



HAL
open science

Graph approach for optimal design of transfer machine with rotary table

Alexandre Dolgui, Nikolai Guschinsky, G. Levin

► **To cite this version:**

Alexandre Dolgui, Nikolai Guschinsky, G. Levin. Graph approach for optimal design of transfer machine with rotary table. *International Journal of Production Research*, 2009, 47 (2), pp.321-341. 10.1080/00207540802425880 . hal-00387687

HAL Id: hal-00387687

<https://hal.science/hal-00387687>

Submitted on 21 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Graph approach for optimal design of transfer machine with rotary table

A. DOLGUI*[†], N. N. GUSCHINSKY[‡] and G. M. LEVIN[‡]

[†]Centre for Industrial Engineering and Computer Science,
Ecole des Mines de Saint Etienne, Saint Etienne, France

[‡]Operations Research Laboratory, United Institute of Informatics Problems of the
National Academy of Sciences, Minsk, Belarus

A line balancing problem for transfer machines with rotary tables is considered. This type of machine is used in mass production. A part is sequentially machined on m working positions and is moved from one position to the next using a rotary table. The operations are grouped into blocks, where the operations of the same block are simultaneously performed by one piece of equipment with several tools (a multi-spindle head). All multi-spindle heads of a machine are simultaneously activated. Therefore, all operations of the machine are executed in parallel on m parts mounted on the rotary table. The line balancing problem consists of partitioning the set of all operations into sub-sets in order to minimize the number of working positions and the total number of spindle heads while satisfying all the constraints (precedence, compatibility for spindle heads, etc.). The method proposed in this paper is based on transforming this line balancing problem into a search for a constrained shortest path. An algorithm for simultaneously generating a graph and finding a constrained shortest path is developed. Some dominance rules for reducing the graph size are provided. An industrial example is presented in detail and experimental results on other industrial instances are reported.

Keywords: Transfer machines; Balancing; Optimization; Constrained shortest path

1. Introduction

In this paper, we deal with an automatic transfer machine with rotary table (hereafter to be called also rotary transfer machine) (Dashchenko 2003). In this type of machine (see figure 1) a part is machined sequentially on m working positions. Each working position is equipped with one or two multi-spindle heads (each contains several cutting tools). The parts are moved periodically from one position to the next by means of a rotary table. One additional position is usually used for loading and unloading the parts.

*Corresponding author. Email: dolgui@emse.fr

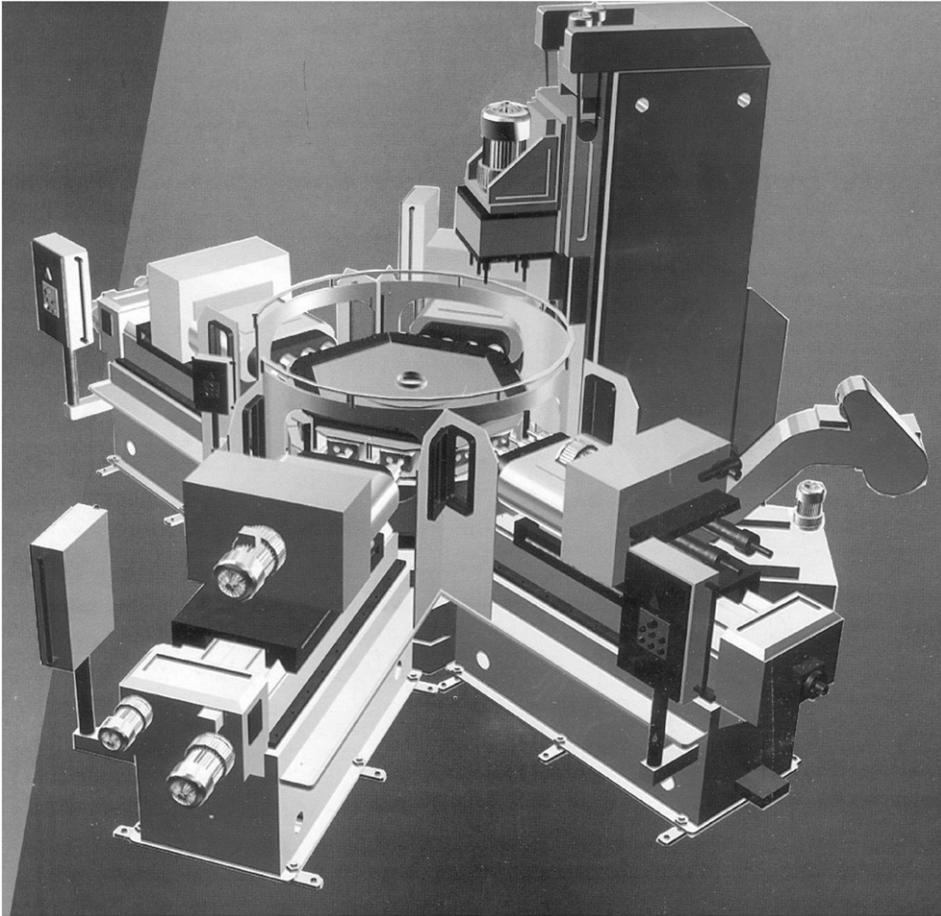


Figure 1. A transfer machine with rotary table.

The automatic rotary transfer machines are expensive and designed for homogenous products of a given type. Normally, the expected life cycle for such a machine configuration is from 5 to 7 years of production. Therefore, the search for an effective and, if possible, an optimal design is an important issue and influences profits.

This type of machine involves three main stages in the design process:

- Manufacturing process design, i.e. choice of operations for machining all part elements and assignment of these operations to working positions and spindle heads.
- Physical layout design for the machine.
- Spindle heads and working positions design (design or choice of fixtures, jigs, etc.).

If at any stage there is no acceptable solution, the user returns to one of the previous stages, i.e., this process may be performed several times, iteratively.

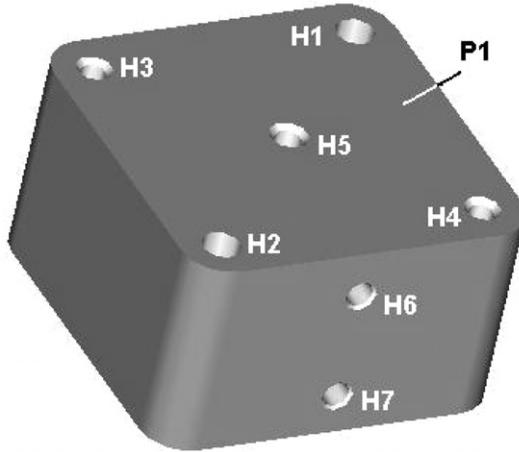


Figure 2. An example of machined part.

In this paper we deal with the first stage and only when all the operations have been chosen already. We assume that the following input data is given:

- The operations which should be executed by the machine.
- Precedence constraints for the machining operations.
- Constraints related to physical layout as well as to spindle heads and working positions design.

We consider the rotary transfer machine with vertical and horizontal spindle heads. In such a machine there is only one vertical spindle head common for all working positions. There are several horizontal spindle heads. However, there is only one horizontal spindle head per position.

Please note that there is another type of rotary transfer machine, not treated here, with only vertical spindle heads. In that case, such a machine can have one vertical spindle head at each position. Nor can horizontal spindle heads be installed on such a machine.

Consider a rotary transfer machine with one vertical and several horizontal spindle heads. Usually, the manufacturing process includes milling, drilling, boring, etc. a set of part elements such as planes, facets, holes of different types (cylindrical, bevel, threaded, etc.). Each element corresponds to a certain side (or surface) of the part. We distinguish the horizontal and lateral sides. Only one horizontal side of a part is accessible in this machine.

An example of a part for such a machine is shown in figure 2.

Let \mathbf{N} be the set of operations needed to machine all the part elements, i.e., the set of all the operations which should be executed by the machine. The set \mathbf{N} consists of subsets \mathbf{N}^r , $r = 1, 2, \dots, q$, which correspond to elements of the r th side of the part. Later, we assume that the set \mathbf{N}^1 includes operations of the accessible horizontal side and these operations are executed by the vertical spindle head. Other operations are executed by one or several ‘horizontal’ spindle heads.

A set of operations executed by the same spindle head at the given working position is called ‘block of operations’. All operations of the same block are

performed simultaneously by the corresponding spindle head. The operations of different lateral sides cannot be included in the same block, because they cannot be executed by a sole horizontal spindle head.

At the working position number zero, a finished part is unloaded and a new blank (billet) is loaded. At the k th working position, a subset N_k of operations of the set \mathbf{N} is performed, $k = 1, 2, \dots, m$. We consider the case when all the operations of $N_k \cap \mathbf{N}^r$ (belonging to the same r th side of the part and assigned to the same working position k) are executed by the same spindle head, $r = 1, 2, \dots, q$. Then each set N_k is uniquely partitioned into n_k blocks of operations (N_{kl} , $l = 1, 2, \dots, n_k$). For these machines, the number of blocks in each working position does not exceed 2. Moreover, $n_k = 1$ if the set N_k does not include operations from \mathbf{N}^1 . Characteristics of each spindle head and its execution time depend both on the set of executed operations and their parameters (cutting modes, working stroke lengths, etc.). A cycle time is the period between two movements of the table. All operations of the set \mathbf{N} are executed during this time. The simultaneous execution of all operations of the same block and all blocks of the same position means that for a cycle all operations of the set \mathbf{N} are executed in parallel on m parts mounted on the rotary table. To execute all the operations of the set \mathbf{N} on a particular part it is necessary to perform m cycles.

For the assignment of operations, the following decision should be made: how to partition the set \mathbf{N} into sub-sets (working positions) in order to minimize the number of working positions and the total number of spindle heads while satisfying all the constraints. This design problem can be approached like those of assembly lines. In fact, these similar problems for assembly lines are known as line balancing problems (Askin and Standridge 1993, Nof *et al.* 1997, Scholl 1999).

In literature, the most well known line balancing problem is Simple Assembly Line Balancing of Type 1 (SALBP-1). For this problem, the set \mathbf{N} of all operations is known. The precedence constraints for operations and a maximum line cycle time T_0 are given. The operations assigned to the same station are executed sequentially; therefore, the station time is equal to the sum of operation times. The obtained station time cannot exceed the given value of the line cycle time T_0 (cycle time constraint). It is necessary to assign the operations to work stations minimizing the number of work stations while respecting precedence and cycle time constraints.

The main hypotheses of SALBP-1 are enumerated in Baybars (1986). Commonly, an optimal solution is obtained by a branch and bound procedure (Johnson 1988, Scholl 1999, Sprecher 1999), Lagrangian relaxation (Aghezzaf and Artiba 1995), shortest path approach or dynamic programming (Gutjahr and Nemhauser 1964, Askin and Zhou 1998). Approximate solutions are obtained using heuristics or meta-heuristics (Helgeson and Birnie 1961, Arcus 1966, Talbot *et al.* 1986). Several comprehensive reviews have been presented in Baybars (1986), Ghosh and Gagnon (1989), Erel and Sarin (1998) and Scholl and Becker (2006).

Surveys on generalizations of SALBP-1 are proposed in Rekiek *et al.* (2002) and Becker and Scholl (2006).

For SALBP-1, the operation times and costs do not depend on the work stations and equipment used. Practically, different stations can be equipped with differing equipment and the operation times and costs depend on the equipment used. Bukchin and Tzur (2000) studied a line balancing problem when several equipment alternatives are available for each station. Each available piece of equipment is

characterized by the cost and processing time. The operations assigned to the same work station are executed sequentially. Only one piece of equipment can be assigned to each station. The design problem addresses the questions of selecting the equipment and assigning operations to work stations. Exact (branch and bound procedure) and approximate algorithms were proposed to minimize the total line cost for this generalization of SALBP.

Some other line balancing and equipment selection approaches for assembly lines are suggested in Graves and Lamar (1983), Graves and Holmes (1988) and Bard and Feo (1991).

Alternative configurations have been considered in Pinto *et al.* (1983) and Sawik (2002), which relate mainly to the equipment selection and where the alternatives are determined by task requirements of either machines or manpower.

In our previous work, some line balancing problems were studied for machining lines with multi-spindle heads. They were called transfer line balancing problems (TLBP). These problems are similar to the problem considered in this paper. For example, in Dolgui *et al.* (2006a) and Belmokhtar *et al.* (2006) a TLBP where blocks of the same station are executed simultaneously and a set of all available spindle heads is known in advance was investigated. This problem is close also to that considered in Bukchin and Tzur (2000), but has several additional properties and constraints. In particular, several pieces of equipment (spindle heads) can be assigned to the same work station as well as all the operations of a work station are executed simultaneously (in parallel). Moreover, inclusion and exclusion constraints for operations and spindle heads must also be satisfied. A graph approach (Dolgui *et al.* 2006a) and a mixed integer program – MIP (Belmokhtar *et al.* 2006) were developed.

TLBPs for the case where the blocks of operations are not known beforehand, and spindle heads of the same work station are activated sequentially (operations are grouped into blocks, but blocks of the same work station are executed in series) were also studied in our work. Inclusion and exclusion constraints are introduced for blocks and work stations. The criterion used was the minimization of a weighted sum of the number of work stations and blocks. Several optimization approaches have been developed: graph approach (Dolgui *et al.* 2008), mixed integer programming (Dolgui *et al.* 2006b), and heuristics based on COMSOAL (Dolgui *et al.* 2005). The main difference between TLBP and SALBP is that all operations of the same block are executed in parallel and so, the block time is not equal to the sum of the operations times.

However, in contrast to Dolgui *et al.* (2008), in the current paper:

- Rotary transfer machines are considered, with all peculiarities of such systems, instead of the linear transfer machines as in our previous publications. For example, all working positions have a common vertical spindle head, and all machine spindle heads are activated in parallel.
- Operation times are calculated taking into account the fact that the tools of the same spindle head have common parameters (the working stroke length and the feed per minute). Therefore, a design decision must consider a choice of these cutting mode parameters for each spindle head.
- Finally, exclusion constraints are enriched by considering the admissible cutting modes.

In comparison with Dolgui *et al.* (2006a) and Belmokhtar *et al.* (2006), here we do not know in advance a set of available spindle heads (blocks).

This paper is an extension of our previous work. Here we consider a special case of transfer machines (machine with a rotary table). This focuses on the optimal assignment of operations to spindle heads and working positions for such a transfer machine. The article is organized as follows. Section 2 presents the problem statement, while section 3 explains the suggested optimization method. An industrial example and experimental results are reported in section 4. Concluding remarks are given in section 5.

2. Problem statement

2.1 Precedence and compatibility constraints

There are several known technological factors (fixed sequences of operations for machining elements, the presence of roughing, semi-finishing and finishing operations, etc.) which determine, as for assembly line balancing, an order relation over the set \mathbf{N} . These precedence constraints can be specified by an acyclic digraph $G^{OR} = (\mathbf{N}, D^{OR})$ where an arc $(i, j) \in D^{OR}$ if and only if the operation i has to be executed before the operation j . Thus, for each operation $j \in \mathbf{N}$, a set $\text{Pred}(j)$ of its immediate predecessors is determined.

The required precision (tolerance) of mutual disposition of machined elements as well as a number of additional factors implies the necessity to perform some operations from \mathbf{N} at the same working position or even by the same spindle head. Such inclusion constraints, specific for the machining environment, can be given by undirected graphs:

- $G^{SP} = (\mathbf{N}, E^{SP})$ where the edge $(i, j) \in E^{SP}$ if and only if the operations i and j must be executed in the same position.
- $G^{SB} = (\mathbf{N}, E^{SB})$ where the edge $(i, j) \in E^{SB}$ if and only if the operations i and j must be executed in the same block.

The possibility or impossibility to perform some pairs of operations at the same working position (or by the same spindle head) depends on a number of machining process constraints: mutual influence of these operations, a forbidden tool location, etc. When it is impossible to perform some pairs of operations at the same working position, we have what is known as exclusion constraints. These exclusion constraints can be defined by undirected graphs:

- $G^{DP} = (\mathbf{N}, E^{DP})$ where the edge $(i, j) \in E^{DP}$ if and only if the operations i and j cannot be executed in the same position.
- $G^{DB} = (\mathbf{N}, E^{DB})$ where the edge $(i, j) \in E^{DB}$ if and only if the operations i and j cannot be executed in the same block.

Such exclusion constraints are sometimes used for the assembly line balancing problems but only for work stations.

Please note that if operations $i, j \in \mathbf{N}^r$ cannot be executed in the same block, then they also cannot be executed in the same position (remember that the operations of different lateral sides cannot be executed in the same block and there is at maximum

one horizontal spindle head per position). Later, we assume that such pairs (i, j) are in E^{DP} only. Similarly, we assume that inclusion constraints for such pairs (i, j) are represented in the graph G^{SB} only, because, operations $i, j \in \mathbf{N}^r$ (i.e. operations on the same side) can be executed at the same position only in the same block.

2.2 Constraint on productivity

The required production rate is provided, if the actual cycle time T_c does not exceed its maximum possible value T_0 . Of course, the actual cycle time value T_c depends on the assignment of operations to blocks and positions.

At the considered design stage, for each operation i , the length of the working stroke $\lambda(i)$, a range $[s_1(i), s_2(i)]$ of possible feeds per revolution, and a range $[\delta_1(i), \delta_2(i)]$ of admissible spindle speeds (cutting speeds) are known. The grouping of operations into a block N leads to choosing common values for the working stroke length $L(N)$ and feed per minute $\gamma(N)$ for all operations of the corresponding spindle head. Note that this rule may not be valid for some specific operations that have always their own working stroke and feed per minute. For example, tapping a hole is often such a specific operation. Later in this article we will not consider the parameters of these operations for the calculation of spindle head (block) times, because the execution time of such an operation cannot exceed T_0 whatever block parameters chosen. Nevertheless, these specific operations are included in the model since they do influence the assignment of all the operations to positions and blocks.

The cycle time T_c is defined by the maximum value among the machining times for working positions. The machining time for a working position (the position time) is defined by the maximum value among the execution times for all blocks (spindle heads) of this position. As aforementioned, the execution time of a block N , i.e. block (spindle head) time $\tau^b(N)$, is defined by the length of the working stroke $L(N)$ and the feed per minute $\gamma(N)$ of the corresponding spindle head. The parameters $L(N)$ and $\gamma(N)$ are calculated based on the possible values of these parameters for all operations from N and maximum admissible value T_0 for the cycle time. Note that for an operation i , $\lambda(i)$ is given, and the minimum $\gamma_1(i)$ and maximum $\gamma_2(i)$ values of the feed per minute are calculated as follows:

$$\gamma_r(i) = s_r(i) \delta_r(i), \quad r = 1, 2$$

The auxiliary time τ^a (rotary table moving, spindle head positioning, etc. between two cycles), which is constant, is neglected here since it is considered when defining T_0 .

Then $\Gamma(N) = [\underline{\Gamma}(N), \bar{\Gamma}(N)]$ is the set of feasible values for the feed per minute $\gamma(N)$ of the spindle head N , where:

$$\begin{aligned} \underline{\Gamma}(N) &= \max\{\max\{\gamma_1(i) | i \in N\}, L(N)/T_0\} \\ \bar{\Gamma}(N) &= \min\{\gamma_2(i) | i \in N\} \end{aligned}$$

If the set $\Gamma(N)$ is empty, then operations of the set N cannot be executed in a single block. Otherwise, a preliminary value of $\gamma(N) \in \Gamma(N)$ might be chosen as follows:

$$\gamma(N) = \min [\max\{\underline{\Gamma}(N), \min\{(L(N)/\lambda(i))^{\mu(i)} \gamma_0(i) | i \in N\}, \bar{\Gamma}(N)]$$

where

- $\mu(i)$ is a constant that characterizes the tool life rate.
- $\gamma_0(i) = s_0(i) \delta_0(i)$.
- $s_0(i), \delta_0(i)$ are the ‘recommended’ values of $s(i)$ and $\delta(i)$.

Note that the recommended value of $s(i)$ and $\delta(i)$ are optimal if operation i is executed separately, i.e. when the corresponding block is only composed of operation i . For calculation of $s_0(i)$ and $\delta_0(i)$ several factors are considered, including tool life.

The feed per revolution for operation i is equal to $s(i, \gamma) = \min[s_2(i), \gamma(N)/\delta_1(i)]$, and the spindle speed is equal to $\delta(i, \gamma) = \gamma(N)/s(i, \gamma)$.

For a fixed feed per minute $\gamma(N)$, the block time $\tau^b(N)$ is equal to $\tau^b(N) = L(N)/\gamma(N)$, where $L(N) = \max\{\lambda(i) | i \in N\}$. If $N_k = (N_{k1}, N_{k2})$ is the set of blocks executed at k th working position, then the working position time is calculated as follows: $\tau^p(N_k) = \max\{\tau^b(N_{kl}) | l = 1, 2\}$, and the machine cycle time is equal to $T_c = \max\{\tau^p(N_k) | k = 1, 2, \dots, m\}$.

Finally, the cycle time (production rate) constraint is as follows: $T_c \leq T_0$.

Notes:

- (1) Considering the sole vertical spindle head of the machine which executes all operations of the set \mathbf{N}^1 , its parameters $L(\mathbf{N}^1)$, $\gamma(\mathbf{N}^1)$ and $\tau^b(\mathbf{N}^1)$ can be determined in advance before optimization. Later we assume that $\tau^b(\mathbf{N}^1) \leq T_0$.
- (2) As aforementioned, in contrast to SALBP, the operation times for TLBP are not known beforehand generally. They are calculated during the grouping of operations into blocks. In this paper, the common cutting modes (specifically the working stroke and feed per minute) are considered for the block time calculation. Obviously, the required production rate cannot be reached if $\lambda(i)/\gamma_2(i) > T_0$ for an operation $i \in N$. This condition is verified in a pre-processing procedure before optimization, therefore, from this point on it is assumed that $\lambda(i)/\gamma_2(i) \leq T_0$ for all operations $i \in N$.

2.3 Mathematical model

Let us introduce the following notation:

- X_{ik} Decision variable which is equal to 1 if the operation i is assigned to the k th position.
- Y_{kr} Auxiliary variable which is equal to 1 if at least one operation from \mathbf{N}^r is executed at the k th working position, $r = 1, 2, \dots, q$, where q is the number of the part sides machined.
- Z_k Auxiliary variable which is equal to 1 if at least one operation from \mathbf{N} is executed at the k th working position.
- C_1, C_2 Two weight coefficients (relative costs for one working position and block).
- m_0 Maximal authorized number of working positions.
- t_{ij} Minimal time necessary for execution of operations i and j in the same block, $t_{ij} = \max(\lambda(i), \lambda(j))/\min(\gamma_2(i), \gamma_2(j))$.

The mathematical model of the considered design problem can be formulated as follows.

The goal is to minimize the estimated equipment cost:

Minimize

$$C_1 \sum_{k=1}^{m_0} Z_k + C_2 \sum_{k=1}^{m_0} \sum_{r=1}^q Y_{kr} \quad (1)$$

subject to the following constraints.

Each operation is assigned to only one working position:

$$\sum_{k=1}^{m_0} X_{jk} = 1; \quad j \in \mathbf{N} \quad (2)$$

All operations are assigned after their predecessors:

$$\sum_{k=1}^{m_0} kX_{ik} \leq \sum_{k=2}^{m_0} (k-1)X_{jk}; \quad (i, j) \in D^{OR} \quad (3)$$

For operations i and j that have to be performed in the same working position:

$$X_{ik} - X_{jk} = 0; \quad (i, j) \in E^{SB} \cup E^{SP}; \quad k = 1, 2, \dots, m_0 \quad (4)$$

For operations i and j that have to be executed in different working positions:

$$X_{ik} + X_{jk} \leq 1; \quad (i, j) \in E^{DP}; \quad k = 1, 2, \dots, m_0 \quad (5)$$

The time of each block cannot exceed the given cycle time T_0 :

$$t_{ij}(X_{ik} + X_{jk} - 1) \leq T_0; \quad k = 1, 2, \dots, m_0; \quad i < j; \quad i, j \in \mathbf{N}^r; \\ r = 2, \dots, q; \quad (i, j) \notin (E^{DP} \cup E^{DB}) \quad (6)$$

The existence condition for the blocks (a block exists if there is at least one operation of a side r assigned to the considered working position k):

$$Y_{kr} \geq X_{jk}; \quad k = 1, 2, \dots, m_0; \quad j \in \mathbf{N}^r; \quad r = 1, \dots, q \quad (7)$$

The constraints on the number of blocks in the working positions (recall that $r=1$ means the horizontal side of the part, i.e. which is machined by a vertical spindle head):

$$\sum_{r=1}^q Y_{kr} \leq 2; \quad \sum_{r=2}^q Y_{kr} \leq 1; \quad k = 1, \dots, m_0 \quad (8)$$

The constraints which define the number of working positions:

$$Z_k \geq Y_{kr}; \quad k = 1, \dots, m_0; \quad r = 1, \dots, q \quad (9)$$

3. Optimization method

3.1 Reduction to a constrained shortest path problem

The decision variables X_{ik} of the model (1)–(9) define the set N_k , i.e. the set of operations that have to be executed at the position k . For the studied transfer

machine, a set N_k is partitioned into blocks (N_{kl} , $l=1,2$) in only one possible way using a straightforward procedure, so this will not be considered in the algorithm.

As for the cases of TLBP with a given set of spindle heads (Dolgui *et al.* 2006a) and where the spindle heads are activated sequentially (Dolgui *et al.* 2008), the problem (1)–(9) can be reduced to searching for a shortest path in a special directed multi-graph G where each feasible solution $P = \langle N_1, N_2, \dots, N_m \rangle$ of the problem (1)–(9) can be viewed as a path in G .

Let

$$v_k = \bigcup_{s=1}^k N_s$$

be the state of the part after machining at k th working position and V be the set of all states of the part for all $P \in \mathbf{P}$ (set of all possible design decisions), also including the states $v_0 = \emptyset$ and $v_{\mathbf{N}} = \mathbf{N}$.

Let us introduce the function $O(N)$ which partitions the set N of the operations executed at the same position into blocks. As aforementioned, this partition is unique. The function $O(N)$ returns the number of blocks for the corresponding working position. If this partition is not possible because of constraints (some exclusion and/or inclusion constraints for blocks and positions can not be respected, and/or there is no common cutting modes for operations from the set N , etc.), then the function $O(N)$ returns a known large number to indicate the infeasibility of the solution.

The above mentioned digraph $G = (V, D)$ is constructed in such a way that an arc $(v, w) \in D$ if and only if $v \subset w$ and $O(w \setminus v) \leq 2$ (remember: for the considered machines the number of spindle heads does not exceed two per position). The arc (v, w) represents the set $w \setminus v$ of operations that are performed at one working position. The cost $C(v, w) = C_1 + C_2 \times O(w \setminus v)$ is assigned to each arc (v, w) .

Each design decision $P \in \mathbf{P}$ can be associated with a path $z(P) = (v_0 = u_0, \dots, u_{j-1}, u_j, \dots, u_{m(x)} = v_{\mathbf{N}})$ in digraph G from the vertex v_0 to the vertex $v_{\mathbf{N}}$.

Let \mathbf{Z} be the set of all paths in G from v_0 to $v_{\mathbf{N}}$. Then a path $z \in \mathbf{Z}$ defines a solution $P(z) = (u_1 \setminus u_0, \dots, u_j \setminus u_{j-1}, \dots, u_{m(x)} \setminus u_{m(x)-1})$ which satisfies inclusion and exclusion constraints but perhaps does not respect the constraint on the maximum number of working positions $m(x) \leq m_0$.

Thus, the initial problem (1)–(9) can be transformed into the following constrained shortest path problem:

$$\text{Min } Q(z) = \sum_{k=1}^{m(z)} C(u_{k-1}, u_k) \quad (10)$$

$$z \in \mathbf{Z} \quad (11)$$

$$m(z) \leq m_0 \quad (12)$$

3.2 Digraph generation and searching for a shortest path

The following algorithm (algorithm 1) generates simultaneously digraph G and solves the problem (10)–(12). Vertices from V can be easily enumerated in a non-decreasing order of their rank in G . In order to do this, we simply partition V into V_i in such a way that $v \in V_i$ if $|v| = i$ for $i = 0, 1, \dots, |\mathbf{N}|$. Obviously, there are no arcs in digraph G between vertices from V_i and V_j if $i \geq j$. The algorithm starts with the initial dummy vertex v_0 . At each iteration of the algorithm, we select a vertex v from the current set V without successors in this set and generate all the possible arcs (v, w) such that $O(w \setminus v) \leq 2$. Selection of v from V can be implemented in different ways. Breadth-first search consists of choosing sequentially v from V_i in the order of their inclusion in V_i and increasing order of i . For a depth-first search, we select v from V_i with maximal i . The depth-first technique may quickly obtain a feasible solution (if it exists). In this case, to find an optimal solution we must restart selection from V_1 and use the breadth-search technique.

In algorithm 1, $CMIN$ is the minimal value of objective function (10). $LM(N)$ and $LC(N)$ are lower bounds on the minimal number of working positions and on the objective function for a set $N \subseteq \mathbf{N}$, respectively. $COST_j(v)$ corresponds to a path with the minimal cost in digraph G from the vertex v_0 to the vertex v which consists exactly of j arcs. Using $PREV_j(v)$, such a path can be easily found in digraph G . Algorithm 1 uses algorithm 2 for the calculations of $LM(v)$ and $LC(v)$.

Algorithm 1:

Step 0. Compute $LM(\mathbf{N})$ and $LC(\mathbf{N})$ by algorithm 2

If $LM(\mathbf{N}) > m_0$ **then stop** (the problem has no feasible solutions)

Step 1. Set $CMIN \leftarrow \infty$, $V_0 \leftarrow \{\emptyset\}$, $V_i \leftarrow \emptyset$, $i = 1, 2, \dots, |\mathbf{N}|$

Step 2. Select a vertex v from V . **If** all the vertices from V have already been selected **then go to Step 7.** Set $m(v) = \min\{k \mid COST_k(v) < \infty\}$ if $v \neq v_0$ and $m(v) = 0$, otherwise

Step 3. **If** $COST_k(v) + LC(\mathbf{N} \setminus v) \geq CMIN$ **for all** $k = 1, 2, \dots, m_0 - LM(\mathbf{N} \setminus v)$ **then go to Step 2**

Step 4. Generate a set $D(v)$ of arcs whose origin is the vertex v

Step 5. **For** each arc $d = (v, w) \in D(v)$:

(a) **if** $w \notin V_{|w|}$ **then** add w to $V_{|w|}$ and compute $LM(\mathbf{N} \setminus w)$ and $LC(\mathbf{N} \setminus w)$ using algorithm 2

(b) **if** $COST_{m(v)}(w) + LC(\mathbf{N} \setminus w) \geq CMIN$ or $m(v) + LM(\mathbf{N} \setminus w) \geq m_0 - 1$ **then** next arc d

(c) **if** $v = v_0$ **then** set $COST_1(w) \leftarrow C(d)$, $PREV_1(w) \leftarrow v_0$

else

for all $k = m(v), m(v) + 1, \dots, m_0 - 1$:

if $COST_k(v) + C(d) < COST_{k+1}(w)$ **then** set

$COST_{k+1}(w) \leftarrow COST_k(v) + C(d)$

$PREV_{k+1}(w) \leftarrow v$

endif

endfor

endif

(a) **if** $w \in V_{|w|}$ **then** set $CMIN \leftarrow \min\{COST_k(w) \mid k = 1, 2, \dots, m_0\}$

Step 6. Go to Step 2

Step 7. If $CMIN = \infty$ then the problem has no feasible solutions

Using values $COST_k(v)$, $k = 1, 2, \dots, m_0$, we can restore all the optimal paths in digraph G .

The set $D(v)$ of arcs whose origin is the vertex v is generated in a similar manner as in Gutjahr and Nemhauser (1964), Dolgui *et al.* (2006a) and Dolgui *et al.* (2008). Several dominance rules can be applied for elements from a set $D(v)$.

Taking into account the objective function (10), the following dominance rule can be proposed.

Rule 1: An arc $d' = (v, w')$ dominates another arc $d = (v, w)$ if $w \subset w'$ and $O(w' \setminus v) = O(w \setminus v)$.

Furthermore, considering cutting modes, this dominance rule 1 can be strengthened as follows.

Rule 2: Arc d' dominates arc d if in addition to the conditions of rule 1, $\tau^p(w' \setminus v) = \tau^p(w \setminus v)$.

Other dominance rules based on the designer's experience can also be applied.

3.3 Calculation of lower bounds

For any $N \subseteq \mathbf{N}$, a lower bound $LC(N)$ can be calculated as follows:

$$LC(N) = C_1 \times LM(N) + C_2 \times LTB(N)$$

where $LTB(N)$ is a lower bound on the total number of blocks needed for the execution of the set of operations N .

$LTB(N)$ can be evaluated as:

$$LTB(N) = \max \left[LM(N), \sum_{r=1}^q LM(N^r) \right]$$

where $N^r = N \cap \mathbf{N}^r$.

If $N \subseteq \mathbf{N}^r$ for some $r \in \{1, 2, \dots, q\}$, then $LM(N)$ can be calculated by finding a lower bound $\chi(H(N))$ on the chromatic number of a mixed graph $H(N) = (N, A, U)$, where:

$A = \{(i, j) \in D^{OR} \mid i, j \in N\}$ is the set of arcs.

$U = \{(i, j) \in E^{DP} \cup D^{OR} \mid i, j \in N\} \cup \{(i, j) \mid i, j \in N; t_{ij} > T_0\}$ is the set of edges.

One can find $\chi(H(N))$ using, for example, methods suggested in Levin (1982), Hansen *et al.* (1997) and Sotskov *et al.* (2001). In general,

$$LM(N) \geq \max \left[\max \{ LM(N^r) \mid r = 1, 2, \dots, q \}, \sum_{r=1}^q LM(N^r) / 2, \sum_{r=2}^q LM(N^r), LM_1(N), LM_2(N), LM_3(N) \right]$$

where $LM_2(N)$ is calculated by algorithm 2, $LM_3(N) = LM_2(N \setminus \mathbf{N}^1)$, and $LM_1(N)$ is a lower bound on the chromatic number of a mixed graph $H_1(N) = (N_1, A_1, U_1)$. In this graph:

- N_1 is the set of connectivity components of the sub-graph of the graph G^{SP} induced by the set N .
- A_1 is the set of arcs (I, J) such that $I, J \in N_1$ and there exist operations $i \in I, j \in J$ for which $i \in \text{Pred}(j)$.
- U_1 is the set of edges (I, J) such that $I, J \in N_1$ and there exist operations $i \in I, j \in J$ for which $(i, j) \in E^{DP} \cup D^{OR}$ or $i, j \in \mathbf{N}^r$ and $t_{ij} > T_0, r \in \{1, \dots, q\}$

$LM_1(N)$ is calculated without the constraint on the number of blocks in a position. Algorithm 2 determines $LM_2(N)$ taking into account all the constraints, and, then $LM_3(N) = LM_2(N \setminus \mathbf{N}^1)$. This algorithm is based on ideas proposed in Levin (1982).

Let us introduce the following additional notation: $lb(j)$ is a lower bound on the position number for the operation j .

$$LP(N, \eta) = \max\{lb_{cur} + \lceil \chi(H(N, lb_{cur}))/\eta \rceil \mid lb_{cur} = lb_{\min}, lb_{\min} + 1, \dots, lb_{\max}\}$$

where

$$lb_{\min} = \min\{lb(j) \mid j \in N\}$$

$$lb_{\max} = \max\{lb(j) \mid j \in N\}$$

$$H(N, lb_{cur}) = (V(N, lb_{cur}), A(N, lb_{cur}), E(N, lb_{cur})) \text{ is a mixed graph}$$

such that

$$V(N, lb_{cur}) = \{j \in N \mid lb(j) = lb_{cur}\}$$

$$A(N, lb_{cur}) = \{(i, j) \in D^{OR} \mid i, j \in V(N, lb_{cur})\}$$

$$E(N, lb_{cur}) = \bigcup_{r=1}^q \{(i, j) \mid i, j \in V(N, lb_{cur}) \cap \mathbf{N}^r; t_{ij} > T_0\} \\ \cup \{(i, j) \in E^{DP} \cup E^{DB} \mid i, j \in V(N, lb_{cur})\}$$

η is an integer, $Succ(j)$ is the set of immediate successors for operation j in digraph G^{OR} .

Algorithm 2:

Step 1. Set $imp \leftarrow 0, ns \leftarrow 0, lb(j) \leftarrow 1$ for all $j \in N$

Step 2. If $(N \cap \mathbf{N}^1 \neq \emptyset)$ and $(N \neq N \cap \mathbf{N}^1)$ then $ns \leftarrow 2$ else $ns \leftarrow 1$

Step 3. For each $j \in N$:

(b) compute $lj = LP(\text{Pred}(j), ns)$

(c) if $lj > lb(j)$ then $lb(j) \leftarrow lj$ and $imp \leftarrow imp + 1$

Step 4. For each $(i, j) \in E^{SP}$ compute $lm = \max[lb(i), lb(j)]$, and set $lb(i), lb(j) \leftarrow lm$

Step 5. If $imp > 0$ then $imp \leftarrow 0$ and go to *Step 3*

Step 6. Set $LM_2(N) \leftarrow LP(\{j \in N \mid Succ(j) \cap N = \emptyset\}, ns) - 1$

In the next section we illustrate how algorithm 2 was applied for an industrial example.

Table 1. Operations and their parameters.

Nbr	Operation i	$\lambda(i)$	$\gamma_1(i)$	$\gamma_2(i)$	$\gamma_0(i)$	$\mu(i)$
1	Drill H1	44.2	22.8	81.3	52.6	0.2
2	Drill H2	44.2	22.8	81.3	52.6	0.2
3	Drill H3	17.6	24.8	86.7	57.4	0.2
4	Bore out face of H3	10.5	29.6	105.7	68.4	0.2
5	Ream H3	14.0	29.8	265.4	139.3	0.4
6	Drill H4	17.6	24.8	86.7	57.4	0.2
7	Bore out face of H4	10.5	29.6	105.7	68.4	0.2
8	Ream H4	14.0	29.8	265.4	139.3	0.4
9	Drill H5	57.5	24.8	86.7	57.4	0.2
10	Bore out face of H5	9.2	29.6	105.7	68.4	0.2
11	Tap H5	29.0	239.2	478.5	49.1	0.6
12	Drill H6	49.7	24.8	86.7	57.4	0.2
13	Bore out face of H6	12.5	29.6	105.7	68.4	0.2
14	Tap H6	15.0	159.5	319	191.4	0.6
15	Drill H7	49.7	24.8	86.7	57.4	0.2
16	Bore out face of H7	12.5	29.6	105.7	68.4	0.2
17	Tap H7	15.0	159.5	319	191.4	0.6
18	Mill P1	171.0	50.2	376.8	167.4	0.35

Table 2. Precedence constraints.

Operation	Direct predecessors	Operation	Direct predecessors	Operation	Direct predecessors
1	18	7	6	13	12
2	18	8	7	14	1 2 5 8 10 13 16
3	18	9	18	15	18
4	3	10	9	16	15
5	4	11	1 2 5 8 10 13 16	17	1 2 5 8 10 13 16
6	18	12	18		

4. An industrial example

Let us consider a machined part as presented in figure 2. This part includes seven holes and a plane to be machined. Five holes (H1–H5) are machined on the horizontal side (P1), and two holes (H6–H7) on one of the lateral sides. For this example $T_0 = 2$ min; $\tau^a = 0.3$ min; $m_0 = 8$; $C_1 = 3$; and $C_2 = 1$. Operations and their parameters are given in table 1.

Precedence as well as inclusion and exclusion constraints are presented in tables 2–5.

In table 6, we give values $lb(j)$ obtained by algorithm 2 for each operation $j \in N$. As a result:

$$LM(N^1) = 5, LM(N^2) = 4, LM_1(N) = 6, \\ LM_2(N) = 6, LM_3(N) = 4, \text{ and } LM(N) = 6.$$

An optimal solution is presented in table 7.

Table 3. Inclusion constraints for blocks.

Operation	Operations required to be in the same block	Operation	Operations required to be in the same block
1	2	5	8
3	6	12	15
4	7	13	16

Table 4. Exclusion constraints for blocks.

Operation	Operations forbidden in the same block	Operation	Operations forbidden in the same block
12	1 2 3 4 6 7 9 10	15	1 2 3 4 6 7 9 10
13	1 2 3 4 6 7 9 10	16	1 2 3 4 6 7 9 10
14	11	17	11

Table 5. Exclusion constraints for positions.

Operation	Operations forbidden in the same position	Operation	Operations forbidden in the same position
5	1 2	10	2 5 8
6	5	12	5 8 11
7	5	13	5 8 11
8	1 2 3 4	15	5 8 11 14
9	2 5 8	16	5 8 11 14
17	14		

Table 6. Calculation of $lb(j)$ for $LM_1(\mathbf{N})$.

Operation j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	$LM_1(\mathbf{N})$
$lb(j)$	2	2	2	3	4	2	3	4	2	3	5	2	3	5	2	3	5	1	6

5. Experimental studies

In tables 8 to 10 we present the characteristics of all the industrial examples studied and the solutions obtained via the proposed algorithm on a Toshiba Satellite A40 Pentium IV 2.66 Ghz 256 Mb RAM. Tables 9 and 10 correspond to versions of the algorithm when graph G is constructed in the ‘forward’ directions. The ‘forward’ versions use the original precedence constraints. The ‘backward’ versions, which deal with the reverse precedence constraints obtained from originals by the inversion of arcs, were also tested. The results of ‘forward’ and ‘backward’ versions are comparable.

In these tables:

- $|\mathbf{N}|$ Number of operations.
- $|D^{SB}|$ Number of edges in graph G^{SB} (the number of inclusion constraints for blocks).

Table 7. An optimal solution.

Operations	$L(N)$	$\gamma(N)$	$\tau^b(N)$
1st position Operations: 18	171	Horizontal side 167.4	1.02
2nd position Operations: 1 2 3 6	57.5	Horizontal side 76.1	0.75
3rd position Operations: 4 7 9	57.5	Horizontal side 76.1	0.75
Operations: 12 15	49.7	Lateral side 57.4	0.86
4th position Operations: 5 8	57.5	Horizontal side 76.1	0.75
5th position Operations: 10	57.5	Horizontal side 76.1	0.75
Operations: 13 16	12.5	Lateral side 68.4	0.18
6th position Operation: 14	15	Lateral side 191.4	0.08
7th position Operation: 11	29	Horizontal side 287.1	0.10
Operation: 17	15	Lateral side 191.4	0.08

Table 8. Characteristics of real life industrial problems.

Example number	$ N $	$ D^{SB} $	$ D^{SP} $	OS	ODB	ODP
1	17	0	0	0.37	0.30	0.10
2	18	6	0	0.44	0.22	0.30
3	18	0	0	0.44	0.22	0.30
4	19	7	2	0.33	0.24	0.22
5	19	0	0	0.33	0.24	0.22
6	20	0	0	0.31	0.28	0.10
7	24	4	1	0.44	0.24	0.32
8	24	0	0	0.44	0.24	0.32
9	33	1	0	0.35	0.21	0.36
10	36	1	0	0.36	0.23	0.36

$|D^{SP}|$ Number of edges in graph G^{SP} (the number of inclusion constraints for positions).

OS Order strength of precedence constraints (the density of the transitive closure of the graph G^{OR}).

ODB Order strength of exclusion constraints for blocks (the density of the graph G^{DB}).

ODP Order strength of exclusion constraints for positions (the density of the graph G^{DP}).

ND Modification of algorithm 1 without the application of the dominance rules and lower bounds.

Table 9. Running times and qualities of solutions obtained with the forward algorithm.

Example number	Time (seconds)										Number of obtained solutions										Number of positions and blocks										
	ND	R1	R2	LB	LB1	LB2	ND	R1	R2	LB	LB1	LB2	ND	R1	R2	LB	LB1	LB2	ND	R1	R2	LB	LB1	LB2	ND	R1	R2	LB	LB1	LB2	
1	2.66	0.05	0.16	0.20	0.03	0.03	50	50	50	50	48	50	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)	(6,10)
2	0.03	0.02	0.02	0.02	0.01	0.02	50	50	50	4	4	4	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)	(7,10)
3	1.75	0.03	0.06	0.05	0.01	0.03	50	50	50	4	4	4	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)
4	0.06	0.02	0.05	0.05	0.03	0.03	50	50	50	50	50	50	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)
5	33.89	0.14	0.95	1.39	0.11	0.38	50	50	50	50	50	50	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)	(6,9)
6	229.80	0.55	1.84	16.94	0.31	0.58	50	50	50	50	50	50	(6,11)	(6,11)	(6,11)	(6,11)	(6,11)	(6,11)	(6,11)	(6,11)	(6,11)	(6,11)	(6,11)	(6,11)	(6,11)	(6,11)	(6,11)	(6,11)	(6,11)	(6,11)	
7	0.63	0.14	0.23	0.63	0.19	0.31	50	50	50	50	50	50	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	
8	146.70	0.28	0.83	112.20	0.31	0.86	50	50	50	50	50	50	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	(12,16)	
9	600	8.17	140.60	600	8.13	109	1	50	50	50	50	50	(14,20)*	(13,19)	(13,19)	(14,20)*	(13,19)	(13,19)	(14,20)*	(13,19)	(13,19)	(14,20)*	(13,19)	(13,19)	(13,19)	(13,19)	(13,19)	(13,19)	(13,19)	(13,19)	
10	600	40.61	600	600	41.78	600	1	50	50	50	50	18	(17,23)*	(16,22)*	(16,22)*	(16,22)*	(16,22)*	(16,22)*	(16,22)*	(16,22)*	(16,22)*	(16,22)*	(16,22)*	(16,22)*	(16,22)*	(16,22)*	(16,22)*	(16,22)*	(16,22)*	(16,22)*	

*Algorithm stops after 600 seconds.

Table 10. Size of graph G for the forward algorithm.

Example number	Number of vertices						Number of arcs					
	ND	R1	R2	LB	LBI	LB2	ND	R1	R2	LB	LBI	LB2
1	1735	43	211	647	35	83	44593	127	1642	3280	68	143
2	80	35	59	34	21	26	353	89	204	41	25	31
3	1736	35	77	210	21	29	34361	89	312	237	25	35
4	151	46	124	106	43	82	948	146	570	308	100	184
5	6919	88	652	2207	71	264	246592	342	4111	13699	209	728
6	13831	97	508	6663	70	1287	859372	451	4957	116948	184	852
7	1112	483	711	1056	420	637	13397	2347	4530	10805	1652	3448
8	15584	275	653	14477	257	618	561929	1409	5216	430393	1012	3796
9	53511	4460	24413	53559	4163	22377	552811	36185	307076	552560	29356	233828
10	57000	13348	51727	57191	12876	52290	430252	132789	537039	429880	117097	527782

- R1 Modification of algorithm 1 with the application of only dominance rule 1.
- R2 Modification of algorithm 1 with the application of only dominance rule 2.
- LB Modification of algorithm 1 which uses lower bounds.
- LB1 Modification of algorithm 1 with dominance rule 1 and lower bounds.
- LB2 Modification of algorithm 1 with dominance rule 2 and lower bounds.

In the studied examples, the number of operations has varied from 17 to 36. The order strength of the precedence constraints is on average 0.38. The density of the exclusion constraints for the blocks is equal to 0.24 on average and for working positions 0.26. Note that these are actual industrial examples, so these densities characterize approximately this type of problem.

In contrast to the exclusion constraints, the densities for the inclusion constraints are very small here.

The available calculation time was set at 600 seconds for each example. The maximal number of optimal solutions was limited to 50 (this is a practical consideration, because it is difficult to choose among more than 50 solutions).

The actual calculation time varied from 0.01 to the upper limit of 600 seconds. Most problems were solved optimally by all versions of algorithm 1. Example 9 with 33 operations could not be solved under the time limit using versions ND and LB (without dominance rules). For some problems (for instance: examples 5 and 6), the use of the lower bounds reduces the calculation time significantly. Example 10 with 36 operations was too hard for versions ND, R2, LB and LB2 (without dominance rule 1). So, only with the dominance rule 1 this problem with 36 operations was solved optimally.

From the obtained results, it seems that the dominance rules are a more powerful technique than lower bounds for reducing the size of graph G and running time of algorithm 1. The lower bounds certainly decrease the size of graph G but sometimes lead to increasing the total running time. This is due to the additional time needed for the lower bound calculation.

Finally, the experimental study showed that for real life industrial problems the calculation time for the proposed approach is acceptable and with the suggested improvements it should be possible to find all optimal solutions in less than 10 minutes.

6. Conclusion

The problem of balancing a transfer machine with a rotary table was studied. This problem consists of partitioning the set of all machining operations into sub-sets minimizing the number of working positions and spindle heads.

The initial problem was reduced to a constrained shortest path search. The proposed approach gives all optimal solutions while respecting production rate, precedence and compatibility constraints.

One advantage of the proposed approach is to easily introduce additional constraints to the initial design problem. It is often necessary to consider them in actual industrial situations. For instance, constraints on the number of operations or constraints on the total power, total feed force, etc. for one spindle head can be examined. These characteristics are calculated by standard user procedures.

Note: the introduction of such procedures in a MIP model results in a significant increase in the model size or the MIP model becomes non-linear.

Another benefit is that all the optimal solutions taken into account the considered criterion are obtained. These optimal solutions can then be evaluated by means of other criteria, if necessary.

The studied problem concerns a transfer machine with a rotary table when all spindle heads of a working position are activated simultaneously. Further research will deal with another class of rotary transfer machines where the spindle heads of each working position are activated sequentially.

Acknowledgments

The research is financially supported by ISTC project B-986 and INTAS Project 03-51-5501. The authors would like to thank Chris Yukna for his help in checking the English.

References

- Aghezzaf, E.-H. and Artiba, A., Lagrangean relaxation technique for the general assembly line balancing problem. *J. Intel. Manuf.*, 1995, **6**, 123–131.
- Arcus, A.L., COMSOAL: a computer method of sequencing operations for assembly lines. *Int. J. Prod. Res.*, 1966, **4**, 259–277.
- Askin, R.G. and Zhou, M., Formation of independent flow-line cells based on operations requirement and machine capability. *IIE Trans.*, 1998, **30**, 319–329.
- Askin, R.G. and Standridge, C.R., *Modeling and Analysis of Manufacturing Systems*, 1993 (John Wiley & Sons, Inc.: New York).
- Bard, F. and Feo, T.A., An algorithm for the manufacturing equipment selection problem. *IIE Trans.*, 1991, **23**, 83–92.
- Baybars, I., A survey of exact algorithms for the simple line balancing problem. *Man. Sci.*, 1986, **32**, 909–932.
- Becker, C. and Scholl, A., A survey on problems and methods in generalized assembly line balancing. *Euro. J. Op. Res.*, 2006, **168**(3), 694–715.
- Belmokhtar, S., Dolgui, A., Guschinsky, N. and Levin, G., Balancing of transfer lines with simultaneously activated spindles. *Computers Indust. Eng.*, 2006, **51**(3), 502–518.
- Bukchin, J. and Tzur, M., Design of flexible assembly line to minimize equipment cost. *IIE Trans.*, 2000, **32**, 585–598.
- Dashchenko, A.I. (ed.), *Manufacturing Technologies for Machines of the Future 21st Century Technologies*, 2003 (Springer: Berlin).
- Dolgui, A., Finel, B., Guschinsky, N.N., Levin, G.M. and Vernadat, F., A heuristic approach for transfer lines balancing. *J. Intel. Manuf.*, 2005, **16**(2), 159–171.
- Dolgui, A., Guschinsky, N.N. and Levin, G.M., A special case of transfer lines balancing by graph approach. *Euro. J. Op. Res.*, 2006a, **168**(3), 732–746.
- Dolgui, A., Finel, B., Guschinsky, N.N., Levin, G.M. and Vernadat, F., MIP approach to balancing transfer lines with blocks of parallel operations. *IIE Trans.*, 2006b, **38**, 869–882.
- Dolgui, A., Guschinsky, N.N., Levin, G.M. and Proth, J.-M., Optimisation of multi-position machines and transfer lines. *Euro. J. Op. Res.*, 2008, **185**(3), 1375–1389.
- Erel, E. and Sarin, S.C., A survey of the assembly line balancing procedures. *Prod. Plan. & Cont.*, 1998, **9**(5), 414–434.
- Ghosh, S. and Gagnon, R.J., A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. *Int. J. Prod. Res.*, 1989, **27**, 637–670.

- Graves, S.C. and Lamar, B.W., An integer programming procedure for assembly design problems. *Op. Res.*, 1983, **31**(3), 522–545.
- Graves, S.C. and Holmes, R.C., Equipment selection and task assignment for multi-product assembly system design. *Int. J. Flex. Manuf. Syst.*, 1988, **1**, 31–50.
- Gutjahr, A.L. and Nemhauser, G.L., An algorithm for the line balancing problem. *Man. Sci.*, 1964, **11**(2), 308–315.
- Hansen, P., Kuplinsky, J. and De Werra, D., Mixed graph coloring. *Math. Meth. Op. Res.*, 1997, **45**, 145–160.
- Helgeson, W.B. and Birnie, D.P., Assembly line balancing using ranked positional weight technique. *J. Indust. Eng.*, 1961, **12**, 394–398.
- Johnson, J.R., Optimally balancing large assembly lines with FABLE. *Man. Sci.*, 1988, **34**, 240–253.
- Levin, G.M., On the evaluation of chromatic characteristics of a mixed graph. *Vesti Akademii Navuk BSSR, Ser. Fiz.-Mat. Navuk*, 1982, **1**, 17–20 (in Russian).
- Nof, S.Y., Wilhelm, W.E. and Warnecke, H.J., *Industrial Assembly*, 1997 (Chapman & Hall: London).
- Pinto, P.A., Dannenbring, D.G. and Khumawala, B.M., Assembly line balancing with processing alternatives: an application. *Man. Sci.*, 1983, **29**, 817–830.
- Rekiek, B., Dolgui, A., Delchambre, A. and Bratcu, A., State of art of assembly line design optimization. *Ann. Rev. Cont.*, 2002, **26**(2), 163–174.
- Sawik, T., Monolithic vs. hierarchical balancing and scheduling of a flexible assembly line. *Euro. J. Op. Res.*, 2002, **109**, 1–23.
- Scholl, A., *Balancing and Sequencing of Assembly Lines*, 1999 (Physica: Heidelberg).
- Scholl, A. and Becker, C., State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *Euro. J. Op. Res.*, 2006, **168**, 666–693.
- Sotskov, Y.N., Dolgui, A. and Werner, F., Mixed graph colouring for unit-time job-shop scheduling. *Int. J. Math. Alg.*, 2001, **2**, 289–323.
- Sprecher, A., A competitive branch-and-bound algorithm for the simple assembly line balancing problem. *Int. J. Prod. Res.*, 1999, **8**, 1787–1816.
- Talbot, F.B., Paterson, J.H. and Gehrlein, W.V., A comparative evaluation of heuristic line balancing techniques. *Man. Sci.*, 1986, **32**, 430–454.