sequentially activated multi-spindle heads (Dolgui & Ihnatsenka, 2009). For large scale problems, several approximate methods have been designed: (i) priority rules heuristics (Finel, Dolgui, & Vernadat, 2008; Guschinskaya, Dolgui, Guschinsky, & Levin, 2008) and (ii) metaheuristics, for example, see Dolgui, Guschinskaya, and Eremeev (2008), a GRASP method and a genetic algorithm have been proposed for the transfer lines balancing problem. However, since all these cited works focused usually on multi-spindle heads and dedicated machines without tool changes, they did not consider setup times or accessibility constraints. Moreover, while if some models considered the parallel execution of some operations on a machine (due to multi-spindle heads), they did not study the duplication of machines in order to reduce the cycle time. None of the models in the transfer line literature deals with a problem similar to the problem considered in this paper.

Among the extensions proposed for the ALBP, two are related to our problem, namely the use of parallel machines and the consideration of setup times between operations. We will now present the principal research work which has considered these two extensions.

Buxey (1974) was the first to introduce the use of parallel machines for ALBP. Workstations are equipped with parallel machines which execute the same operations on different parts of the product. The author presented the advantages of such a line: reduced nonproductive time (idle time), cycle time respected (when there are operations with an operational time exceeding the cycle time), improved production output imposed by the longest operation and, finally, transport time and matter flows on the line are reduced. Pinto, Dannenbring, and Khumawala (1975) considered the balancing problem with paralleling of workstations. They were the first to propose a resolution approach for this problem with a branch and bound algorithm. Bard (1989) proposed an extension of the problem considering dead times. A heuristic approach was proposed by Askin and Zhou (1997) for the multi-product line balancing problem with parallel workstations. McMullen and Frazier (1998) studied the multi-objective assembly line balancing problem with stochastic operational times and parallel machines. Bukchin and Rubinovitz (2002) proposed an interesting approach for assembly line design with parallel stations.

Setup times between operations were examined at first in Wilhelm (1999) as tool changes times; author proposed a column generation approach to solve this problem. Since this publication, there were a few articles on this subject, including Scholl, Boysen, and Fliedner (2008), which defined the sequence-dependent assembly line problem (SDALBP), setup times varying according to the sequence in which the operation was processed, and Andrés, Miralles, and Pastor (2008), which notably proposed a heuristic approach for this problem.

In this section, a non exhaustive survey of publications dealing with line balancing problems was presented. Although several research papers have been devoted to the study of the transfer lines balancing problem, none dealt with the constraints of our problem. Nor has any research been done on the balancing assembly lines problem considering both parallel machines and setup times between operations.

3. Problem statement

The studied machining lines are equipped with CNC machines. All the machines are identical following a line modularity principle.

In contrast with dedicated transfer lines with multi-spindle machines, each machine contains one spindle and a magazine for tools. An additional time is required for each machine to pass from one operation to the next due to tool changes and displacements or/and the rotation of the part (setup time). Taking into account the fact that a part is held at a machine with some fixtures in a given position (part fixing and clamping), some sides and elements of the part are not accessible for machining even after the part displacement or rotation. No matter what positioning and clamping is chosen, some areas on the part will be hidden or covered. Therefore, the choice of a part position for part fixing should be also considered in the optimization procedure. Fig. 1 presents an example of the studied line configuration. This line is designed to manufacture automotive cylinder heads.

In Fig. 1, the lines (1) represent the transport system which is composed of conveyors. Robots are used for part loading and unloading. The boxes (2) represent the CNC machines. Machines in a group aligned vertically represent a workstation. A workstation can comprise more than one machine; in this case, the same operations are duplicated and executed on different machines and the local cycle time of the workstation is equal to the number of parallel machines multiplied by the line cycle time. With the parallel machines at each workstation, the line is easily reconfigurable. The line cycle time can be modified, if necessary, and even be shorter than the time of an operation. The boxes (3) represent dedicated workstations for specific operations such as assembly or washing.

We consider the optimization problem of this type of line and develop a model for line balancing. The input data used is:

- Cycle time ("takt time") imposed by the objective production rate: one part is produced at each cycle.
- Precedence constraints: order relations between operations. These relations define feasible sequences of operations.
- Inclusion constraints: the need to carry out fixed groups of operations on the same workstation.
- Exclusion constraints: the impossibility of carrying out certain subsets of operations on the same workstation.
- Accessibility constraints are related to the positioning of the part; indeed, for a position some part sides may not be accessible, and thus, the operations concerning these sides cannot be carried out without repositioning. In the considered machining line, only one part fixing position is defined for each workstation (the part repositioning is made between two workstations).
- Sequence-dependent setup times: the time required for the execution of two sequential operations is not equal to the sum of their times but depends also on the order in which they are done, since the time needed for the displacement/change of tool and part rotation are not negligible.

Hence, here, we have a special case of line balancing with a sequential execution of operations, setup times, parallel machines, as well as accessibility, exclusion and inclusions constraints, and which represents a significant investment for industry.

At this point, a mixed-integer programming (MIP) model for the design of this line for a given product will be presented.

4. Mixed-integer programming (MIP) approach

4.1. Mathematical formulation

We will introduce the following notations.

Indexes:

- i, j for operations;
- q for the place (order) of an operation in the sequence of assigned operations;
- n for the number of parallel machines at a workstation;
- k for the workstations;
- a for the part fixing positions.

Parameters:

- N set of operations to be assigned $(i,j = 1, \ldots, |N|)$;
- A set of possible part positions for part fixing in a machine; only one of these positions is chosen for each workstation; a part fixing position defines the accessibility constraints for the part $(a = 1,\ldots,|A|)$;
- l_0 maximum number of operations authorized to be assigned to a workstation: each created workstation cannot contain more than l_0 operations;
- n_0 maximum number of machines to be installed in a workstation;
- m_0 maximum number of workstations;
- $q_0 = l_0 \cdot m_0$: we consider a sequence of operations and q_0 represent the maximum number of possible assignments (places) for any operation on this sequence;
- t_i operational time for operation i $(i = 1, \ldots, |N|)$;
- t_{ij} setup time needed if operation j is processed directly after operation i at the same workstation;
- T_0 objective line cycle time (takt time);
- P_i set of direct predecessors of operation i;

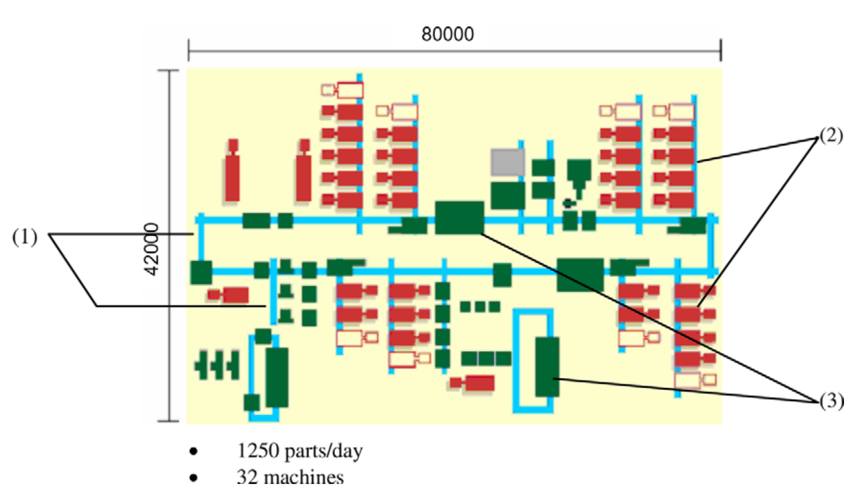

Fig. 1. Schema of a line for machining cylinder heads (PCI/SCEMM).

- P_i^* set of all predecessors of i (direct and indirect predecessors);
- F_i^* set of all successors of i (direct and indirect successors);
- ES a collection of subsets e ($e \subset N$) of operations which must be imperatively assigned to the same workstation;
- $\overline{ES}$ a set of pairs of operations (i,j) which cannot be assigned to the same workstation;
- $A(i)$ set of the possible part fixing positions for which the execution of operation i is possible;
- LB_{ws} a lower bound for the number of workstations;
- $S(k)$ set of possible places for operations at workstation k, this set is given by an interval of indexes, the maximum possible interval is: $S(k) = \{l_0(k-1)+1, l_0(k-1)+2, \ldots, l_0k\}; \forall k = 1,2,\ldots,m_0$;
- $K(i)$ set of workstations on which operation i can be processed, $K(i) \subseteq \{1,2,\ldots,m_0\}$;
- $Q(i)$ set of possible places for operation i in the sequence of all operations, $Q(i) \subseteq \{1,2,\ldots,l_0m_0\}$;
- $N(k)$ set of operations which can be processed at workstation k;
- $M(q)$ set of operations which can be assigned to the place q in the sequence;
- E_i earliest workstation to which operation i can be assigned;
- L_i last workstation to which operation i can be assigned.

Variables:

- $x_{iq} = 1$, if operation i is assigned to the qth place (q is its order in the overall assignment sequence), otherwise $x_{iq} = 0$;
- τ_q setup time required between operations assigned to the same workstation in place q and $q+1$;
- $y_{nk} = 1$ if there are n parallel machines at the workstation k, 0 otherwise;
- $z_{ka} = 1$ if for the part of the workstation k the fixing position a is used, 0 otherwise.

The optimization model is as follows:

- The objective function (1) minimizes the total number of machines:

$$\text{Minimize} \sum_{k=1}^{m_0} \sum_{n=1}^{n_0} n \cdot y_{nk} \tag{1}$$

- Eq. (2) verifies that there is only one value for the number of parallel machines on each workstation:

$$\sum_{n=1}^{n_0} y_{nk} \leqslant 1, \quad \forall k = 1,2,\ldots,m_0 \tag{2}$$

- Eq. (3) assures that a workstation is open only if the preceding workstation is also opened:

$$\sum_{n=1}^{n_0} y_{nk} \geqslant \sum_{n=1}^{n_0} y_{n(k+1)}, \quad \forall k = LB_{ws}, LB_{ws}+1, \ldots, m_0 - 1 \tag{3}$$

- Eq. (4) assures that each operation i is assigned once and only once:

$$\sum_{q \in Q(i)} x_{iq} = 1, \quad \forall i \in N \tag{4}$$

- Eq. (5) assure that a place in the sequence is occupied by only one operation:

$$\sum_{i \in M(q)} x_{iq} \leqslant 1, \quad \forall q = 1,2,\ldots,q_0 \tag{5}$$

- Eq. (6) assures that an operation (expect the first of each workstation) is assigned to a place only if another operation is assigned to the preceding place of the sequence (there is no empty place between two operations on the same workstation):

$$\sum_{i \in M(q)}^{x_{iq}} \geqslant \sum_{p \in S(k);\ p>q} \sum_{i \in M(p)} x_{ip}, \quad \forall q \in S(k) \setminus \max\{S(k)\},$$
$$\forall k = 1,2,\ldots,m_0 \tag{6}$$

- Eq. (7) verifies that only one part fixing position is chosen for each workstation:

$$\sum_{a \in A} z_{ka} \leqslant 1, \quad \forall k = 1,2,\ldots,m_0 \tag{7}$$

- Eq. (8) assures that accessibility constraints are respected (the part fixing position chosen for a workstation authorizes the execution of every operation assigned to this workstation):

$$\sum_{q \in S(k)} x_{iq} \leqslant \sum_{a \in A(i)} z_{ka}, \quad \forall k \in K(i), \ \forall i \in N \tag{8}$$

- Eq. (9) calculates the additional time between operation i and operation j when operation j is processed directly after operation i at the same workstation:

$$\tau_q \geqslant \sum_{j \in M(q+1) \setminus \{i\}} t_{ij} \cdot (x_{iq} + x_{j(q+1)} - 1),$$
$$\forall i \in M(q), \ \forall q \in S(k) \setminus \max\{S(k)\}, \ \forall k = 1,2,\ldots,m_0 \tag{9}$$

- Eq. (10) assures that the workload time of every workstation does not exceed the local cycle time which corresponds to the number of installed parallel machines at this workstation multiplied by the objective cycle time of the line:

$$\sum_{q \in S(k) \setminus \max\{S(k)\}} \tau_q + \sum_{i \in N(k)} \sum_{q \in S(k)} t_i \cdot x_{iq} \leqslant T_0 \cdot \sum_{n=1}^{n_0} n \cdot y_{nk},$$
$$\forall k = 1,2,\ldots,m_0 \tag{10}$$

- Eq. (11) defines the precedence constraints between operations:

$$1 + \sum_{q \in Q(j)} q \cdot x_{jq} \leqslant \sum_{q \in Q(i)} q \cdot x_{iq}, \quad \forall i \in N, \ \forall j \in P_i \tag{11}$$

- Eq. (12) represents the inclusion constraints:

$$\sum_{q \in S(k) \cap Q(i)} x_{iq} = \sum_{q \in S(k) \cap Q(j)} x_{jq}, \quad \forall i,j \in e,$$
$$\forall e \in ES, \ \forall k \in K(i) \cap K(j) \tag{12}$$

- Eq. (13) represents the exclusion constraints:

$$\sum_{q \in S(k)} (x_{iq} + x_{jq}) \leqslant 1, \quad \forall (i,j) \in \overline{ES}, \ \forall k \in K(i) \cap K(j) \tag{13}$$

- Eqs. (14)–(16) provide additional constraints on the possible values of variables:

$$0 \leqslant \tau_q \leqslant \max_{(i,j) \in N*N;\ i \neq j} (t_{ij}), \quad \forall q = 1,2,\ldots,q_0 \tag{14}$$
$$x_{iq} \in \{0,1\}, \quad \forall i \in N, \ \forall q \in Q(i) \tag{15}$$
$$y_{nk} \in \{0,1\}, \quad \forall n = 1,2,\ldots,n_0, \ \forall k = 1,2,\ldots,m_0 \tag{16}$$
$$z_{ka} \in \{0,1\}, \quad \forall k = 1,2,\ldots,m_0, \ \forall a \in A \tag{17}$$

4.2. Computing ranges for variables

Model (1)–(17) can be solved using ILOG Cplex or an other standard. Nevertheless, since this problem is NP-hard, the calculation time can become prohibitive. The resolution time for this model can be decreased using efficient techniques to reduce the number of variables (the size of the model) and consequently to accelerate the search of an optimal solution.

At the beginning, the different sets ($K(i)$, $N(k)$, $S(k)$, $Q(i)$ and $M(q)$) are initialized to the largest possible subset. We propose a technique for calculating bounds for the possible indexes for the variables of the mathematical model. This can simplify the problem and thus lessen calculation time.

Taking into account the different constraints between operations, we can calculate the sets $K(i)$, $N(k)$, $S(k)$, $Q(i)$ and $M(q)$ more precisely. Note that these sets give intervals of possible values for the corresponding indexes.

The following additional notations can be defined:

- $E_i[r]$ is a recursive variable for the step by step calculation of the value of E_i taking into account setup times etween operations, $r \in \{0,1\}$;
- $L_i[r]$ is a recursive variable for the step by step calculation of the value of L_i taking into account setup times between operations, $r \in \{0,1\}$.

With P_i^* which is the set of all predecessors of operation i, and F_i^* which is the set of all successors of operation i, we can also introduce:

- $Sp_i[r]$: the sum of the $(|P_i^*| - E_i[r] + 1)$ shortest setup times among the operations of the set $P_i^* \cup \{i\}$ composed of operation i and all its predecessors, $i \in N$;
- $Sf_i[r]$: the sum of the $(|F_i^*| - m_0 + L_i[r])$ shortest setup times among the operations of the set $F_i^* \cup \{i\}$ composed of operation i and all its successors, $i \in N$;
- $d[i,j]$: a parameter (distance) which has the following property: if $\{i,j\} \in \overline{ES}$, then $d[i,j] = 1$, else $d[i,j] = 0$.

The total operational time T_{sum} without considering the setup times between operations is calculated as follows:

$$T_{sum} = \sum_{i \in N} t_i$$

A lower bound on the number of workstations can be calculated by supposing that each workstation contains n_0 machines. Therefore, the local cycle time of each workstation is equal to $(T_0 \cdot n_0)$. The line becomes a serial line composed of identical workstations with a cycle time which is equal to $(T_0 \cdot n_0)$. Then, a lower bound on the number of workstations LB_{ws} can be calculated as follows:

$$LB_{ws} = \lceil T_{sum}/(T_0 \cdot n_0) \rceil$$

Notation. $\lceil x \rceil$ is the lowest integer value higher or equal to x.

In the same way, a lower bound on the number of machines in the line (LB_m) can be determined by the following expression:

$$LB_m = \lceil T_{sum}/T_0 \rceil$$

The following procedure calculates the sets $K(i), Q(i), M(q), N(k)$ and $S(k)$. Note: the operations are numbered in the order of precedence graph ranks (in topological order). Some lines are annotated with comments. The symbol "//" is used to mark the beginning and the end of comments.

Algorithm.

Step 1// calculation of E_i and L_i taking into account precedence constraints and setup times//
for all $i \in N$ **do**
  $E_i[0] \leftarrow \left\lceil (t_i + \sum_{j \in P_i^*} t_j)/(n_0 \cdot T_0) \right\rceil$; // calculate the earliest workstation $E_i[0]$ on which operation i can be processed taking into account the precedence constraints//
  $L_i[0] \leftarrow m_0 - \left\lceil (t_i + \sum_{j \in F_i^*} t_j)/(n_0 \cdot T_0) \right\rceil + 1$; // calculate the latest workstation $L_i[0]$ on which operation i can be processed considering the precedence constraints//
  $E_i[1] \leftarrow \left\lceil (t_i + Sp_i[0] + \sum_{j \in P_i^*} t_j)/(n_0 \cdot T_0) \right\rceil$; // calculate $E_i[1]$ which are new values of E_i obtained by taking into account in addition setup times between operations//
  $L_i[1] \leftarrow m_0 - \left\lceil (t_i + Sf_i[0] + \sum_{j \in F_i^*} t_j)/(n_0 \cdot T_0) \right\rceil + 1$; // calculate $L_i[1]$ which are new values of L_i obtained by taking into account in addition setup times between operations//
  if $E_i[1] \neq E_i[0]$ **then**
  $E_i \leftarrow \max\left(E_i[0] + 1, \left\lceil (t_i + Sp_i[1] + \sum_{j \in P_i^*} (t_j))/(n_0 \cdot T_0) \right\rceil \right)$
  else $E_i \leftarrow E_i[1]$; // updating the values of E_i//
  if $L_i[1]; \neq L_i[0]$ **then**
  $L_i \leftarrow \min\left(L_i[0] - 1, m_0 - \left\lceil (t_i + Sf_i[1] + \sum_{j \in F_i^*} t_j)/(n_0 \cdot T_0) \right\rceil + 1\right)$
  else $L_i \leftarrow L_i[1]$; // updating the values of L_i//
end for
Step 2 // calculation of E_i taking into account exclusion and inclusion constraints//
$j_{cur} \leftarrow 1$;
do
  $j_{min} \leftarrow j_{cur}$; $j_{cur} \leftarrow |N|$;
// new values of E_i are calculated by considering exclusion constraints//
  for $j \leftarrow j_{min} + 1, \ldots, |N|$ **do** $E_j \leftarrow \max(\max_{i \in P_j^*}\{E_i + d[i,j]\}, E_j)$;
  end for;
  for each $e \in ES$ **do**
    $E_e \leftarrow \max_{j \in e}(E_j)$;
    for each $j \in e$ **do if** $E_j < E_e$ **then**
      $E_j \leftarrow E_e$;// E_i is calculated taking into account an inclusion constraint//
      $j_{cur} \leftarrow \min\{j_{cur}, j\}$;
    end for;
  end for;
until $j_{cur} = |N|$.
Step 3 //calculation of L_i taking into account inclusion and exclusion constraints//
$j_{cur} \leftarrow |N|$;
do
  $j_{max} \leftarrow j_{cur}$; $j_{cur} \leftarrow 1$; // new values of L_i are calculated by considering exclusion constraints//
  for $j \leftarrow j_{max} - 1, \ldots, 1$ **do** $L_j \leftarrow \min(\min_{i \in F_j^*}\{L_i - d[j,i]\}, L_j)$;
  end for;
  for each $e \in ES$ **do**
    $L_e \leftarrow \min_{j \in e}(L_j)$;
    for each $j \in e$ **do if** $L_j > L_e$ **then**
      // L_i calculated taking into account inclusion constraints//
      $L_j \leftarrow L_e$;
      $j_{cur} \leftarrow \max\{j_{cur}, j\}$;
    end for;
  end for;
until $j_{cur} = 1$.
Step 4 // calculation of the sets $K(i), N(k), S(k), Q(i)$ and $M(q)$//
for all $i \in N$ **do** $K(i) \leftarrow [E_i, L_i]$; **end for**;
for $k \leftarrow 1, 2, \ldots, m_0$ **do**

$N(k) \leftarrow \{i | i \in N, k \in K(i)\}$;
$S(k) \leftarrow \left[1 + \sum_{k'=1}^{k-1} |S(k')|, \min(|N(k)|, l_0) + \sum_{k'=1}^{k-1} |S(k')|\right]$;
end for;
for all $i \in N$ **do** $Q(i) \leftarrow [\min\{S(E_i)\}, \max\{S(L_i)\}]$; **end for;**
for $q \leftarrow 1, 2, \ldots, \max\{S(m_0)\}$ **do** $M(q) \leftarrow \{i | q \in Q(i)\}$; **end for**;
End of algorithm.

4.3. Illustration

In order to better explain the suggested algorithm, we present a numerical example with 15 operations. Fig. 2 shows the precedence graph and operational times.

The objective line cycle time is: $T_0 = 16$ units of time; the maximum number of workstations $m_0 = 6$; the maximum number of machines to be installed on a workstation $n_0 = 3$; the maximum number of operations to be assigned to a workstation $l_0 = 8$.

The inclusion constraints are : $ES = \{(2,4);(6,8);(5,7);(11,12);(14,15)\}$.
The exclusion constraints are : $\overline{ES} = \{(2,7);(1,4);(5,15);(6,7);(8,14)\}$.

The setup times are reported in Table 1. For example, the setup time $t_{4,5} = 3$ corresponds to the setup time that is required if operation 5 is performed immediately after operation 4.

The total operational time, $T_{sum} = \sum_{i \in N} t_i = 161$ units of time.

A lower bound on the number of workstations is: $LB_{ws} = \lceil T_{sum}/(T_0 * n_0) \rceil = \lceil 161/(16 * 3) \rceil = 4$.

Thus, the optimal solution cannot have less than four workstations.

A lower bound on the number of machines is: $LB_m = \lceil T_{sum}/T_0 \rceil = \lceil 161/16 \rceil = 11$.

Then, the optimal solution cannot have less than 11 machines.

Now, the procedure of range calculation for indexes is applied:

Step 1: The initial values of E_i and L_i for each operation are calculated considering setup times between operations and precedence constraints (see Fig. 3).

Step 2: The new values of E_i are obtained by considering the exclusion and inclusion constraints (see Fig. 4).

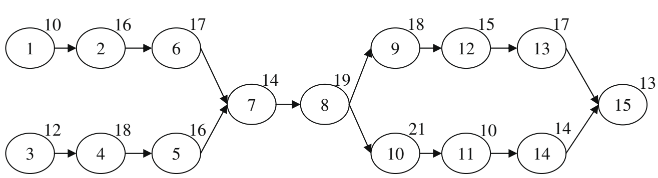

Fig. 2. Precedence graph.

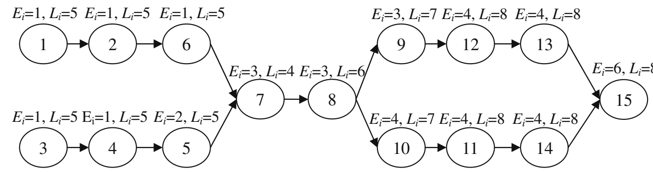

Fig. 3. Values of E_i and L_i obtained considering precedence constraints and setup temps.

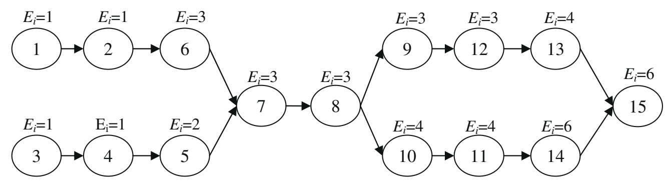

Fig. 4. Values of E_i considering exclusion and inclusion constraints.

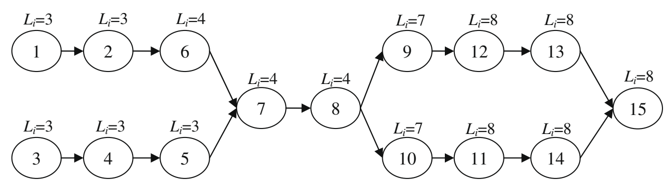

Fig. 5. Values of L_i considering exclusion and inclusion constraints.

Step 3: The new values of L_i are calculated by considering the exclusion and inclusion constraints (see Fig. 5).

Step 4:

Sets $K(i)$ for operations $i \in N$ are obtained (Table 2).

Sets of operations $N(k)$ for workstations $k = 1, 2, \ldots, m_0$ are defined (Table 3).

Range of places $S(k)$ for operations of workstation k is calculated, $k = 1, 2, \ldots, m_0$ (Table 4).

Finally, range of places $Q(i)$ for operation i is found, for all $i \in N$ (Table 5).

Applying this procedure to the MIP model allow to decrease the number of variables from 1096 to 758 and the number of constraints from 1278 to 911.

5. Experimental results

In order to evaluate the performance of the model, several instances have been randomly generated with characteristics close to those of actual industrial problems. All these instances have

Table 1
Setup times.

i	j														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	–	4	4	3	2	1	1	2	1.5	1.5	1.6	3.6	2.8	2.4	1.2
2	5	–	1.5	1	1	2.5	3	3	3	3.5	4	4.3	4.9	4.9	3.5
3	1.5	3.5	–	2.5	1.2	3.4	4	4.2	3	2.2	4.3	3.8	2.9	1.8	3.9
4	4.5	4	3	–	3	1.5	3	2	4.5	4	4.9	4	2.3	1.6	1.8
5	4	4	5	5	–	2.5	2	2	4.5	1	1.6	4.4	4.2	3.3	1.7
6	25	1.5	3	1.2	1	–	2	2	3	4	4.3	1.1	3	3.6	2.6
7	4	3.8	3	1.8	4	3	–	4.7	4	1.4	2.5	1.7	4	4.3	3
8	3	2.5	4	3.4	3.2	2.4	1.6	–	2	4	2.2	4.7	2.1	2.3	1.5
9	3	4.9	4	2.3	1.6	3.6	3	1.2	–	3	1.4	2.8	4	3.7	3.1
10	1.5	4	1.5	4.6	3.7	2.2	2.7	1.2	4.8	–	1.7	2.3	1.6	4.9	2.8
11	4.7	1.1	4.5	2.1	1.8	4	2.6	1.8	3.4	3.4	–	2.8	3.3	3.5	4.4
12	1.7	3.9	3.2	4.6	1.9	1.7	3.4	3.7	3.3	2.4	2.9	–	3.9	3.5	1.4
13	4.6	3.4	3.8	1.6	3.6	4.8	2.2	3.2	2.2	1.6	1.4	4.4	–	3.9	1.6
14	3	2.7	2.6	2.1	2.8	3	1.6	2.2	4.2	4.8	4.4	3	1.2	–	3
15	3.4	4.8	1.5	4.4	3.6	2.2	1.4	3	1.8	2.1	4.6	3.2	3.6	1.4	–

Table 2
The ranges $K(i)$ for the operations.

Operation i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$K(i)$	[1,3]	[1,3]	[1,3]	[1,3]	[2,3]	[3,4]	[3,4]	[3,4]	[3,7]	[4,7]	[4,8]	[4,8]	[4,8]	[6,8]	[6,8]

Table 3
The set of operations $N(k)$ for each workstation.

St. k	1	2	3	4	5	6	7	8
$N(k)$	{1,2,3,4}	{1,2,3,4,5}	{1,2,3,4,5,6,7,8,9}	{6,7,8,9,10,11,12,13}	{9,10,11,12,13}	{9,10,11,12,13,14,15}	{9,10,11,12,13,14,15}	{11,12,13,14,15}

Table 4
The ranges of places $S(k)$ for workstations.

St. k	1	2	3	4	5	6	7	8
$S(k)$	[1,4]	[5,9]	[10,17]	[18,25]	[26,30]	[31,37]	[38,44]	[45,49]

Table 5
The ranges of places $Q(i)$ for operations.

Op. i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$Q(i)$	[1,17]	[1,17]	[1,17]	[1,17]	[5,17]	[10,25]	[10,25]	[10,25]	[10,25]	[18,44]	[18,49]	[18,49]	[18,49]	[31,49]	[31,49]

been solved on a SUN UltraSPARC IIIi with 1593 MHz CPU and 16 GB of memory using ILOG CPLEX 11.0 with default parameters. The objective of the experimental study was to (i) evaluate the impact of the procedure proposed to limit the ranges of variables and (ii) to observe the influence of two following problem characteristics: the number of operations in the set N and the density D_p of the precedence graph, which is a measure introduced by Scholl (1999) and defined as follows:

$$D_p = \frac{2 * \sum_{i \in N} |P_i^*|}{|N| * (|N| - 1)},$$

This ratio is equal to zero when no precedence constraints exist between operations and is equal to one if only a sole sequence is possible. The quality of the algorithm is measured with the computational time (in seconds) which corresponds to the time to find and prove the optimality of a solution, as well as by the gap between the best solution found and the lower bound when ILOG Cplex was not able to solve the problem optimally under a given time (i.e., 10,000 s).

To generate the instances, different numbers of operations (10, 12, 14, 16, 18 and 20) and different density values (D_p) for the precedence constraints (10%, 25% and 40%) have been considered. The inclusion and exclusion constraints have been generated with a fixed density equal to 3% and 2%, respectively. The cycle time T_0, the maximum number of machines by workstations n_0 and the maximum number of workstations m_0 on the line varied according the number of operations in each considered instance. For each set of data, we generate randomly 10 different instances.

The results of our experiments are presented below. First, in Table 6 the variation of the average number of variables used in MIP in function of the number of operations of the problem (before the reduction procedure). Since this number does not depend on the density D_p, only one value is reported for each number of operations.

Table 6
Average number of variables.

# Operations	10	12	14	16	18	20
# Variables	560	773	1014	1288	1440	1753

Table 7
Average variation of the number of variables with the reduction procedure (%).

D_p (%)	#Op.					
	10	12	14	16	18	20
10	−3.5	−3.15	−5.24	−4.25	−4.2	−4.4
25	−9.7	−9.13	−10	−7.14	−7.11	−8.65
40	−10.3	−14.5	−15.3	−8.66	−10.5	−7.3

As expected, the number of variables of the model seems to grow linearly with the number of operations treated.

Then the impact of the reduction procedure on the average number of variables can be examined. These results are reported in Table 7 in which the percentage of reduction obtained is indicated in function to the number of operations and the density D_p of the problem.

Independently of the number of operations, the impact of the reduction procedure is increasingly significant with a high value of density for the precedence graph, except for the instance with 20 operations ($D_p = 25\%$/variation $= -8.65\%$, $D_p = 40\%$/variation $= -7.3\%$). Note: for cases with low value of $D_p(5\%;\ 10\%)$, the reduction procedure has a very low impact on the initial model (i.e., between 3% and 5% of reduction). In addition, if there is no major trend showing an impact of the number of operations on the effectiveness of the reduction procedure, there is an important variability to the input data.

Finally, Table 8 reports the results obtained with Cplex on all the instances using the reduction procedure. For each instance, the average computational time (CPU Time) is indicated, as well as the final gap between the best solution produced and the lower bound (this gap is equal to 0% when the instance has been solved within the 10,000 s limit), and finally, the number of resolved instances.

The computational time increases strongly with the number of operations. Also the resolution is easier when the density D_p is important, which is logical considering the impact of the reduction

Table 8
Average CPU time, average gap and number of instances resolved with Cplex.

D_p (%)	#Op.					
	10	12	14	16	18	20
10	467.8 s	4701 s	6524.6 s	8467.1 s	10,000 s	10,000 s
	0%	4%	9.17%	12%	19.2%	19.6%
	10	8	4	2	0	0
25	181.9 s	2173.2 s	5931.5 s	7433 s	9730 s	10,000 s
	0%	1.25%	6%	9%	18%	22%
	10	9	5	3	1	0
40	146.3 s	543.1 s	2020.3 s	7503.1 s	10,000 s	10,000 s
	0%	0%	1.5%	11.8%	16.9%	21.2%
	10	10	8	3	0	0

procedure. All the instances with 10 operations were solved in a reasonable computational time ($< 10,000$ s), as well as all instances with 12 operations and 40% for the precedence density and most instances with 12 operations (80% for instances with $D_p = 10\%$ and 90% for instances with $D_p = 25\%$). It is important to note that for several cases which were not solved in time, the gap correspond to only one machine (gap = 11.11%, 9.09%).

6. Conclusions and perspectives

In today's competitive business environment, manufacturers have opted for flexible transfer lines based on CNC machines. Optimizing the configuration of this type of line is crucial. In this paper, a novel transfer line balancing problem was studied and a model formulated. A MIP resolution approach was proposed. The lines considered are equipped with identical standard CNC machines. Parallel machines are authorized on workstations, accessibility constraints as well as setup times between operations are considered. As stated in the paper, a great number of technological constraints were taken into account. Considering the increase in complexity of the problem, a great deal of time and effort were required to completely comprehend this sophisticated problem and to model it as a mixed-integer program. A pre-processing procedure to reduce the number of variables was also developed. The model is implemented and tested. Experimental results are reported. Our model seems to be able to solve instances up to 18 operations in an acceptable time, especially when the density of the precedence graph is high enough. Indeed, the impact of the reduction procedure appears nearly negligible when the precedence graph has a very low density. This is logical since it mainly uses the precedence constraints. Since industrial problems can involve more than 100 operations, their resolution is not yet possible with this approach and more work is needed.

Further research will concern the improvement of the model and/or procedure to consider larger instances. However, the problem is NP-hard. Therefore the exact resolution of large instances could prove to be impossible within a reasonable computational time. Consequently, the use of a heuristic may be interesting. Following a framework which has already been successfully applied to close problems in (Guschinskaya et al., 2008 & Dolgui et al., 2008), the proposed MIP could be advantageously used within this heuristic to solve sub-problems. Beside the resolution of this optimization problem, the integration of such a resolution method in a complete decision support system which integrates simulation model. Such a system would take into account the unreliability of the machines.

Acknowledgement

The authors would appreciate the support of PCI–SCEMM and help of Chris Yukna for English.

References

Andrés, C., Miralles, C., & Pastor, R. (2008). Balancing and scheduling tasks in assembly lines with sequence-dependent setup times. *European Journal of Operational Research, 187*, 1212–1223.
Askin, R. G., & Zhou, M. (1997). A parallel workstation heuristic for the mixed-model production line balancing problem. *International Journal of Production Research, 35*(11), 3095–3105.
Bard, J. F. (1989). Assembly line balancing with parallel workstations and dead time. *International Journal of Production Research, 27*(6), 1005–1018.
Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing problem. *Management Science, 32*(8), 909–932.
Belmokhtar, S., Dolgui, A., Guschinsky, N., & Levin, G. (2006). An integer programming model for logical layout design of modular machining lines. *Computers and Industrial Engineering, 51*(3), 502–518.
Bhattachajee, T. K., & Sahu, S. (1990). Complexity of single model assembly line balancing problems. *Engineering Costs and Production Economics, 18*, 203–214.
Boysen, N., Fliedner, M., & Scholl, A. (2008). Assembly line balancing: Which model to use when? *International Journal of Production Economics, 111*, 509–528.
Bukchin, J., & Rubinovitz, A. (2002). A weighted approach for assembly line design with workstation paralleling and equipment selection. *IIE Transactions, 35*, 73–85.
Buxey, G. M. (1974). Assembly line balancing with multiple workstations. *Management Science, 20*, 1010–1021.
Delorme, X., Dolgui, A., Essafi, M., Linxe, L., & Poyard, D. (2009). Machining Lines Automation. In S. Y. Nof (Ed.), *Springer handbook of automation*. Springer.
Dolgui, A., Guschinsky, N., & Levin, G. (1999). On problem of optimal design of transfer lines with parallel and sequential operations. In J. M. Fuertes (Ed.), *Proceedings of the 7th IEEE international conference on emerging technologies and factory automation* (ETFA'99), October 18–21, 1999 (Vol. 1, pp. 329–334). Barcelona, Spain: IEEE.
Dolgui, A. (Ed.), Feature cluster on the balancing of assembly and transfer lines. *European Journal of Operational Research, 168*(3), 663–951, 2006.
Dolgui, A., Finel, B., Guschinsky, N., Levin, G., & Vernadat, F. (2006). MIP approach to balancing transfer lines with blocks of parallel operations. *IIE Transactions, 38*, 869–882.
Dolgui, A., & Ihnatsenka, I. (2009). Branch and bound algorithm for a transfer line design problem: Workstations with sequentially activated multi-spindle heads. *European Journal of Operational Research, 197*(3), 1197–1132.
Dolgui, A., Guschinskaya, O., & Eremeev, A. (2008). MIP-based GRASP and genetic algorithm for balancing transfer lines. In Proceedings of the Matheuristics 2008: The second international workshop on model based metaheuristics, June 16–18, 2008 (pp. 22). Italy: University Residential Centern Bertinoro (Forlì-Cesena), *Operations Research/Computer Science Interfaces Series*: Springer-Verlag.
Finel, B., Dolgui, A., & Vernadat, F. (2008). A random search and backtracking procedure for transfer line balancing. *International Journal of Computer Integrated Manufacturing, 21*(4), 376–387.
Guschinskaya, O., Dolgui, A., Guschinsky, N., & Levin, G. (2008). A heuristic multi-start decomposition approach for optimal design of serial machining lines. *European Journal of Operational Research, 189*, 902–913.
McMullen, P. R., & Frazier, G. V. (1998). Using simulated annealing to solve a multiobjective assembly line balancing problem with parallel workstations. *International Journal of Production Research., 39*(10), 2717–2741.
Pinto, P. A., Dannenbring, D. G., & Khumawala, B. M. (1975). A branch and bound algorithm for assembly line balancing with paralleling. *International Journal of Production Research., 13*(2), 183–196.
Rekiek, B., Dolgui, A., Delchambre, A., & Bratcu, A. (2002). State of the art of assembly lines design optimisation. *Annual Reviews in Control, 26*(2), 163–174.
Salveson, M. E. (1955). The assembly line balancing problem. *Journal of Industrial Engineering, 6*(4), 18–25.
Scholl, A. (1999). *Balancing and sequencing of assembly lines*. Physica-Verlag Heidelberg.
Scholl, A., Boysen, N., & Fliedner, M. (2008). The sequence-dependent assembly line balancing problem. *OR Spectrum, 30*(3), 579–609.
Szadkowski, J. (1997). Critical path concept for multi-tool cutting processes optimization. In Kopacek, P. (Ed.), *Manufacturing systems modeling, management and control: Proceedings of the IFAC workshop* (pp. 393–398). Vienna, Austria: Elsevier.
Wilhelm, W. E. (1999). A column-generation approach for the assembly system design problem with tool changes. *International Journal of Flexible Manufacturing Systems, 11*, 177–205.