Graph approach for balancing a class of transfer lines

Alexandre Dolgui, N. Guschinsky, G. Levin, J.-M. Proth

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Graph approach for balancing a class of transfer lines

A. Dolgui - N. Guschinsky - G. Levin
J.-M. Proth

N° 4996
Novembre 2003

THÈME 4

Rapport de recherche

Les rapports de recherche de l'INRIA
sont disponibles en format postscript sous
ftp.inria.fr (192.93.2.54)

si vous n'avez pas d'accès ftp
la forme papier peut être commandée par mail :
e-mail : dif.gesdif@inria.fr
(n'oubliez pas de mentionner votre adresse postale).

par courrier :
Centre de Diffusion
INRIA
BP 105 - 78153 Le Chesnay Cedex (FRANCE)

INRIA research reports
are available in postscript format
ftp.inria.fr (192.93.2.54)

if you haven't access by ftp
we recommend ordering them by e-mail :
e-mail : dif.gesdif@inria.fr
(don't forget to mention your postal address).

by mail :
Centre de Diffusion
INRIA
BP 105 - 78153 Le Chesnay Cedex (FRANCE)

Graph approach for balancing a class of transfer lines

A. Dolgui - N. Guschinsky - G. Levin
J.-M. Proth

N° 4996
Novembre 2003

THÈME 4

Graph approach for balancing a class of transfer lines

Dolgui, A.[1], Guschinsky, N.[2], Levin, G.[2] and Proth, J.-M.[3]

[1] *Dept. Scientific Methods for Industrial Management*
Centre SIMMO - Ecole des Mines de Saint Etienne
158, Cours Fauriel, 42023 Saint Etienne cedex 2, France
Ph.: +33 (0)4.77.42.01.66 Fax: +33 (0)4.77.42.66.66
E-mail: dolgui@emse.fr

[2] *Operations Research Lab., United Institute of Informatics Problems*
Surganov St. 6, 220012 Minsk, Belarus
Ph.: +375 (17) 284 21 52 Fax: +375 (17) 231 84 03
E-mail: {gyshin,levin}@newman.bas-net.by

[3] *INRIA / SAGEP, UFR Scientifique, Universite de Metz*
Ile du Saulcy - 57012 Metz Cedex 01, France

Abstract: The optimal design of automatic transfer lines is considered. To this end, operations are partitioned into blocks. The operations of the same block are performed simultaneously, and different blocks of the same workstation are activated sequentially. Constraints related to the design of blocks and workstations, as well as precedence constraints related to operations, are given. The problem consists in minimizing the total number of workstations and blocks, while reaching a given cycle time and satisfying the above constraints. We develop an exact method which proceeds by setting the problem as a constrained shortest path problem. This method is efficient for small and medium size problems.

Keywords: Manufacturing systems, Optimization, Constrained shortest path.

Approche de graphe pour l'optimisation d'un type de lignes de transfert

Dolgui, A., Guschinsky, N., Levin, G. and Proth, J.-M.

Résumé : La conception optimale des lignes de transfert est le sujet de cet article. Pour minimiser le coût de la ligne ainsi que la surface occupée, les opérations sont regroupées en blocs. Les opérations du même bloc sont exécutés simultanément par une seule tête d'usinage. Les têtes d'usinage de la même station de travail sont activées séquentiellement. Les contraintes liées avec la possibilité de conception des têtes d'usinage et des stations correspondantes ainsi que les contraintes de précédence entre les opérations sont connues. Le problème consiste alors à minimiser le nombre total de stations et le nombre total de blocs tout en respectant ses contraintes et en obtenant le temps de cycle prévu. Nous proposons une méthode exacte basée sur la recherche de chemin plus court sous contraintes dans un graphe spécialement conçu. La méthode est efficace pour les problèmes d'une petite taille et d'une taille moyenne.

*Mots clé : **Systèmes de fabrication, Optimisation, Plus court chemin.***

1. INTRODUCTION

We investigate the problem that consists of designing paced transfer lines with workstations in series. Operations that are executed in the same workstation are grouped into blocks. All operations of the same block are performed simultaneously by one spindle head, and different blocks of the same workstation are activated sequentially. A part is fixed in a workstation and all blocks of operations of this workstation are performed. It is only when all the operations of a workstation are performed block after block that the part is transferred to the next workstation. The reposition of part can be realized only during transportation of the part from the current workstation to the next workstation. When all the blocks of the same workstation are terminated, the cycle of machining for this workstation is done. There are no buffers between stations and the line cycle time is constrained by the bottleneck workstation.

Such paced transfer lines are widely used in mechanical industry for mass production (Groover, 1987; Hitomi, 1996). The architecture based on blocks and workstation results in reduction of equipment compared to classical solutions. As a result, the investment cost, as well as surface of the required area, decrease drastically. An example of the transfer line layout is given in Fig.1.

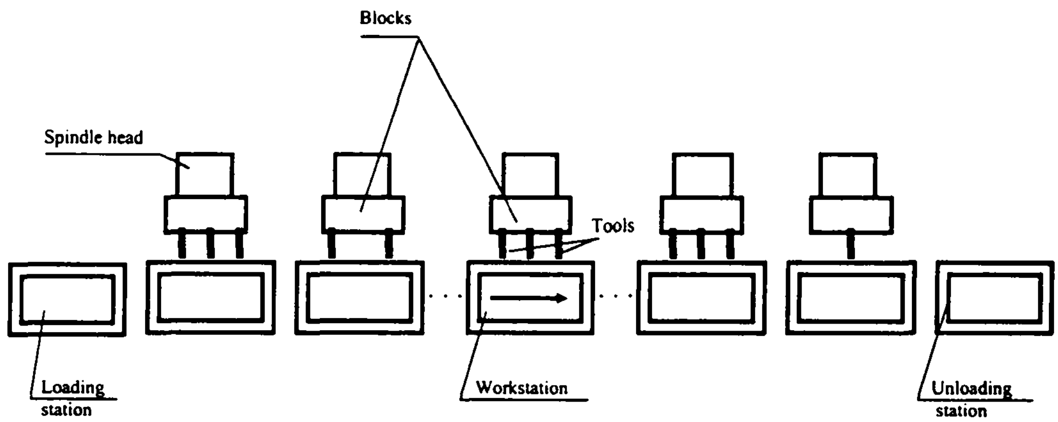

Fig.1. Transfer line example

In this paper, we study one of the key problems concerning the preliminary design of such a transfer line, that is the partition of the set of operations into blocks and workstations.

The goal is to minimize the investment cost (it is estimated by a weighted sum of the number of the workstations and the number of the blocks required) taking into account the following constraints:

a) A given productivity must be reached. In other words, the line cycle time must remain lower than a given value.
b) A partial order relation between operations must be respected.
c) The solution must satisfy constraints related to the grouping of operations in the same block and their implementation in the same workstation.
d) The maximal number of workstations in a line and the maximal number of blocks in a workstation must be not exceeded.

Furthermore, the following data are given:

- the operation time of each operation when it is executed in isolation (separately);
- the relative cost of a station and a block.

The necessity of grouping operations in the same block or in the same workstation often follows from precision constraints: repositioning a part may result in violating the tolerances.

Being compelled to implement operations in different blocks or workstations is the consequence of design or manufacturing rules or constraints as, for instance:

- the nature of the operations;
- the locations of the operations on the part;
- some operations (e.g. milling and drilling at the same part side) require different directions and/or spindle head feeding speeds that forbid their grouping in the same block;
- a small distance between machined elements may prevent an allocation of corresponding tools in the same block;
- power required for feeding or performing a set of operations may prevent to implement this set in one block;
- spindle heads for some operation types cannot be allocated at the same workstation.

We investigate the case when the time required to perform all the operations of a block (also called block time) is equal to the maximal time required by an operation of this block when performed in isolation. We also consider that the time required to execute all the operations of a workstation (also called the workstation time) is the sum of the times of the blocks belonging to the workstation, and that the line cycle time is the maximal workstation time among the workstations of the line.

A problem closed to the problem at hand is known in the literature and referred to as the Simple Assembly Line Balancing Problem (SALB). A detailed analysis of SALB is given in (Scholl, 1999). For SALB, the total line idle time (load unbalance) is minimal if and only if the number of workstations is minimal too. So, for designing an assembly line for a new product, the problem is to assign all the operations to workstations in order to minimize the number for workstations under a given partial order on the set of operations and a fixed cycle time.

In most papers related to SALB, integer linear or dynamic programming models are formulated (Graves and Lamar, 1983; Baybars, 1986; Talbot *et al.*, 1986). The exact methods are mainly based on branch and bound algorithms, as for instance in (Johnson, 1988; Scholl and Klein, 1998). Since the problem is known to be NP-hard, exact solution of large problems will often require a large computation time. The most known heuristic methods are RWP (Helgenson and Birnie, 1961), COMSOAL (Arcus, 1966), MALB (Dar-El, 1973). Very interesting results are obtained by using genetic algorithms (Rekiek *et al.*, 2000) and Tabu search (Pastor *et al.*, 2002). Comprehensive reviews of works on this area are given in (Baybars, 1986; Ghosh and Gadnon, 1989; Erel and Sarin, 1998; Rekiek *et al.*, 2002).

The assembly line balancing methods cannot be directly used to solve the problem studied in this paper for the following reasons:

- the blocks cannot be a priori considered as macro-operations because they are decision variables (must be found during the optimization process);
- the block time is not equal to the sum of the operation times but to the maximum of operation times (for more detail see Section 1.2);
- the precedence constraints and operation compatibility constraints are more complex;
- the line cost depends not only on the number of workstations but also on the number of blocks.

This is a new line balancing problem (transfer line balancing). Modifications of the existing methods for assembly line balancing as well as a development of new methods are needed to solve the investigated problem.

The considered transfer lines are designed for a long exploitation time and they are expensive. The economic benefit, which can be achieved by an optimal line variant, justifies the search of an exact (or "good" approximate) design decision for each concrete line, although it is time-consuming.

An exact method is proposed in this paper. It is based on the modeling of the problem at hand as a constrained shortest path problem. The provisory results on this method are already presented in (Dolgui *et al.*, 1999; Dolgui *et al.*, 2000; Dolgui *et al.*, 2001).

The paper is organized as follows. In Section 1, problem statement and its mathematical model are described. Section 2 is dedicated to the graph approach to the problem: in Section 2.1, the investigated problem is modeled as a constrained shortest path problem in a special digraph; an algorithm for the digraph generation is considered in Section 2.2. Experimental results with the proposed algorithms are reported in Section 3.

2. PROBLEM STATEMENT

The problem is to determine an optimal preliminary design decision, i.e.:

- an optimal number m of workstations;
- an optimal assignment of the set N of all operations, needed for machining the corresponding product, to workstations (N_k, $k=1,...,m$);
- an optimal number n_k of blocks for each workstation (number of spindle heads);
- an optimal partition of N_k into blocks (N_{kl}, $l=1,...,n_k$).

The collection $P=((N_{11},..., N_{1n_1}),..., (N_{m1},..., N_{mn_m}))$ represents a design decision.

We use the following notations:

- T_0 is a given maximum line cycle time;
- t_i is the execution time of operation $i \in$N when it is performed in isolation;
- m_0 and n_0 are respectively the maximal number of workstations in the line and the maximal number of blocks for one workstation.

Below we consider the problem in detail.

2.1. Constraint modeling

The manufacturing and design constraints can be given as follows (Dolgui *et al.*, 2000):

1) The compatibility constraints, i.e. the constraints of necessity and the constraints of impossibility of grouping operations can be represented by collections E^b, E^s, $\underline{E}^b$ and $\underline{E}^s$ of subsets of operations of **N**. The collection E^b (E^s) consists of subsets of operations that must be grouped in one block (at the same workstation). Similarly, the collection $\underline{E}^b$ ($\underline{E}^s$) consists of subsets of operations that cannot be grouped in one block (at the same workstation).

The use of the collections E^b, $\underline{E}^b$, E^s, and $\underline{E}^s$ is in congruence with how the designer describes such constraints. Moreover, most of these constraints can be constructed automatically on the basis of the description of the product, and the equipment possibilities (as, for instance, required machining tolerance, upper bounds of the total power and the total feed force for blocks, design features of blocks and workstations, etc).

2) A non-strict partial order relation over operation set **N** can be represented by digraph G^r=(**N**,D^r). The arc $(i,j)\in$**N**×**N** belongs to set D^r if and only if operation j cannot precede operation i.

The digraph G^r presents a non-strict partial order relation because the specificity of manufacturing constraints of this problem. Cases are possible when an operation j can be performed either simultaneously with an operation i or after i is completed, but cannot precede i. A typical example of such a case is the hole drilling (the operation i) and the chamfering (the operation j). These operations can be performed either separately (i before j) or simultaneously by a combined tool.

If it is necessary to present strict partial order relation between i and j (i strictly before j), then both an arc (i,j) in G^r and the pair $\{i,j\}$ in $\underline{E}^b$ should be used.

2.2. Calculation of the line cycle time and the line cost

In general case, the technique for calculating the time $t^b(N)$ of simultaneous performance of the operation set $N \subseteq \mathbf{N}$ in one block (so-called block time) depends on the specificity of the machining process, line equipment and the method of determining the machining modes (Dolgui *et al.*, 2000). In this paper, we assume that the block time $t^b(N)$ is defined as:

$$t^b(N)=\max\{t_i | i \in N\}, \qquad (1)$$

where t_i is the time of operation i.

This case conforms to a special class of production lines and equipment.

The time $t^s(N_k)$ for the workstation k is equal to

$$t^s(N_k)= \sum_{l=1}^{n_k} t^b(N_{kl}), \qquad (2)$$

where $t^b(N_{kl})$ is the time of the l-th block of the workstation.

Then, the line cycle time $T(P)$ for the design decision P is equal to

$$T(P)=\max\{t^s(N_k) |\ 1 \leq k \leq m\}. \qquad (3)$$

The problem is to assign operations to blocks and workstations so as to minimize the line cost $Q(P)$ subject to the required productivity (i.e. $T(P) \leq T_0$) and the given constraints. We evaluate the line cost as:

$$Q(P)=C_1 m+C_2 \sum_{k=1}^{m} n_k , \qquad (4)$$

where C_1 and C_2 are the given relative costs of one station and one block.

2.3. Constraint compatibility

The constraints defined by the collection E^b are of specific type (they can be treated before optimization). Using the graph G^r and the collection E^b, the set $\mathbf{N}$ can be beforehand partitioned into operation subsets (called macro-operations) such that all the operations of each macro-operation must be performed in the same block for any design decision. This transformation leads to reducing the problem size and to eliminate the constraints E^b. Algorithm 1 does such a transformation (Dolgui *et al.*, 2000).

***Algorithm* 1.**

Step 1. Each group K of intersecting subsets in the collection E^b is replaced by the union of these subsets (call it K). We add also to this subset K all vertices (operations) that belong to a path between any pair of vertices in the subset K in the graph G^r. As a result, we have a new current collection E^b.

Step 2. Each subset N_k, k=1,...,$|E^b|$ from the obtained collection E^b is replaced by one vertex k in the new digraph $\mathbf{G}^r$. We add an arc (k, j) to the new set $\mathbf{D}^r$ if there is an arc (i, l) in the initial graph G^r such that the operation i belongs to the set N_k and the operation l to the set N_j. Multiple arcs are reduced to one arc and loops are deleted.

Step 3. If the digraph $\mathbf{G}^r$ is circuit-free, go to *Step* 4. Otherwise, we consider the set $\mathbf{C}$ of circuits of the digraph $\mathbf{G}^r$, which are not in other circuits (basic circuits).

For each circuit $c \in \mathbf{C}$, the set $V^r(c)$ of vertices of the digraph $\mathbf{G}^r$ that compose the circuit $c \in \mathbf{C}$ is replaced by a new vertex. Multiple arcs are reduced to one arc and loops are deleted.

Step 4. The vertices (operations) of the new digraph $\mathbf{G}^r$ represent macro-operations and arcs of the digraph $\mathbf{G}^r$ determine a new non-strict partial order relation over the set of these macro-operations. The time execution of each macro-operation with a set N of operations is equal to the maximal operation time from N.

Step 5. In subsets of the initial collection E^s each operation from $\mathbf{N}$ is replaced by the macro-operation containing this operation. Procedures similar to steps 1-3 of this algorithm are applied to the resulting collection E^s. In this case we form an intermediate graph $\underline{G}^r$ on the basis of the resulting graph $\mathbf{G}^r$. If there are circuits in $\underline{G}^r$, we add new subsets to E^b and repeat *Steps* 1-4. Otherwise we consider the resulting digraph $\mathbf{G}^r$ as the digraph G^r. As a result, we obtain a new collection E^s.

Step 6. To transform the initial families $\underline{E}^b$ and $\underline{E}^s$ in accordance with the obtained macro-operations, it is sufficient to do the following:

i) Each operation of the initial subsets $\underline{E}^b$ and $\underline{E}^s$ is replaced by the macro-operation containing this operation.

ii) In the resulting collection $\underline{E}^b$ (and $\underline{E}^s$), all subsets containing other subsets of this collection are deleted. To avoid the redundancy, all subsets of $\underline{E}^s$ can be eliminated from $\underline{E}^b$.

Proposition 1. The *constraints defined by graph G^r and families E^b, E^s, $\underline{E}^b$ and $\underline{E}^s$ are compatible* if and only if after this transformation none of the subsets of the new families $\underline{E}^b$ and $\underline{E}^s$ are included in a subset of the new collection E^s.

The following example illustrates Algorithm 1 (steps 1-5).

Example 1. The initial set N consists of 15 operations. Precedence constraints on the set N are given by the digraph G^r (see Fig. 2). The necessity and the impossibility of grouping operations in one block and workstation are determined by the initial families E^b={{2, 13},{3, 12},{5, 6},{7, 8},{7, 14},{8, 15},{9, 10, 11}}, E^s={{4, 10}}, $\underline{E}^b$={1,4}, {2, 7},{2, 14},{3, 14},{5, 15}} and $\underline{E}^s$={{1, 4, 11}}, respectively.

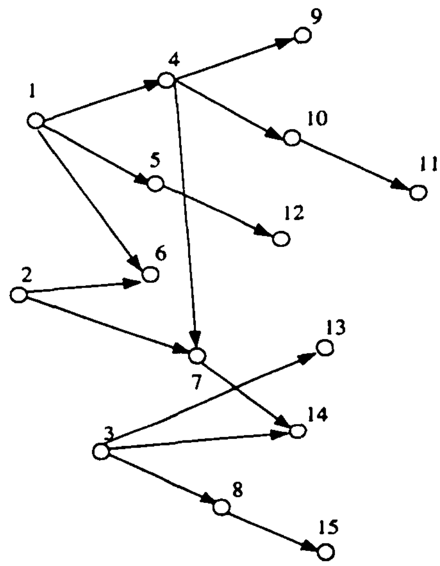

Fig. 2. The initial precedence graph G^r

The digraph G^r, obtained after preliminary transformations (Algorithm 1, steps 1,2), and the new graph G^r obtained after final transformations by this algorithm and grouping operations in macro-operations are given in Fig. 3 and Fig. 4, respectively. The new families E^s, $\underline{E}^b$ and $\underline{E}^s$ are the following: E^s={{3,5}}, $\underline{E}^b$={{1,3},{2,4}} and $\underline{E}^s$={{1,3,5}}.

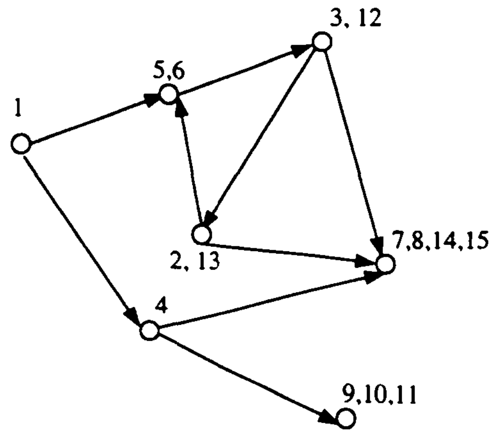

Fig. 3. Graph $\mathbf{G}^r$ after the first step of transformation

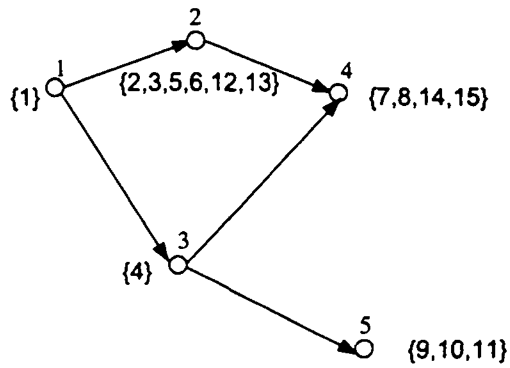

Fig. 4. Graph G^r after the final transformation

In the remaining of the paper we assume that Algorithm 1 has been applied and **N** is the set of macro-operations. The prefix «macro» is omitted for macro-operations.

Further reduction of constraints can be performed on the basis of the following propositions.

Proposition 2. If a set $e \in \underline{E}^b$ ($\underline{E}^s$) includes operations i, j, k, the operation i is the predecessor (not obligatory immediate) of the operation j and the operation j is the predecessor of the operation k, then the operation j can be deleted from the set e.

Proposition 3. If sets $e'=\{i',j'\}$, $e''=\{i'',j''\}$ belong to the family $\underline{E}^b$ ($\underline{E}^s$), the operation i' is the predecessor (not obligatory immediate) of the operation j', the operation i'' is the predecessor of the operation j'', the operation i' is the predecessor of the operation i'' or $i' = i''$ and the operation j'' is the predecessor of the operation j' or $j' = j''$, then the set e'' can be deleted from the family $\underline{E}^b$ ($\underline{E}^s$).

2.4. General mathematical model

Let $M(i)=(k,l)$ if operation $i \in N_{kl}$. We write $(k',l') \prec (k'',l'')$ if *a)* $k' < k''$ or *b)* $k'=k''$ and $l' \le l''$. Then, the line balancing problem can be formulated as follows:

$$Q(P) = C_1 m + C_2 \sum_{k=1}^{m} n_k \rightarrow min, \tag{5}$$

subject to

$$T(P) = \max\{t^s(N_k) \mid 1 \le k \le m\} \le T_0\,; \tag{6}$$

$$\bigcup_{k=1}^{m} \bigcup_{l=1}^{n_k} N_{kl} = \mathbf{N}; \tag{7}$$

$$N_{k'l'} \cap N_{k''l''} = \varnothing,\;\; k',k''=1,\ldots,m,\;\; l'=1,\ldots,n_{k'},\;\; l''=1,\ldots,n_{k''},\;\; k'l' \;\; k''l''; \tag{8}$$

$$M(i) \prec M(j),\quad (i,j) \in D^r; \tag{9}$$

$$N_k \cap e \in \{\varnothing, e\},\; e \in E^s,\;\; k=1,\ldots,m; \tag{10}$$

$$e \not\subset N_{kl},\;\; e \in \underline{E}^b,\;\; k=1,\ldots,m,\;\; l=1,\ldots,n_k; \tag{11}$$

$$e \not\subset N_k,\;\; e \in \underline{E}^s,\;\; k=1,\ldots,m; \tag{12}$$

$$m = m(P) \le m_0; \tag{13}$$

$$n_k = n_k(P) \le n_0,\;\; k=1,\ldots,m. \tag{14}$$

The objective function (5) is the line cost; constraint (6) is the constraint on the cycle time; constraints (7)-(8) reflect the fact that all operations belong the set **N** and that each operation is including in one block only; constraints (9) define the precedence constraints on the set **N**; constraints (10) determine the necessity of grouping the corresponding operations in the same workstation; constraints (11)-(12) take care of the impossibility of grouping operations in one block and the impossibility of execution operations at the same workstation, respectively; (13)-(14) provide the constraints on the number of workstations and blocks for each workstation; m_0 and n_0 determine the maximal number of workstations and blocks for one station, respectively.

Note that if the collection $\underline{E}^b$ contains all pairs of operations from **N** (i.e. each block must consist of one operation only) and constraints (14) are omitted, we obtain the classical assembly line balancing problem. As mentioned above, the solution of the classical assembly line balancing problem is an operation assignment that minimizes the number of workstations, i.e. the total unbalance ($mT_0 - \sum_{k=1}^{m} t(N_k)$).

Example 2. The set N consists of 5 operations, T_0 = 7, operation times t_i are given in Table 1, $E^b = E^s = \underline{E}^s = \varnothing$, precedence graph G^r is shown in Fig. 5, and $\underline{E}^b$={{1,4},{1,5}, {2,3},{2,5}}.

Table 1: Operation times

Operation	1	2	3	4	5
Operation time	4	5	7	7	7

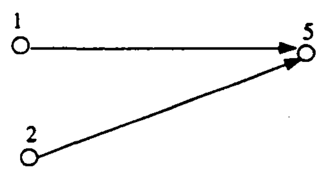

Fig. 5. The precedence graph G^r

An optimal solution to the problem (5-14) is P=(N_1=N_{11}=(1,2), N_2=N_{21}=(3,4,5)) with $Q(P)$=2C_1+2C_2, $T(N_1)$=$T(N_{11})$=5 and $T(N_1)$=$T(N_{21})$=7. The total unbalance for this solution is equal to 7-5+7-7=2. Note that an optimal solution with regard to the total unbalance is P'= (N'_1 =N'_{11}=(1,3), N'_2 =N'_{21} = (2,4), N'_3 =N'_{31}=(5)) $T(N'_1)$ = $T(N'_{11})$ = $T(N'_2)$ = $T(N'_{21})$ = $T(N'_3)$ = $T(N'_{31})$ = 7 and $Q(P')$ = 3C_1+ 3C_2.

3. GRAPH APPROACH

For SALB, the first graph approach was proposed in (Gutjahr and Nemhauser, 1964). The graph approach developed in the paper is based on transformation of the problem (5)-(14) to a problem of finding a constrained shortest path (Garey and Johnson, 1979) in a special digraph. Such a transformation and the algorithm for solving the resulting problem are proposed in Section 2.1. An algorithm of the digraph generation is described in Section 2.2. An example illustrating these algorithms is discussed in Section 2.3.

3.1. A reduction to a constrained shortest path problem

Let **P** be a set of collections $P=((N_{11},\cdots,N_{1n_1}),\cdots,(N_{m1},\cdots, N_{mn_m}))$, satisfying constraints (7)-(12). The set $v_{kl}=\bigcup_{r=1}^{k-1}\bigcup_{q=1}^{n_r} N_{rq} \cup \bigcup_{q=1}^{l} N_{kq}$ can be considered as a state of the part after machining it by the l-th block of the k-th workstation. Let V be the set of all states for all $P\in\mathbf{P}$, including $v_0=\varnothing$. For each vertex $v\in V$ we define a parameter $\Gamma(v)=1$ if for all $e\in E^s$ either $e\cap v=\varnothing$ or $e\subseteq v$, otherwise $\Gamma(v)=0$. The value 1 means that there are no operations in v linked with operations not included in v by a constraint on the necessity of grouping operations in the same workstation.

We construct a digraph $G=(V,D)$, in which the arc $(v',v'')\in D$ if and only if $v'\subset v''$, and $e\not\subseteq N''=v''\backslash v'$ for all $e\in \underline{E}^{bs}=\underline{E}^b\cup\underline{E}^s$, i.e. the arc (v',v'') represents the block $v''\backslash v'$ of operations. For each arc (v',v''), we assign a time parameter $t(v',v'')=t^b(N'')$. An algorithm for construction of the digraph G is described in Section 2.2.

To each design decision $P\in\mathbf{P}$ we can associate a parameterized path $x(P)=((u_0= v_0,\ldots,u_{j-1},u_j,\ldots,u_{l(x)}= v_N),(\gamma_0=1,\ldots,\gamma_j,\ldots,\gamma_{l(x)}=1))$ in the digraph G from the vertex v_0 to the vertex v_N. The parameter γ_j is equal to 1 if the block $u_j\backslash u_{j-1}$ is the last block of the corresponding workstation of P.

Let **X** be the set of all parameterized paths $x=((v_0=u_0,\ldots,u_{j-1},u_j,\ldots,u_{l(x)}=v_N),(\gamma_0=1,\ldots,\gamma_j,\ldots,\gamma_{l(x)}=1))$ in the digraph G from v_0 to v_N where γ_j takes the value 0 if $\Gamma(u_j)=0$ and 0 or 1 otherwise. In the parameterized path $x\in\mathbf{X}$, there is a sequence $j_0=0,j_1,j_2,\ldots,j_{m(x)}=l(x)$ of indices j_r with $\gamma_{j_r}=1$ for $r=0,\ldots,m(x)$. Then, for each parameterized path $x\in\mathbf{X}$, a design decision $P(x)=\{\{u_l\backslash u_0,\ldots,u_{j_1}\backslash u_{j_1-1}\},\{u_{j_1+1}\backslash u_{j_1},\ldots,u_{j_2}\backslash u_{j_2-1}\},\ldots,\{u_{j_{m(x)-1}+1}\backslash u_{j_{m(x)-1}},\ldots,u_{j_m}\backslash u_{j_{m(x)}-1}\}\}$ satisfies the constraints (7)-(11) but may violate the constraint (12).

Thus, the initial problem (5)-(14) can be reduced to the following constrained shortest parameterized path problem:

$$Q(x)=C_1 m(x)+C_2\, l(x) \to \min, \tag{15}$$

$$x\in X, \tag{16}$$

$$\sum_{i=j_{r-1}+1}^{j_r} t^b(u_i \setminus u_{i-1}) \le T_0, \ r=1,\dots,m(x), \tag{17}$$

$$e \not\subset (u_{j_r} \setminus u_{j_{r-1}}), \ e \in \underline{E}^s, \ r=1,\dots,m(x), \tag{18}$$

$$j_r - j_{r-1} \le n_0, \ r=1,\dots,m(x), \tag{19}$$

$$m(x) \le m_0. \tag{20}$$

Algorithm 2 can be used to solve the problem (15)-(20). It is based on the shortest path algorithm (Gallo and Pallottino, 1988). Vertices from V can be easily enumerated in the non-decreasing order of their rank in G. In order to do this, we simply partite V into V_i in such a way that $v \in V_i$ if $|V_i|=i$, $i=0, 1, \dots, |\mathbf{N}|$.

Algorithm 2.

Step 1. **For** each $v \in V$ such that $\Gamma(v)=1$ set $c_k(v) \leftarrow \infty$, $k=1, \dots, m_0$, $c_1(v_0)=0$.

Step 2. **For** $i=0,\dots,|\mathbf{N}|-1$

For each $v \in V_i$ such that $\Gamma(v)=1$ and $\min\{c_k(v) | k=1, \dots, m_0\} < \infty$

a) Assign $g_k(v') \leftarrow \infty$, $T_k(v') \leftarrow \infty$, $k=1, \dots, n_0$, $v' \in V_j$, $j=i+1, \dots, |\mathbf{N}|$.

b) **For** $v'=v$ and $v \in V_j$, $j=i+1, \dots, |\mathbf{N}|-1$, repeat *Steps* 3 and 4.

Step 3. **For** each v'' such that $(v', v'') \in D$, do:

If $v'=v$ **then** $g_1(v'') \leftarrow C_2$, $T_1(v'') \leftarrow t(v', v'')+\tau^s$,

else

for $k=1, \dots, n_0-1$ do:

if $g_k(v') < \infty$, $g_{k+1}(v'') < g_k(v') + C_2$ and $T_k(v')+t(v', v'') \le T_0$, **then**

set $g_{k+1}(v'') = g_k(v') + C_2$, $T_{k+1}(v'') = T_k(v')+t(v', v'')$

endif

Step 4. **If** $\Gamma(v')=1$, v' v, $e \not\subset (v' \setminus v)$ for all $e \in \underline{E}^s$, and $g_0(v')=\min\{g_k(v') | k=1, \dots, n_0\} < \infty$ **then**

If $v=v_0$ **then** $c_1(v') \leftarrow g_0(v')+C_1$,

else

for $k=1, \dots, m_0-1$ do:

if $c_k(v) < \infty$ and $c_{k+1}(v') < c_k(v) + g_0(v')+C_1$, **then**

set $c_{k+1}(v) = c_k(v) + g_0(v')+C_1$

endif

endif

endif

Step 5. Assign

C_{min}=min$\{c_k(v_N) | k=1, ..., m_0\}$.

At Step 3 we analyze all the feasible configurations of one workstation with $k=1, ..., n_0$ blocks after execution of the set v of operations and keep the minimal costs $g_k(v'')$ for each set $v''\backslash v$. At Step 4 we compare all the possible variants of transfer line for the set v' of operations: the current best layout and layout with the last workstation $v'\backslash v$.

It is easy to see that the complexity of the algorithm does not exceed $O(|V|(m_0+ n_0|V| +(m_0+n_0)|D|))$ since the complexity of *Step* 1 is $O(|V|m_0)$ and the complexity of each iteration of *Step* 2 is $O(m_0+ n_0|V| +(m_0+n_0)|D|)$.

3.2. Digraph generation

The following step by step procedure can be used for generating the digraph G. At each step, for an existing vertex v, the set $B(v)$ of the possible output arcs (the operation blocks) is created and the corresponding new vertices are added to G.

Let $E^s=\{e_r^s | r=1,...,qs\}$. For each subset $N \subseteq \mathbf{N}$ we can define a vector $\eta(N) = (\eta^1,...,\eta^{qs})$, where $\eta^i = 1$ if $N \cap e_i^s \quad \varnothing$ and $\eta^i = 0$ otherwise. We let $\mu(N)$=1 if a set N does not involve any $e \in \underline{E}^{bs}$ and $\mu(N)$=0 otherwise. A block $b \in B(v)$ if $\mu(b)$=1 and the set $v \cup b$ contains all the operations that must precede to operations from b.

We can construct the set $B(v)$ by modifying the algorithm for generating states for SALB (Gutjahr and Nemhauser, 1964). The operations from $\mathbf{N}\backslash v$ without any predecessors or whose predecessors belong to v are placed in stage 1. Call this set of operations U. Then we generate all subsets of U and keep a collection S_1 of only those subsets $S \subseteq U$, for which $\mu(S)$=1. At the end of stage 1 the operations of U are "marked" (all operations which are not yet marked are considered as "unmarked").

An immediate follower of a set S is defined as an operation which is an immediate successor of at least one operation in S and is not preceded by any operation that does not belong to $v \cup S$. In general, for any set $S \in \mathrm{S}_k$ in stage k, the unmarked immediate followers of

S are placed in a list called $F(S)$. For each subset $W \subseteq F(S)$, $W \cup S$ is placed in a collection $\mathbf{S}_{k+1}$ for stage k+1 if $\mu(W \cup S)=1$. When all sets of $\mathbf{S}_k$ in stage k have been considered, each operation in $\bigcup_{S \in \mathbf{S}_k} F(S)$ is marked and the process is repeated for stage k+1. When there are no unmarked operations, then the construction of $B(v)$ is completed ($B(v)$ is the union of all $\mathbf{S}_k$).

The process of constructing the set $B(v)$ is illustrated in Fig. 6.

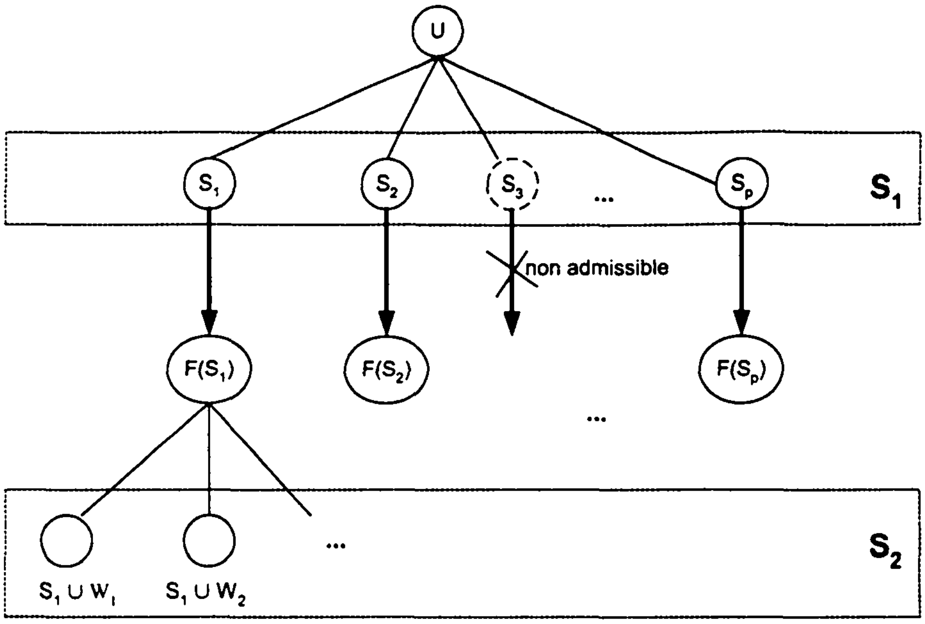

Fig. 6. Branching process

The following dominance relation $\prec_v$ can be introduced on the set $B(v)$. Let's note $b'' \prec_v b'$ for $v \in V$ and blocks $b', b'' \in B(v)$ if the following conditions are satisfied:

i1) $b' \subset b''$;

i2) $t^b(b')=t^b(b'')$;

i3) $\eta(v \cup b')=\eta(v \cup b'')$;

i4) $(b'' \setminus b') \cap e=\varnothing$ for all $e \in \underline{E}^s$.

Proposition 4. Let $P(v,b)$ be a set of collections from **P** such that v is a set of operations preceding a block b. If $b'' \prec_v b'$ for $b', b'' \in B(v)$, then $min\{Q(P)|P \in P(v,b'')\} \leq min\{Q(P)|P \in P(v,b')\}$.

Proof. Assume there exists a collection $P' \in P(v,b')$ such that $Q(P') < \min\{Q(P)|P \in P(v,b'')\}$. Let us construct a collection P'' in the following way. We replace the set b' by b'' and remove from other sets N'_{kl} of the collection P' all the operations which belong to $b''\backslash b'$. Then $t^b(N''_{kl}) \le t^b(N'_{kl})$ for all k, l. Due to assumptions *i1) – i4*), the partial order relation as well as the constraints on the necessity and the impossibility of executing operations at the same workstation are taken into account. As a consequence, $P'' \in P(v,b'')$ and $Q(P'') \le Q(P')$, which contradict the initial assumption.

Therefore, the block b' can be deleted from $B(v)$ if there exists a block $b'' \in B(v)$ such that $b'' \prec_v b'$.

The reduced digraph G is generated by Algorithm 3.

***Algorithm* 3.**

Step 0. Assign $V_0 \leftarrow \{\varnothing\}$, $V_i = \varnothing$, $i=1,2,\ldots,|\mathbf{N}|$.

Step 1. Select vertex v from V (in the order of their inclusion in V_i and the increasing order of i) and perform *Steps* 2-4. If all the vertices from V have already been selected, then stop.

Step 2. Construct $B(v)$ as described above.

Step 3. Check proposition 4 for the set $B(v)$ and exclude dominated blocks b from $B(v)$.

Step 4. **For** each $b \in B(v)$ do:

i) $v' \leftarrow (v \cup b)$;

ii) **if** $v' \notin V_{|v'|}$ **then** include v' into $V_{|v'|}$ and determine $\Gamma(v')$; **endif**

iii) add the arc (v,v') to D.

It should be noted that digraph G constructed by Algorithm 3 does not depend on the line cycle time T_0 as well as m_0 and n_0. So, we can use Algorithm 2 for solving problems (5)-(14) with different values of these parameters without multiple generation of digraph G.

The size of the digraph G depends on the number of constraints (9)-(12). If there are no constraints (9)-(12) at all, Algorithm 3 will construct the set $B(v)$ with $2^{|\mathbf{N}|}$ elements for $v=\varnothing$. At the same time, if the graph G^r is a path consisting of $|\mathbf{N}|$-1 arcs, $E^s = \varnothing$, $\underline{E}^s = \varnothing$ and $\underline{E}^b$ includes only $\{i,j\}$ when $(i,j) \in D^r$, then graph G is also a path of $|\mathbf{N}|$-1 arcs and $|B(v)| = 1$ for each $v \in V$.

For the problems with large number of operations, the digraph G may also be large. In this case, special techniques are required for reducing the digraph G. We propose the following heuristic rules for eliminating elements of $B(v)$ on the basis of relaxed conditions *i2)-i4*). Block b is excluded from $B(v)$, if there exists block $b'\in B(v)$ such that $b\subset b'$ and one of the following conditions holds:

i2') $t^b(b')\delta \geq t^b(b'')$.

i3') Vectors $\eta(v\cup b')$ and $\eta(v\cup b'')$ differ by at most k_1 elements.

i4') The set $(b''\backslash b')$ has non-empty intersections with at most k_2 sets e from $\underline{E}^s$.

Here $\delta \geq 1.0$, $k_1 \geq 0$ and $k_2 \geq 0$ are control parameters.

Elements of $B(v)$ can be also eliminated at random, i.e, block $b\in B(v)$ is deleted from $B(v)$, if there exists block $b'\in B(v)$ such that $b\subset b'$ and current random value within [1,100] is lower than a given value ρ.

In Section 3 we report the results of computational experiments with Algorithm 2 and Algorithm 3.

3.3. Example

In order to illustrate Algorithms 2 and 3, we continue to consider Example 1 from Section 1.3. For this example, after grouping operations in macro-operations by Algorithm 1, we obtained the new digraph G^r (see Fig. 4) and the new families E^s={{3,5}}, $\underline{E}^b$={{1,3},{2,4}} and $\underline{E}^s$={{1,3,5}}. Operations times are given in Table 2. The digraph generation process is shown in Table 3.

Table 2: Operations and their times

Operation	1	2	3	4	5
Time	3	3	4	3	4

Table 3: Digraph generation process

v	Stage for v	Marked operations	Block b	Unmarked immediate followers	Block time	$\eta(v\cup b)$
{∅}	1	1	{1}	2,3	3	(0)
	2	2,3	{1,2}		3	(0)
{1,2}	1	3	{3}	4,5	4	(1)
	2	4,5	{3,4}		4	(1)
			{3,5}		4	(1)
			{3,4,5}		4	(1)

Proposition 4 holds for blocks {1,2} and {1}; {3} and {3,4}; {3,5} and {3,4,5}. The condition *i*4) is not satisfied for blocks {3,4} and {3,4,5}.

The digraph G obtained by Algorithm 3 is shown in Fig. 6. Vertices $v \in V$ with $\Gamma(v)=1$ are drawn by a solid pen.

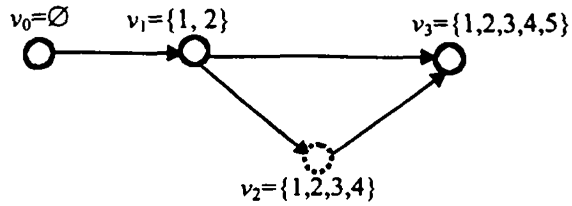

Fig. 7. Graph G

Calculation results of Algorithm 2 are presented in Tables 4 and 5 for $T_0=7$, $m_0=2$ and $n_0=2$.

Table 4: Calculation results for vertex v_0

v	v_0	v_1	v_2	v_3
$g(v)$ $T(v)$	$(0, \infty)$, $(0,\infty)$	$(2,\infty)$, $(3,\infty)$	$(\infty,4)$, $(\infty,7)$	(∞,∞), (∞,∞)
$c(v)$	$(0,\infty)$	$(12,\infty)$		(∞,∞)

Table 5: Calculation results for vertex v_1

v	v_0	v_1	v_2	v_3
$g(v)$ $T(v)$		$(0,\infty)$, $(0,\infty)$	$(2,\infty)$, $(4,\infty)$	$(2,\infty)$, $(4,\infty)$
$c(v)$		$(12,\infty)$		$(\infty,24)$

The optimal solution of the reduced problem (in terms of macro-operations) is N_1=((1,2)), N_2=((3,4,5)) and the optimal solution of the initial problem (in terms of initial operations) is N_1=((1,2,3,4, 5,6,12,13)), N_2=((4,7,8,14,15,9,10,11)).

4. EXPERIMENTAL STUDY OF THE ALGORITHM

The purpose of this study is to examine the impact of various problem parameters on the algorithms performance. We report on four performance measures:

- the total number of generated vertices of graph G;
- the total number of generated arcs of graph G;
- the total number of generated subsets in all $B(v)$;
- the running time of the algorithm.

Test instances were generated in random way for different values of the following problem parameters: |N|, m^s, $\underline{m}^b$, $\underline{m}^s$ p^r, λ^s, $\underline{\lambda}^b$ and $\underline{\lambda}^s$, where:

- m^s,($\underline{m}^b$, $\underline{m}^s$) is the maximal number of operations in subsets from E^s ($\underline{E}^b$, $\underline{E}^s$), respectively;

- λ^s, $\underline{\lambda}^b$ and $\underline{\lambda}^s$ are equal to $\lambda(E)=2|E|/(|N|(|N|-1))$ for $E=E^s$, $\underline{E}^b$ and $\underline{E}^s$, respectively;

- p^r, $\underline{p}^r$ and p^{*r} are equal to $p(D')=2|D'|/|N|/(|N|-1)$ for $D'=D^r$, $\underline{D}^r$ and D^{*r}, respectively, where $\underline{D}^r$ is a set of arcs for immediate predecessors and D^{*r} is a set of arcs in the transitive closure of G^r. The parameters $\underline{p}^r$ and p^{*r} evaluate "order strength" of precedence constraints.

We choose $\lambda^s=0.01$ since we need to coordinate the collection E^s with the graph G^r and families $\underline{E}^b$, $\underline{E}^s$ to provide feasible solutions (all these components are generated independently and randomly). So, we delete subsets from the collection E^s if they violate compatibility with subsets from $\underline{E}^b$ and $\underline{E}^s$.

In Tables 6-10 we give the results for exact variant of Algorithm 3. For each performance measure, the minimal, maximal and average values are presented.

Table 6: Results for |N|=20, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^s=0.05$, $p^r=0.15$, $\underline{p}^r=0.11$, $p^{*r}=0.32$

	$\underline{\lambda}^b=0.02$	$\underline{\lambda}^b=0.05$	$\underline{\lambda}^b=0.10$	$\underline{\lambda}^b=0.20$	$\underline{\lambda}^b=0.40$
Minimal number of V	59	87	89	92	100
Maximal number of V	1920	2440	3108	3248	3568
Average number of V	646	831	1116	1535	1476
Minimal number of D	382	744	622	517	506
Maximal number of D	85488	89890	107407	80306	101788
Average number of D	18728	20604	25163	29338	27664
Minimal total number of sets in $B(v)$	1102	3779	1941	815	861
Maximal total number of sets in $B(v)$	112489	126057	153338	119325	142404
Average total number of sets in $B(v)$	40835	41257	47006	55741	53509
Minimal running time	0.049"	0.159"	0.063"	0.027"	0.031"
Maximal running time	10.058"	10.073"	12.897"	9.108"	12.23"
Average running time	2.9894"	2.4952"	2.8547"	3.3243"	3.4053"

Table 7: Results for |N|=20, $m^s=\underline{m}^b=\underline{m}^s=3$, $\underline{\lambda}^s=0.05$, $p^r=0.15$, $\underline{p}^r= 0.11$, $p^{*r}=0.32$

	$\underline{\lambda}^b=0.02$	$\underline{\lambda}^b=0.05$	$\underline{\lambda}^b=0.10$	$\underline{\lambda}^b=0.20$	$\underline{\lambda}^b=0.40$
Minimal number of *V*	239	247	295	580	558
Maximal number of *V*	1460	2007	2575	2857	2425
Average number of *V*	711	884	1043	1290	1196
Minimal number of *D*	6443	5278	4828	9604	10203
Maximal number of *D*	102127	114935	129188	131925	113221
Average number of *D*	29799	36196	36503	46447	43963
Minimal total number of sets in *B(v)*	14627	9686	10257	18771	21440
Maximal total number of sets in *B(v)*	113501	184373	200907	224758	171638
Average total number of sets in *B(v)*	47611	59006	58856	78178	72708
Minimal running time	0.85"	0.499"	0.409"	0.74"	1.042"
Maximal running time	12.079"	18.384"	18.713"	22.202"	17.822"
Average running time	3.8"	4.7291"	4.345"	6.1602"	5.8185"

From results of Tables 6 and 7, we may conclude that problems with binary relations of the impossibility to group operations in blocks are less time-consuming (up to 2 times) than with ternary relations. Moreover, it seems to be an "optimal" value of $\underline{\lambda}^b$ when test instances can be solved easily.

Table 8: Results for |N|=20, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $p^r=0.15$, $\underline{p}^r= 0.11$, $p^{*r}=0.32$

	$\underline{\lambda}^s=0.02$	$\underline{\lambda}^s=0.05$	$\underline{\lambda}^s=0.10$	$\underline{\lambda}^s=0.20$	$\underline{\lambda}^s=0.40$
Minimal number of *V*	79	139	798	956	956
Maximal number of *V*	235	1240	3375	2808	4424
Average number of *V*	151	579	1655	1712	2349
Minimal number of *D*	599	1416	13283	21260	20320
Maximal number of *D*	1824	38571	136475	100968	172542
Average number of *D*	1211	12074	55322	53762	86964
Minimal total number of sets in *B(v)*	3481	3786	19000	24205	25255
Maximal total number of sets in *B(v)*	16043	58613	169851	138453	252718
Average total number of sets in *B(v)*	6896	25231	70416	60891	105873
Minimal running time	0.095"	0.151"	0.747"	1.267"	1.381"
Maximal running time	0.801"	3.374"	17.594"	12.132"	29.393"
Average running time	0.3112"	1.2278"	5.7314"	4.9898"	11.4314"

It is not surprising (Table 8) that increasing the cardinality of $\underline{E}^s$ results in decreasing the efficiency of domination rules because of violating condition *i*4.

Results of Tables 9 - 10 confirm that most important parameter of test instances is the order strength of precedence constraints.

Table 9: Results for |N|=25, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $\underline{\lambda}^s=0.05$

	$p^r=0.02$, $\underline{p}^r=0.02$, $p^{*r}=0.02$	$p^r=0.05$, $\underline{p}^r=0.04$, $p^{*r}=0.07$	$p^r=0.10$, $\underline{p}^r=0.08$, $p^{*r}=0.17$	$p^r=0.20$, $\underline{p}^r=0.13$, $p^{*r}=0.42$	$p^r=0.40$, $\underline{p}^r=0.14$, $p^{*r}=0.77$
Minimal number of *V*	1858	1563	447	132	18
Maximal number of *V*	24687	13360	2492	493	104
Average number of *V*	10617	4356	1349	310	52
Minimal number of *D*	42514	29991	7630	1108	40
Maximal number of *D*	3253671	1161455	118308	9518	960
Average number of *D*	862265	233036	39556	4616	322
Minimal total number of sets in *B(v)*	111579	73535	19578	2367	111
Maximal total number of sets in *B(v)*	7627700	1898980	172152	29960	1433
Average total number of sets in *B(v)*	2181680	498083	87071.1	10612.6	570
Minimal running time	123.584"	9.384"	0.699"	0.07"	0.008"
Maximal running time	4600.32"	659.84"	16.179"	1.277"	0.043"
Average running time	1176.14"	117.033"	6.6758"	0.3843"	0.0202"

Table 10: Results for $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $\underline{\lambda}^s=0.02$, $p^r=0.25$

	|N|=25, $\underline{p}^r=0.12$, $p^{*r}=0.61$	|N|=50, $\underline{p}^r=0.07$, $p^{*r}=0.77$	|N|=75, $\underline{p}^r=0.04$, $p^{*r}=0.83$	|N|=100, $\underline{p}^r=0.03$, $p^{*r}=0.87$	|N|=125, $\underline{p}^r=0.03$, $p^{*r}=0.89$
Minimal number of *V*	52	332	732	975	863
Maximal number of *V*	239	704	2886	2801	3415
Average number of *V*	135	556	1441	1783	2036
Minimal number of *D*	273	3795	12096	15356	14115
Maximal number of *D*	2751	16193	130892	98662	157569
Average number of *D*	1162	9893	43016	52547	70472
Minimal total number of sets in *B(v)*	1487	17189	47876	50973	75116
Maximal total number of sets in *B(v)*	13798	71161	410776	449356	760786
Average total number of sets in *B(v)*	4753	35652	132100	203307	256020
Minimal running time	0.048"	0.918"	3.499"	5.157"	11.573"
Maximal running time	0.578"	7.324"	57.35"	153.626"	203.078"
Average running time	0.1659"	2.4445"	15.2775"	36.1309"	52.3231"

In Tables 11-19 we give the results when applying heuristic domination rules for different values of control parameters δ for condition *i*2, k_1 for condition *i*3 and k_2 for condition *i*4. The value $k_1 = \propto$ ($k_2 = \propto$) means that the condition *i*3 is not taken into account. Tables 20-22 gather the results of applying the random elimination of elements from *B(v)*.

Table 11: Results for $|N|=20$, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $\underline{\lambda}^s=0.05$, $p^r=0.15$, $\underline{p}^r=0.11$, $p^{*r}=0.32$ with different values of δ

	$\delta = 1.0$	$\delta = 1.1$	$\delta = 1.2$	$\delta = 1.5$	$\delta = 2.0$
Minimal number of *V*	139	111	81	80	80
Maximal number of *V*	1240	1232	1034	991	913
Average number of *V*	673	611	513	410	381
Minimal number of *D*	1416	982	670	662	662
Maximal number of *D*	38571	38371	25988	25530	24407
Average number of *D*	14956	13161	9652	7444	6869
Minimal total number of sets in *B(v)*	3786	2186	1478	1438	1438
Maximal total number of sets in *B(v)*	105767	76483	49702	43897	40765
Average total number of sets in *B(v)*	33036.7	28128.4	21844.4	16223.5	15034.5
Minimal running time	0.057"	0.035"	0.029"	0.024"	0.027"
Maximal running time	2.607"	1.57"	0.918"	0.894"	0.812"
Average running time	0.6488"	0.5153"	0.3635"	0.2608"	0.2379"
Minimal deviation of optimal cost (%)	0	0	0	0	0
Maximal deviation of optimal cost (%)	0	0	0	0	0
Average deviation of optimal cost (%)	0	0	0	0	0
Number of problems solved	10	10	10	10	10

Table 12: Results for $|N|=20$, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $\underline{\lambda}^s=0.10$, $p^r=0.15$, $\underline{p}^r=0.11$, $p^{*r}=0.32$ with different values of δ

	$\delta = 1.0$	$\delta = 1.1$	$\delta = 1.2$	$\delta = 1.5$	$\delta = 2.0$
Minimal number of *V*	798	798	797	794	725
Maximal number of *V*	3375	3099	3099	3099	2979
Average number of *V*	1807	1713	1639	1550	1480
Minimal number of *D*	13283	11911	11727	11586	11551
Maximal number of *D*	139219	130161	125994	125253	123221
Average number of *D*	62541	58365	54856	51979	50096
Minimal total number of sets in *B(v)*	19000	17476	17312	17162	17112
Maximal total number of sets in *B(v)*	195668	177819	162575	160671	157671
Average total number of sets in *B(v)*	81400.6	76236.5	72865	67580.9	64577.3
Minimal running time	0.259"	0.233"	0.235"	0.234"	0.233"
Maximal running time	6.175"	5.917"	5.729"	5.757"	5.65"
Average running time	2.3703"	2.2299"	2.1105"	2.0058"	1.9451"
Minimal deviation of optimal cost (%)	0	0	0	0	0
Maximal deviation of optimal cost (%)	0	0	0	0	0
Average deviation of optimal cost (%)	0	0	0	0	0
Number of problems solved	10	10	10	10	10

Table 13: Results for |N|=20, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $\underline{\lambda}^s=0.20$, $p^r=0.15$, $\underline{p}^r=0.11$, $p^{*r}=0.32$ with different values of δ

	$\delta = 1.0$	$\delta = 1.1$	$\delta = 1.2$	$\delta = 1.5$	$\delta = 2.0$
Minimal number of V	956	956	956	956	860
Maximal number of V	2808	2808	2808	2808	2808
Average number of V	1749	1728	1718	1694	1672
Minimal number of D	21260	19922	19922	19886	19886
Maximal number of D	100968	100968	100968	100968	100968
Average number of D	55629	54226	53761	53263	52932
Minimal total number of sets in $B(v)$	24205	24205	24205	24205	22141
Maximal total number of sets in $B(v)$	138453	121803	120727	118511	118511
Average total number of sets in $B(v)$	62231	60395.3	60225.1	59613.3	59139.9
Minimal running time	0.429"	0.409"	0.41"	0.413"	0.403"
Maximal running time	4.283"	3.655"	3.675"	3.675"	3.689"
Average running time	1.7841"	1.7223"	1.7201"	1.7186"	1.7173"
Minimal deviation of optimal cost (%)	0	0	0	0	0
Maximal deviation of optimal cost (%)	0	0	0	0	0
Average deviation of optimal cost (%)	0	0	0	0	0
Number of problems solved	10	10	10	10	10

Table 14: Results for |N|=20, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $\underline{\lambda}^s=0.05$, $p^r=0.15$, $\underline{p}^r=0.11$, $p^{*r}=0.32$ with different values of k_1

	$k_1 = 0$	$k_1 = 1$	$k_1 = 2$	$k_1 = 3$	$k_1 = \infty$
Minimal number of V	139	139	139	139	139
Maximal number of V	1240	1240	1240	1240	1240
Average number of V	673	579	579	579	579
Minimal number of D	1416	1416	1416	1416	1416
Maximal number of D	38571	38571	38571	38571	38571
Average number of D	14956	12074	12074	12074	12074
Minimal total number of sets in $B(v)$	3786	3786	3786	3786	3786
Maximal total number of sets in $B(v)$	105767	58613	58613	58613	58613
Average total number of sets in $B(v)$	33036.7	25230.8	25230.8	25230.8	25230.8
Minimal running time	0.055"	0.058"	0.055"	0.06"	0.055"
Maximal running time	2.61"	1.258"	1.255"	1.253"	1.267"

Average running time	0.6499"	0.4579"	0.4569"	0.4577"	0.46"
Minimal deviation of optimal cost (%)	0	0	0	0	0
Maximal deviation of optimal cost (%)	0	5.56	5.56	5.56	5.56
Average deviation of optimal cost (%)	0	0.556	0.556	0.556	0.556
Number of problems solved	10	10	10	10	10

Table 15: Results for |N|=20, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $\underline{\lambda}^s=0.10$, $p^r=0.15$, $\underline{p}^r=0.11$, $p^{*r}=0.32$ with different values of k_1

	$k_1=0$	$k_1=1$	$k_1=2$	$k_1=3$	$k_1=\propto$
Minimal number of V	798	798	798	798	798
Maximal number of V	3375	3375	3375	3375	3375
Average number of V	1807	1655	1655	1655	1655
Minimal number of D	13283	13283	13283	13283	13283
Maximal number of D	139219	136475	136475	136475	136475
Average number of D	62541	55322	55322	55322	55322
Minimal total number of sets in $B(v)$	19000	19000	19000	19000	19000
Maximal total number of sets in $B(v)$	195668	169851	169851	169851	169851
Average total number of sets in $B(v)$	81400.6	70415.8	70415.8	70415.8	70415.8
Minimal running time	0.26"	0.262"	0.264"	0.26"	0.262"
Maximal running time	6.175"	6.175"	6.177"	6.23"	6.168"
Average running time	2.3705"	2.0457"	2.0461"	2.0514"	2.0483"
Minimal deviation of optimal cost (%)	0	0	0	0	0
Maximal deviation of optimal cost (%)	0	4.17	4.17	4.17	4.17
Average deviation of optimal cost (%)	0	0.417	0.417	0.417	0.417
Number of problems solved	10	10	10	10	10

Table 16: Results for |N|=20, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $\underline{\lambda}^s=0.20$, $p^r=0.15$, $\underline{p}^r=0.11$, $p^{*r}=0.32$ with different values of k_1

	$k_1=0$	$k_1=1$	$k_1=2$	$k_1=3$	$k_1=\propto$
Minimal number of V	956	956	956	956	956
Maximal number of V	2808	2808	2808	2808	2808
Average number of V	1749	1712	1712	1712	1712
Minimal number of D	21260	21260	21260	21260	21260
Maximal number of D	100968	100968	100968	100968	100968
Average number of D	55629	53762	53762	53762	53762
Minimal total number of sets in $B(v)$	24205	24205	24205	24205	24205
Maximal total number of sets in $B(v)$	138453	138453	138453	138453	138453
Average total number of sets in $B(v)$	62231	60891.5	60891.5	60891.5	60891.5
Minimal running time	0.433"	0.443"	0.444"	0.44"	0.445"
Maximal running time	4.26"	4.369"	4.351"	4.36"	4.325"
Average running time	1.7799"	1.7683"	1.7673"	1.7736"	1.7613"
Minimal deviation of optimal cost (%)	0	0	0	0	0
Maximal deviation of optimal cost (%)	0	0	0	0	0
Average deviation of optimal cost (%)	0	0	0	0	0

Number of problems solved	10	10	10	10	10

Table 17: Results for |N|=20, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $\underline{\lambda}^s=0.05$, $p^r=0.15$, $\underline{p}^r=0.11$, $p^{*r}=0.32$ with different values of k_2

	$k_2 = 0$	$k_2 = 1$	$k_2 = 2$	$k_2 = 3$	$k_2 = \infty$
Minimal number of *V*	139	44	44	44	44
Maximal number of *V*	1240	292	292	292	292
Average number of *V*	673	127	127	127	127
Minimal number of *D*	1416	157	157	157	157
Maximal number of *D*	38571	1857	1857	1857	1857
Average number of *D*	14956	650	650	650	650
Minimal total number of sets in *B(v)*	3786	1111	1111	1111	1111
Maximal total number of sets in *B(v)*	105767	24346	24346	24346	24346
Average total number of sets in *B(v)*	33036.7	5527.4	5527.4	5527.4	5527.4
Minimal running time	0.057"	0.015"	0.013"	0.014"	0.012"
Maximal running time	2.612"	0.49"	0.476"	0.48"	0.466"
Average running time	0.6514"	0.087"	0.0856"	0.0859"	0.0834"
Minimal deviation of optimal cost (%)	0	0	0	0	0
Maximal deviation of optimal cost (%)	0	0	0	0	0
Average deviation of optimal cost (%)	0	0	0	0	0
Number of problems solved	10	10	10	10	10

Table 18: Results for |N|=20, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $\underline{\lambda}^s=0.10$, $p^r=0.15$, $\underline{p}^r=0.11$, $p^{*r}=0.32$ with different values of k_2

	$k_2 = 0$	$k_2 = 1$	$k_2 = 2$	$k_2 = 3$	$k_2 = \infty$
Minimal number of *V*	798	67	67	67	67
Maximal number of *V*	3375	579	579	579	579
Average number of *V*	1807	227	227	227	227
Minimal number of *D*	13283	234	234	234	234
Maximal number of *D*	139219	4262	4262	4262	4262
Average number of *D*	62541	1230	1230	1230	1230
Minimal total number of sets in *B(v)*	19000	1211	1211	1211	1211
Maximal total number of sets in *B(v)*	195668	34872	34872	34872	34872
Average total number of sets in *B(v)*	81400.6	7444.9	7444.9	7444.9	7444.9
Minimal running time	0.262"	0.01"	0.013"	0.012"	0.01"
Maximal running time	6.184"	0.565"	0.565"	0.569"	0.552"
Average running time	2.37"	0.0927"	0.093"	0.094"	0.0899"
Minimal deviation of optimal cost (%)	0	0	0	0	0
Maximal deviation of optimal cost (%)	0	26.32	26.32	26.32	26.32
Average deviation of optimal cost (%)	0	5.264	5.264	5.264	5.264
Number of problems solved	10	10	10	10	10

Table 19: Results for |N|=20, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $\underline{\lambda}^s=0.20$, $p^r=0.15$, $\underline{p}^r=0.11$, $p^{*r}=0.32$ with different values of k_2

	$k_2 = 0$	$k_2 = 1$	$k_2 = 2$	$k_2 = 3$	$k_2 = \infty$
Minimal number of V	956	82	82	82	82
Maximal number of V	2808	629	629	629	629
Average number of V	1749	334	334	334	334
Minimal number of D	21260	326	326	326	326
Maximal number of D	100968	3750	3750	3750	3750
Average number of D	55629	1817	1817	1817	1817
Minimal total number of sets in $B(v)$	24205	1356	1356	1356	1356
Maximal total number of sets in $B(v)$	138453	21383	21383	21383	21383
Average total number of sets in $B(v)$	62231	8641.9	8641.9	8641.9	8641.9
Minimal running time	0.433"	0.009"	0.013"	0.013"	0.014"
Maximal running time	4.255"	0.264"	0.264"	0.265"	0.254"
Average running time	1.7802"	0.0909"	0.091"	0.0906"	0.0873"
Minimal deviation of optimal cost (%)	0	0	0	0	0
Maximal deviation of optimal cost (%)	0	0	0	0	0
Average deviation of optimal cost (%)	0	0	0	0	0
Number of problems solved	10	10	10	10	10

Table 20: Results for |N|=20, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $\underline{\lambda}^s=0.05$, $p^r=0.15$, $\underline{p}^r=0.11$, $p^{*r}=0.32$ with different values of ρ

	ρ = 0	ρ = 5	ρ = 10	ρ = 20	ρ = 40
Minimal number of V	139	906	775	528	193
Maximal number of V	1240	3514	2833	1369	609
Average number of V	673	2007	1547	907	345
Minimal number of D	1416	21735	13148	5058	920
Maximal number of D	38571	139337	80988	19002	4010
Average number of D	14956	66359	33856	10844	1999
Minimal total number of sets in $B(v)$	3786	30819	22919	11643	3231
Maximal total number of sets in $B(v)$	105767	269703	211961	70997	26204
Average total number of sets in $B(v)$	33036.7	112452	73815.4	32191.8	9771.1
Minimal running time	0.055"	0.458"	0.304"	0.14"	0.024"
Maximal running time	2.603"	8.809"	5.379"	1.207"	0.359"
Average running time	0.649"	3.0277"	1.4862"	0.4444"	0.1117"
Minimal deviation of optimal cost (%)	0	0	0	0	0
Maximal deviation of optimal cost (%)	0	0	0	0	0
Average deviation of optimal cost (%)	0	0	0	0	0
Number of problems solved	10	10	10	10	10

Table 21: Results for |N|=20, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $\underline{\lambda}^s=0.10$, $p^r=0.15$, $\underline{p}^r=0.11$, $p^{*r}=0.32$ with different values of ρ

	ρ = 0	ρ = 5	ρ = 10	ρ = 20	ρ = 40
Minimal number of *V*	798	939	823	686	359
Maximal number of *V*	3375	4374	3770	2738	1200
Average number of *V*	1807	2309	1923	1323	552
Minimal number of *D*	13283	15909	10319	6002	1838
Maximal number of *D*	139219	152946	93822	41192	8612
Average number of *D*	62541	69861	40870	16737	3422
Minimal total number of sets in *B(v)*	19000	21645	16725	12651	5765
Maximal total number of sets in *B(v)*	195668	236486	193434	111725	37118
Average total number of sets in *B(v)*	81400.6	107330	77948.8	42890.7	13367.6
Minimal running time	0.262"	0.293"	0.185"	0.115"	0.045"
Maximal running time	6.18"	7.923"	4.963"	1.878"	0.39"
Average running time	2.3706"	2.989"	1.6409"	0.6126"	0.131"
Minimal deviation of optimal cost (%)	0	0	0	0	0
Maximal deviation of optimal cost (%)	0	0	0	0	0
Average deviation of optimal cost (%)	0	0	0	0	0
Number of problems solved	10	10	10	10	10

Table 22: Results for |N|=20, $m^s=\underline{m}^b=\underline{m}^s=2$, $\underline{\lambda}^b=0.05$, $\underline{\lambda}^s=0.20$, $p^r=0.15$, $\underline{p}^r=0.11$, $p^{*r}=0.32$ with different values of ρ

	ρ = 0	ρ = 5	ρ = 10	ρ = 20	ρ = 40
Minimal number of *V*	956	922	795	658	383
Maximal number of *V*	2808	4585	4182	3218	1576
Average number of *V*	1749	2420	2156	1640	841
Minimal number of *D*	21260	16358	10344	6109	2040
Maximal number of *D*	100968	138493	94038	46271	11419
Average number of *D*	55629	67307	43690	20941	5608
Minimal total number of sets in *B(v)*	24205	22238	16646	12927	6071
Maximal total number of sets in *B(v)*	138453	221182	184991	114285	43600
Average total number of sets in *B(v)*	62231	100946	80577.6	52130.2	21046.4
Minimal running time	0.434"	0.315"	0.195"	0.133"	0.049"
Maximal running time	4.283"	7.155"	4.559"	2.09"	0.5"
Average running time	1.7855"	2.8954"	1.7596"	0.8112"	0.2266"
Minimal deviation of optimal cost (%)	0	0	0	0	0
Maximal deviation of optimal cost (%)	0	0	0	0	25
Average deviation of optimal cost (%)	0	0	0	0	2.5
Number of problems solved	10	10	10	10	10

The best results were obtained for the domination rule with parameter $k_2 = 1$.

In Tables 23-24 we provide parameters of the test examples from (Dolgui *et al.*, 2000) and the results of their solution.

Table 23: Parameters of the test examples

Example	$\lvert\mathbf{N}\rvert$	$\lvert D^r\rvert$	$\lvert\underline{E}^b\rvert$	$\lvert\underline{E}^s\rvert$	$\lvert E^s\rvert$	m_0	n_0	T_0	$\underline{m}^b$	$\underline{m}^s$	m^s
1	4	3	2	1	1	2	2	9	2	2	2
2	6	6	2	1	1	3	2	7	2	2	2
3	9	10	4	2	2	4	2	60	2	2	2
4	10	10	3	2	2	4	2	9	2	2	3
5	11	11	2	1	1	2	2	9	2	3	2
6	13	15	4	3	2	4	3	11	3	2	3
7	15	15	5	3	2	4	3	7	2	3	2
8	17	24	4	3	3	4	3	12	3	3	2
9	18	26	3	3	3	5	3	15	2	3	3
10	20	27	4	4	3	5	3	30	3	3	3
11	20	27	3	2	3	5	4	14	4	3	3
12	23	39	4	4	3	4	4	70	3	3	3
13	25	38	5	3	3	6	3	16	3	2	3
14	25	31	5	3	4	5	4	15	3	3	2
15	29	44	3	3	4	6	3	80	3	3	3
16	35	58	4	3	4	5	4	3	3	4	4
17	38	62	4	6	5	7	3	20	2	2	3
18	40	3	4	4	5	7	2	14	2	2	2
19	45	67	5	4	4	7	3	100	3	3	3

Table 24: Results for test examples

Example	Criterion value (number of stations, blocks)					Time (sec)				
	Exact	$\delta = 1.5$	$k_1 = 1$	$k_2 = 1$	$\rho = 20$	Exact	$\delta = 1.5$	$k_1 = 1$	$k_2 = 1$	$\rho = 20$
1	23(2,3)	23(2,3)	23(2,3)	-	23(2,3)	0.015	0	0	0	0
2	33(3,3)	33(3,3)	33(3,3)	33(3,3)	33(3,3)	0	0.001	0	0.001	0
3	35(3,5)	35(3,5)	35(3,5)	35(3,5)	-	0.001	0	0.001	0.002	0
4	23(2,3)	23(2,3)	23(2,3)	23(2,3)	23(2,3)	0.003	0	0	0	0.001
5	23(2,3)	23(2,3)	23(2,3)	23(2,3)	-	0	0	0.003	0	0.003
6	33(3,3)	33(3,3)	33(3,3)	33(3,3)	-	0.001	0.001	0.003	0.003	0
7	22(2,2)	22(2,2)	22(2,2)	22(2,2)	22(2,2)	0.028	0	0.02	0.022	0.033
8	33(3,3)	33(3,3)	-	33(3,3)	33(3,3)	0.004	0.002	0.004	0.002	0.003
9	35(3,5)	35(3,5)	-	35(3,5)	-	0.002	0.005	0.003	0.003	0.004
10	35(3,5)	35(3,5)	-	35(3,5)	35(3,5)	0.033	0.024	0.017	0.024	0.024
11	33(3,3)	33(3,3)	-	33(3,3)	33(3,3)	0.004	0.004	0.002	0.002	0.008
12	35(3,5)	35(3,5)	45(4,5)	35(3,5)	35(3,5)	0.018	0.008	0.012	0.01	0.015
13	25(2,5)	25(2,5)	25(2,5)	25(2,5)	25(2,5)	0.015	0.007	0.019	0.01	0.071
14	23(2,3)	23(2,3)	24(2,4)	23(2,3)	23(2,3)	0.664	0.48	0.4	0.144	0.198
15	34(3,4)	34(3,4)	-	34(3,4)	-	0.161	0.065	0.094	0.105	0.08
16	35(3,5)	35(3,5)	-	-	-	0.63	0.22	0.234	0.15	0.201
17	46(4,6)	46(4,6)	47(4,7)	47(4,7)	47(4,7)	0.585	0.146	0.485	0.15	0.485
18	34(3,4)	34(3,4)	34(3,4)	34(3,4)	34(3,4)	3.109	1.786	2.665	1.95	6.21
19	35(3,5)	35(3,5)	35(3,5)	36(3,6)	35(3,5)	21.32	1.88	18.849	4.095	7.16

For these problems, the most promising rule was to take parameter $\delta = 1.5$.

5. CONCLUSIONS

A graph approach for a problem of optimal designing transfer lines in mechanical industry is proposed. This problem is very complicated problem. In comparison with the well-known problem of assembly lines balancing, the studied problem has a lot of additional constraints.

The proposed graph model permits to describe the manufacturing and design constraints and to verify their compatibility. The graph model with special technique of weights assignment makes possible the use of the constrained shortest path approach. An exact optimization algorithm is developed on the base of the graph model. This approach is more effective when the initial problem has a lot of manufacturing and design constraints which restrict the digraph size.

In industry, the preliminary design stage of transfer line takes several months. Since these lines are very expensive, an exact solution is possible and more important than a heuristic solution, even if the first demands considerable time resource.

ACKNOWLEDGEMENTS

This work is partially supported by INTAS project 00-217.

REFERENCES

Arcus, A. L. (1966). COMSOAL: A computer method of sequencing operations for assembly lines. *Int. J. of Production Research*, **4**, 259-277.

Baybars, I. (1986). A survey of exact algorithms for the simple line balancing problem. *Management Science*, **32**, 909-932.

Dar-El, E.M. (1973). MALB – A heuristic Technique for Balancing Large Single-Model Assembly Lines, *AIIE Transactions*, **5**, 343-356

Erel, E., and S.C. Sarin, (1998). A survey of the assembly line balancing procedures, *Production Planning and Control*, **9**, 414 – 434.

Dolgui, A., N. Guschinsky and G. Levin (1999). On problem of optimal design of transfer lines with parallel and sequential operations. *Proc. of the 7th International Conference on Emerging Technologies and Factory Automation (ETFA'99),* (J.M. Fuertes Ed.), Vol. 1, pp. 329-334, IEEE, Spain.

Dolgui A., Guschinsky and G. Levin (2000). *Approaches to balancing of transfer lines with blocks of parallel operations*; Preprints N°8, Institute of Engineering Cybernetics of Academy of Sciences of Belarus/University of Technology of Troyes, 42 pages

Dolgui, A., N. Guschinsky, G. Levin and Y. Harrath (2001). Optimal design of class transfer lines with blocks of parallel operations. *Manufacturing, Modeling, Management and Control: A Proceedings volume of the Symposium (MIM'2000)*, pp. 36-41, Patras, Greece.

Gallo, G. and S. Pallottino (1988). Shortest path algorithms, *Ann. Oper. Res*, **13**, 3-79.

Garey, R., and D.S. Johnson (1979). *Computer and Intractability: a Guide to the Theory of NP-Completeness*. W.H. Freeman & Company, San Francisco.

Ghosh, S. and Gadnon, R.J. (1989). A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. *International Journal of Production Research*, **27**, 637-670.

Graves, S.C. and Lamar, B.W. (1983). An integer programming procedure for assembly design problems. *Operations Research*, **31**(3), 522-545.

Groover, M. P. (1987). Automation, Production Systems and Computer Integrated Manufacturing, Prentice Hall, Englewood Cliffs, New Jersey.

Gutjahr, A.L. and N.K. Nemhauser (1964). An algorithm for the line balancing problem, *Management Science*, **11** (2), 308-315.

Helgenson, W.B. and Birnie D.P. (1961). Assembly Line Balancing Using Ranked Positional Weight Technique. *Journal of Industrial Engineering*, **12**, pp. 394-398.

Hitomi, K. (1996). Manufacturing System Engineering, Taylor &Francis.

Johnson, J.R. (1988). Optimally balancing large assembly lines with FABLE. *Management Science*, **34**, 240.

Pastor, R, Andrés C, Duran A and M Pérez (2002). Tabu search algorithms for an industrial multi-product and multi-objective assembly line balancing problem, with reduction of the task dispersion. *Journal of Operational Research Society*, **53**, 1317-1324

Rekiek B., De Lit P. and A. Delchambre (2000). Designing Mixed-Product Assembly Lines. *IEEE Transactions on Robotics and Automation*, **16**(3), 268-280.

Rekiek B., Dolgui A., Delchambre A., and A. Bratcu (2002). State of art of assembly lines design optimization. *Annual Reviews in Control*, **26**/2, 163-174.

Scholl, A. (1999). Balancing and Sequencing of Assembly Lines, Physica-Varlag, Heidelberg.

Scholl, A. and R. Klein (1998). Balancing assembly lines effectively: a computational comparison, *European Journal of Operational Research*, **114**, pp. 51-60.

Talbot, F.B., J.H. Paterson and W.V. Gehrlein (1986). A comparative evaluation of heuristic line balancing techniques. *Management Sciences*, **32**, 430-454.

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399

ISRN INRIA/RR--4996--FR+ENG

RR - 4996