15th Triennial World Congress, Barcelona, Spain

OPTIMAL DESIGN OF TRANSFER LINES WITH BLOCKS OF PARALLEL OPERATIONS

Alexandre Dolgui[*], Nikolai Guschinsky[**] and Genrikh Levin[**]

[*] *Industrial Engineering Department*
University of Technology of Troyes
12, rue Marie Curie, B.P. 2060
10010 Troyes Cedex, France
E-mail: dolgui@utt.fr

[**] *Operations Research Lab.*
Institute of Engineering Cybernetics
Surganov St. 6, 220012 Minsk, Belarus
E-mail: {gyshin, levin}@newman.bas-net.by

Abstract: This paper is devoted to a preliminary design problem of paced automatic transfer lines. The aim is to choose a logical layout of the line which minimises its cost subject to the given productivity and technological constraints. The paper focuses on a mathematical model of the problem and a method to solve it. The proposed method is based on the transformation of the initial problem into a constrained shortest path problem. *Copyright©2002 IFAC*

Keywords: Production systems, Preliminary design, Mathematical programming, Graph theory, Optimisation.

1. INTRODUCTION

Paced transfer lines with workstations in series and with the possibility of simultaneous execution of several manufacturing operations in one block (by one spindle head) at the same workstation are widely used in mechanical industry for mass production (Groover, 1987; Hitomi, 1996). The operation concentration in blocks and at workstations allows to decrease considerably the total number of spindle heads and workstations providing a required output. This decreases the total cost of the line and the occupied area.

In this paper, a problem of optimal logical layout (generalized line balancing problem) for the class of transfer lines with blocks of operations is investigated. The blocks of the same workstation are executed sequentially. The sequential execution of the blocks is usually provided by using removable or revolving spindle heads.

The preliminary stage of logical layout line design involves the following basic phases:

- selection of operations sequence for processing each element of a piece and preliminary determination of machining modes for executing each operation individually. It results in a set **N** of operations that have to be performed at transfer line;

- formation of main technological and constructive constraints that must be taken into account in logical layout synthesis;
- logical layout synthesis.

In a real design process each of these phases is a complicated independent problem. This paper deals with the last phase under the assumption that the first two stages have been accomplished.

It is assumed that the block time (i.e. the time of block execution) is determined only by the operations of this block and is not shorter than the time for performing each operation of the block when the operation is executed individually. The workstation time is the sum of execution times of its blocks and the line cycle time is the maximum workstation time.

The considered line balancing problem is to assign operations of the set **N** to workstations and blocks in such a way that:

- a weighted sum of the number of workstations and blocks is as small as possible;
- a given line cycle time is provided;
- given technological constraints are satisfied.

The following technological constraints are considered:

i) a partial order relation over set **N**, which defines the set of feasible operation sequences;

ii) the necessity to perform some groups of operations in one block or at the same workstation (e.g., because of a required machining tolerance);

iii) the impossibility to perform some groups of operations in one block or at the same workstation. In particular, such constraints take into account the following factors: a) technological incompatibility of the operations, b) structural features of spindle heads and workstations, c) inadmissibility of exceeding cumulative values of certain characteristics of the operations in block (e.g., the total power, the total feed force, etc).

The above constraints can also reflect general rules of such type line design as well as the existing tendencies and designer experience. By changing the set of such constraints, the designer can control the process of solving the considered problem.

If every block comprises one operation only and there are no constraints ii) and iii), the considered problem coincides with the classical assembly line balancing problem. A detailed analysis of the assembly line balancing problem is given in (Baybars, 1986; Scholl, 1999; Rekiek *et al.*, 2000). The problem is to assign all operations to workstations in order to minimize a specified criterion under a given partial order of operations and a fixed cycle time. Usually, the criterion is the unbalance of the assembly line. The unbalance is minimal if and only if the number of workstations is minimal too.

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;
- the block time is, as a rule, equal to the maximal time needed to perform each operation of the block individually;
- the line cost depends not only on the number of workstations but also on the number of blocks.

Modifications of the existing methods for assembly line balancing as well as a development of new methods are needed to solve the investigated transfer line balancing problem. The considered transfer lines are designed for a long exploitation time and they are expensive. The economic benefits, which can be achieved by an optimal line variant, justify the search of an exact (or "good" approximate) design decision for each concrete line, even though it is time-consuming.

In this paper, a model and a method for synthesis of line logical layout are proposed.

2. PROBLEM STATEMENT

The investigated problem is to determine the following parameters of the designed line:

a) a number m of workstations;

b) an assignment of operations from **N** to workstations (N_k, k=1,...m);

c) a number n_k of blocks for each workstation;

d) a partition of N_k into blocks (N_{kl}, l=1,...,n_k).

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

2.1. Modeling of the constraints

The technological constraints can be given in the following way:

1) A non-strict partial order relation over 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 the operation j cannot precede the operation i.

2) The constraints of the necessity and the possibility of grouping operations can be represented by hypergraphs H^b=(**N**, E^b), H^s=(**N**, E^s), $\underline{H}^b$=(**N**, $\underline{E}^b$) and $\underline{H}^s$=(**N**, $\underline{E}^s$). The set E^b (or E^s) consists of

subsets of the operations from **N** that must be performed in one block (or at the same workstation). Analogously, the set $\underline{E}^b$ (or $\underline{E}^s$) consists of subsets of the operations from **N** that cannot be grouped in one block (or at the same workstation).

3) The constraints that take into account cumulative values of certain characteristics of the operations in block N are given as

$$\mathrm{W}(N) = \sum_{i\in N} w_{ip} \le W_p,\ \ p=1,\ldots,\underline{p}, \tag{1}$$

where w_{ip} is the *p-th* characteristic for the operation i, and W_p is an upper bound on a cumulative value of the corresponding characteristic for the block, $\underline{p}$ is the total number of the characteristics. These constraints are special case of constraints 2).

***Note*.** The digraph G^r represents a non-strict partial order relation due to the specificity of technological constraints of this problem. An operation j can be performed either simultaneously or after an operation i, but it cannot precede i. A typical example 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.

2.2. Line cycle time calculation

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

In this paper, it is assumed that the block operations have the common working stroke $L(N)=\max\{l_i|i\in N\}$ and the common feed per minute $S(N)=\max\{s_i|i\in N\}$ where l_i is working stroke, s_i is feed per minute for operation i. Then the block time

$$t^b(N)=L(N)/S(N)+\tau^b, \tag{2}$$

where τ^b is a constant.

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

$$t^s(N_k)= \sum_{l=1}^{n_k} t^b(N_{kl}) +\tau^s, \tag{3}$$

where τ^s is a constant which gives an additional auxiliary time, it is the same for all workstations (this constant can be omitted if it is beforehand subtracted from the given value T_0 of the line cycle time).

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

$$T(P)=\max\{t^s(N_k)|\ 1\le k \le m\}. \tag{4}$$

2.3. General mathematical model

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. $T(P)\le T_0$) and the given constraints. The cost $Q(P)$ depends on the number of the workstations and on the number of the blocks for each workstation:

$$Q(P)=C_1m+C_2 \sum_{k=1}^{m} n_k\ , \tag{5}$$

where C_1, C_2 are some constants, which reflect the relation between workstation and spindle head costs.

Using graphs G^r and H^b, the set **N** can be beforehand partitioned into operation subsets (called macro-operations) such that all the operations of a macro-operation must be performed in the same block for any design decision. This transformation leads to reducing the problem size and excluding the constraints of the necessity to group the operations in one block. An algorithm for such transformation is proposed in (Dolgui *et al.*, 2000). In the rest of the paper, it is assumed that the algorithm has been applied and **N** is the set of macro-operations.

Let $M(i)= (k,l)$ if operation $i\in N_{kl}$. The following notation is introduced $(k',l') \prec (k'',l'')$, if a) $k' < k''$ or b) $k'=k''$ and $l'\le l''$. Then, the following mathematical model can be used for the considered problem:

$$Q(P) \to min, \tag{6}$$

$$T(P)=\max\{t^s(N_k)|\ 1\le k \le m\} \le T_0; \tag{7}$$

$$\bigcup_{k=1}^{m} \bigcup_{l=1}^{n_k} N_{kl} = \mathbf{N}; \tag{8}$$

$$N_{k'l'} \cap N_{k''l''} = \varnothing,\ k',k''=1,\ldots,m,$$
$$l'=1,\ldots,n_{k'},\ l''=1,\ldots,n_{k''},\ k'l'\ne k''l''; \tag{9}$$

$$M(i) \prec M(j),\quad (i,j)\in D^r; \tag{10}$$

$$N_k\cap e\in \{\varnothing,e\},\ e\in E^s,\ \ k=1,\ldots m; \tag{11}$$

$$e\not\subset N_{kl},\ \ e\in \underline{E}^b,\ \ k=1,\ldots,m,\ \ l=1,\ldots,n_k; \tag{12}$$

$$\sum_{i\in N_{kl}} w_{ip} \le W_p,$$

$$k=1,\ldots,m,\ l=1,\ldots,n_k,\ p=1,\ldots,\underline{p}; \qquad (13)$$

$$e \not\subset N_k,\ e\in \underline{E}^s,\ k=1,\ldots,m; \qquad (14)$$

$$m=m(P) \le m_0; \qquad (15)$$

$$n_k = n_k(P) \le n_0,\ k=1,\ldots,m. \qquad (16)$$

The objective function (5-6) is the line cost; the constraint (7) provides the given cycle time; the constraints (8-9) reflect the condition of assigning all operations of the set **N** and including each operation only in one block; the constraints (10) define the precedence constraints over the set **N**; the constraints (11) determine the necessity of executing the corresponding operations at the same workstation; the constraints (12-14) take care of the possibility of grouping some operations in one block and the possibility of execution some operations at the same workstation; the constraints (15,16) provide an admissible number of workstations and blocks for each workstation.

Note: if each block can consist of only one operation and if constraints (16) are omitted, then this is the classical assembly line balancing problem.

3. GRAPH APPROACH

The problem (6)-(16) can be formulated as mixed linear programming problem (Dolgui *et al.*, 2000). In this paper, another approach is proposed which is based on a transformation of the initial problem into a problem of finding a constrained shortest path (Garey and Johnson, 1979) in a special digraph.

3.1. 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 (8)-(14). 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 l-th block of the k-th workstation. Let V be the set of all states for all $P\in\mathbf{P}$, including $v_0=\varnothing$ and $v_N=\mathbf{N}$.

A digraph $G=(V,D)$ can be constructed, in which the arc $(v',v'')\in D$ if and only if $v'\subset v''$, and the set $N''=v''\backslash v'$ satisfies the following conditions: $t^b(N'')\le T_0$ and $e\not\subset N''$ for all $e\in \underline{E}^{bs}=\underline{E}^b\cup\underline{E}^s$. The arc (v',v'') represents the block $v''\backslash v'$ of operations. For each arc (v',v''), a time parameter $t(v',v'')=t^b(N'')$ and a set $\Gamma_{v'v''}$ are assigned. The set $\Gamma_{v'v''}$ is equal to $\{0,1\}$ if $e\cap v''=\varnothing$ or $e\subseteq v''$ for all $e\in E^s$. Otherwise, the set $\Gamma_{v'v''}$ is equal to $\{0\}$.

The graph G has the following properties:

- if the operation $i\in v\in V$ and there exists $(j,i)\in D^r$, then the operation $j\in v$;
- the set $\Gamma_{wv}=\Gamma_{uv}$ for all vertices $v,w,u\in V$ such that (w,v) and $(u,v)\in D$.

A parameterised path $x(P)=((v_0=u_0,\ldots,u_{j-1},u_j,\ldots,u_{l(x)}=v_N),(\gamma_1,\ldots,\gamma_j,\ldots,\gamma_{l(x)}))$ in the digraph G from the vertex v_0 to the vertex v_N is associated to each design decision $P\in\mathbf{P}$. 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 in the digraph G from v_0 to v_N. In the parameterized path $x=((v_0=u_0,\ldots,u_{j-1},u_j,\ldots,u_{l(x)}=v_N),(\gamma_1,\ldots,\gamma_j,\ldots,\gamma_{l(x)}))\in\mathbf{X}$ a sequence $j_1,j_2,\ldots,j_{m(x)}=l(x)$ of indices j_r with $\gamma_{j_r}=1$ for $r=1,\ldots,m(x)$ can be selected. Then, for each parameterized path $x\in\mathbf{X}$, a design decision $P(x)=\{\{u_1\backslash u_0,\ldots,u_{j_1}\backslash u_{j_1-1}\},\{u_{j_1+1}\backslash u_{j_1},\ldots,u_{j_2}\backslash u_{j_2-1}\},\ldots,\{u_{j_m}\backslash u_{j_m-1}\}\}$ satisfies the constraints (8)-(13) but may violate the constraint (14).

Thus, the initial problem (6)-(16) can be reduced to the following constrained shortest parameterized path problem:

$$Q(x)=C_1 m(x)+C_2\, l(x) \rightarrow \min, \qquad (17)$$

$$x\in \mathrm{X}, \qquad (18)$$

$$\sum_{i=j_{r-1}+1}^{j_r} t^b(u_i\backslash u_{i-1}) \le T_0,\ r=1,\ldots,m(x), \qquad (19)$$

$$e\not\subset (u_{j_r}\backslash u_{j_{r-1}}),\ e\in\underline{E}^s,\ r=1,\ldots,m(x), \qquad (20)$$

$$j_r - j_{r-1} \le n_0,\ r=1,\ldots,m(x),\ j_0=0, \qquad (21)$$

$$m(x)\le m_0. \qquad (22)$$

Algorithm 1 can be used to solve the problem (17)-(22). It is based on the shortest path algorithm (Gallo and Pallottino, 1988). In Algorithm 1, $\overline{V}$ is a set of vertices $v\in V$ such that $\Gamma_{wv}=\{0,1\}$ for $(w,v)\in D$, $v_0\in\overline{V}$ and all vertices from V are enumerated in the non-decreasing order of their rank in G. Operation MINMIN denotes finding non-dominated pairs (a pair (a_1,b_1) dominates a pair (a_2,b_2), if $(a_1\le a_2)$ and $(b_1\le b_2)$).

Algorithm 1.

Step 1. Assign

$\{(c(v_0),m(v_0))\}\leftarrow\{(0,0)\}$,

$\{(c(v_j),m(v_j))\}\leftarrow\{(\infty,\infty)\}$ for $v_j\in\overline{V}, v_j\ne v_0$.

Step 2. For $i=0,\ldots,|V|-2$ such that $v_i \in \overline{V}$ and $\min\{m(v_i)\} < m_0$

1. Assign

$\{(g(v_i,v_i),T(v_i,v_i))\} \leftarrow \{(0,0)\}$,

$\{(g(v_i,v_j),T(v_i,v_j))\} \leftarrow \{(\infty,\infty)\}$, $j=i+1,\ldots,|V|-1$.

2. **For** $j=i,\ldots,|V|-1$, do

a) **For** each v_k such that $(v_j,v_k)\in D$, the current set of non-dominated pairs $\{g(v_i,v_k),T(v_i,v_k)\}$ is updated

$$\{(g(v_i,v_k),T(v_i,v_k))\} \leftarrow \mathrm{MINMIN}\{(g(v_i,v_k),T(v_i,v_k))\}\cup \{(g(v_i,v_j)+C_2, T(v_i,v_j)+t(v_j,v_k)) \mid g(v_i,v_j)< n_0C_2, T(v_i,v_j)+t(v_j,v_k)\le T_0\};$$

b) **If** $v_j \in \overline{V}, v_j \neq v_i$, and $e \not\subset (v_j \setminus v_i)$, $e\in \underline{E}^s$,

then the current set of non-dominated pairs $(c(v_j),m(v_j))$ is updated

$$\{(c(v_j),m(v_j))\}\leftarrow \mathrm{MINMIN}\{(c(v_j),m(v_j))\}\cup \{(c(v_i)+\min\{g(v_i,v_j)\},m(v_i)+1)\}.$$

Step 3. Assign

$C_{min}=\min\{c(v_N) \mid m(v_N) \le m_0\}$.

The value $\sum_{v_i\in\overline{V}} \sum_{j=i}^{|V|} d(v_j)(A(v_i,v_j)+K(v_j))$ determines the time complexity of the algorithm, where $d(v_j)$ is the outdegree of the vertex j; $A(v_i,v_j)=|\{(g(v_i,v_j),T(v_j))\}|$; $K(v_j)=|\{(c(v_j),m(v_j))\}|$ if $v_j \in \overline{V}$, and $K(v_j)=0$ otherwise. In the considered problem, $A(v_i,v_j)\le n_0$ and $K(v_j)\le m_0$. Therefore, the complexity of the algorithm does not exceed $O(|V|^3(m_0+n_0))$.

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 corresponding new vertices are added to G. The set $B(v)$ can be formed directly during the digraph generation or can be chosen from the set B of all feasible blocks preliminary obtained.

Let $E^s=\{ e_r^s \mid r=1,\ldots,q^s \}$; $a_{ij}=1$, if there is a path from the vertex i to the vertex j in the digraph G^r, and $a_{ij}=0$ otherwise. A block b is feasible if the following conditions hold:

a) $e\not\subset b$ for any $e\in \underline{E}^{bs}$;

b) if $i,j\in b$, $r\in \mathbf{N}$ and $a_{ir}=a_{rj}=1$, then $r\in b$;

c) $t^b(b)\le T_0$;

d) $\sum_{i\in b} w_{ip} \le W_p$, $p=1,\ldots,\underline{p}$.

For each block $b\in B$, the following sets can be determined:

- $Pr(b)$ is the set of operations $i\in \mathbf{N}\setminus b$ such that $(i,j)\in D^r$ for some $j\in b$;
- $Sc(b)$ is the set of operations $i\in \mathbf{N}\setminus b$ such that $(j,i)\in D^r$ for some $j\in b$;
- $Pra(b)$ is the set of operations $i\in \mathbf{N}\setminus b$ such that there exists a path in G^r from i to j for some $j\in b$;
- $Sca(b)$ is the set of operations $i\in \mathbf{N}\setminus b$ such that there exists a path in G^r from some $j\in b$ to i;
- vector $\eta(b)=(\eta^1,\ldots,\eta^{q^s})$, where $\eta^i=1$ if $b\cap e_i^s \neq \emptyset$ and $\eta^i=0$ otherwise.

The following dominance relation $\prec$ can be introduced over the set B. Let $b''\prec b'$ for blocks $b',b''\in$ B if the following sufficient dominance conditions are satisfied:

i1) $b'\subset b''$ and $t^b(b')=t^b(b'')$;

i2) $(b''\setminus b')\cap e=\emptyset$ for all $e\in \underline{E}^s$;

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

i4) $Pr(b'')\setminus Pr(b')\subseteq Pra(b')$ and $Sc(b'')\setminus Sc(b')\subseteq Sca(b')$.

The following proposition shows that $\prec$ is the dominance relation and the block b' can be deleted from B if there exists a block $b''\in B$ such that $b''\prec b'$. It is easy to see that the following proposition is true.

Proposition 1. Let $P(b')$ and $P(b'')$ be sets of collections from **P** in which the subsets b' and b'' are blocks from B, respectively. If $b''\prec b'$, then $min\{Q(P)|P\in P(b'')\}\le min\{Q(P)|P\in P(b')\}$.

Similarly, the conditional dominance relation $\prec_v$ can be introduced over the set $B(v)$. Let $b''\prec_v b'$ for $v\in V$ and $b', b''\in B(v)$ if $\eta(b'\cup v)=\eta(b''\cup v)$ and the conditions *i1*) –*i2*) hold.

Proposition 2. Let $P(v,b')$ and $P(v,b'')$ be sets of collections from **P** in which the subsets b', b'' are blocks from $B(v)$ and v is the set of all operations performed before these blocks. If $b''\prec_v b'$, then $min\{Q(P)|P\in P(v,b'')\}\le min\{Q(P)|P\in P(v,b')\}$.

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 digraph G is generated by Algorithm 2. In this algorithm, the set V is a current set of vertices of the digraph G; the set V_0 is a set of current vertices from V with zero outdegree; $\lambda(v)=0$, if there exist $i\in\mathbf{N}\setminus v$ and $j\in v\in V$ such that $a_{ij}=1$, and $\lambda(v)=1$ otherwise. The condition $v\cap b_k=\emptyset$ provides the constraints (9); $\lambda(v\cup b_k)=1$ provides the precedence constraints (10); $\eta(v)=0$ provides the constraints (11). The constraints (12), (13) and a part of (14) are taken into account during the formation of the set $B(v)$. Finally, Algorithm 1 ensures the constraints (14).

***Algorithm* 2**.

Step 0. Assign $v_0 \leftarrow \{\varnothing\}$, $V_0 \leftarrow \{v_0\}$, $V \leftarrow \{v_0\}$,

$\mu(v) \leftarrow 0$, $\eta^r(v) \leftarrow 0$, $r=1,...,q^s$.

Step k. **If** V_0 is empty **then** stop, **else**:

a) choose any vertex v in V_0 and exclude it from V_0;

b) construct $B(v)=\{b\in B|\ v\cap b=\varnothing, \lambda(v\cup b)=1\}$;

c) exclude block b from $B(v)$ if there is block $b'\in B(v)$ such that $b' \prec_v b$.

For each $b\in B(v)$ do:

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

ii) **if** $v' \notin V$ **then** include v' into V_0 and V, and assign $\eta(v') \leftarrow \eta(v)$;

if $\eta(b)=0$ **then** $\eta(v') \leftarrow \eta(v)$, **else**

$\eta(v') \leftarrow \eta(v)\nabla\eta(b)$; (where ∇ is a component-wise operation *OR*)

for all $j=1,..., q^s$ such that $\eta^j(v)\&\eta^j(b)=1$

if $v' \cap e_j^s = e_j^s$ **then** $\eta^j(v') \leftarrow 0$;

iii) add the arc (v,v') to D and assign $\Gamma_{vv'} \leftarrow \{0,1\}$, if $\eta(v)=0$, and $\Gamma_{vv'} \leftarrow \{0\}$, otherwise.

The complexity of the proposed approach for solving the initial problem is mainly determined by the size of the digraph G, in particular, the value of $|V|$. In real-life design problems, there are usually many constraints of type (10)-(14) and $|V|$ decreases exponentially with increasing the number of such constraints. For the problem (Dolgui *et al.*, 2000) with the number of operations equal to 15, $|V|=104$, $|D|=1202$, and the total calculation time is 1” on PC Pentium II-450.

For the problems with large number of operations, the digraph G may also be large. So, for testing problems with the number of operations equal to 150 the mean values of $|V|\approx 4200$, $|D|\approx 155800$ and the mean calculation time is about 35’. Such time is completely acceptable taking into account the problem complexity and its importance. For reducing the digraph G, special techniques are required. Some approaches to develop such techniques are given in (Dolgui *et al.*, 2000).

4. CONCLUSIONS

The generation of optimal or “good” design decision for considered lines is a very complex problem. In comparison with the classical assembly line balancing problem, the considered problem has many additional properties and constraints. The most important difference is the possibility to combine operations in a block. All the operations of the same block are executed simultaneously and have common parameters, for instance, the length of working stroke and the feed per minute, which determine block time.

In the paper, a new approach for solving transfer line balancing problem is proposed. It uses graph theory and reduces the initial problem to a constrained shortest parameterised path problem in a special digraph. The algorithms for generating the digraph and searching the optimal parameterized path are developed. This approach is more effective when the initial problem has a lot of technological constraints, which restrict the digraph size.

The proposed method is used in a multi-function computer-aided decision support system (process design, logical and physical layout) for production line design that is developing in co-operation between the Institute of Engineering Cybernetics of the National Academy of Sciences of Belarus and the University of Technology of Troyes (France).

ACKNOWLEDGMENT

This work is supported by INTAS project 00-217.

REFERENCES

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

Dolgui, A., N. Guschinsky and G. Levin (2000). Approaches to balancing of transfer lines with blocks of parallel operations. *Institute of Engineering Cybernetics, Preprint N°8, Minsk,* 2000, 42 pages.

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.

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

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

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

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