TRANSFER LINE BALANCING BY A COMBINED APPROACH

Alexandre Dolgui[1], Nikolai Guschinsky[1,2], Genrikh Levin[2]

[1] *Ecole des Mines de Saint Etienne*
158,Cours Fauriel 42033 Saint Etienne Cedex2 France
e-mail:{*dolgui,guschinsky*}*@emse.fr*
[2] *United Institute of Informatics Problems*
Surganov Str, 6, 220012 Minsk Belarus
e-mail:{*gyshin,levin*}*@newman.bas-net.by*

Abstract: A balancing problem for transfer lines with workstations in series and simultaneously executed blocks of operations is considered. Inclusion constraints related to operations and exclusion constraints with regard to blocks as well as precedence constraints are given. The problem is to choose blocks from a given set and to assign them to workstations while minimizing the line cost and satisfying the above constraints. A combined heuristic approach is proposed. It is based on decomposition of the initial problem into several sub-problems and solving them by an exact algorithm. Results of computational experiments are presented.

Keywords: Computer-aided design, machining, optimization, graph theory, integer programming, heuristics.

1. INTRODUCTION

Transfer lines are widely used in mechanical industry for mass production of a single type of product (Hitomi, 1996). The line is equipped with a common transfer mechanism and a common control system. All stations of the line perform their operations simultaneously, and failure of one station (or the necessity to change tools) results in stoppage of the line. 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 which is equipped by corresponding tools.

Dolgui et al. (2003a, 2005) consider the balancing problem for transfer lines with sequential execution of blocks at workstations. In this paper, this problem is investigated for the case when all blocks (spindle heads) of the same workstation are executed simultaneously. The workstation time is the maximal value among operation times of its spindle heads (block times) and the line cycle time is the maximum of workstation times.

It is supposed that a set **B** of all blocks, which can be used for the execution of a set **N** of operations, is known. For each block from **B**, its cost and execution time are given. The problem is to choose blocks from **B** and to assign them to workstations in such a way that:

i) the total *line cost* is as small as possible,
ii) a given productivity is reached (the line *cycle time* does not exceed a given value),
iii) all operations from **N** are assigned and a *partial order relation* on the set **N** (e.g., because of the presence of roughing, semi-finishing and finishing operations) is satisfied,

iv) *inclusion constraints* (some operations must be performed at the same workstation due to tolerance requirements) are respected;
v) *exclusion constraints* (some blocks cannot be allocated in the same workstation because of technological incompatibility of operations or constructional incompatibility of corresponding spindle heads) are not violated.

The problem is close to Simple Assembly Line Balancing Problem (Scholl, 1999). For Simple Assembly Line Balancing Problem (SALBP), two basic approaches are developed: branch and bound algorithms (Scholl and Klein, 1998) as well as heuristics and meta-heuristics (Helgenson and Birnie, 1961; Arcus, 1966; Rekiek *et al*., 2000). Surveys of main publications on SALBP are given in (Erel and Sarin, 1998; Scholl, 1999; Rekiek *et al*., 2002).

The SALBP methods cannot be directly used for solving the studied problem for the following reasons:

- several blocks from **B** can include the same operation from the set **N** and it is necessary to choose only one block;
- the workstation time is not the sum of block times;
- the objective function is not only the number of workstations but depends also on the number and the cost of blocks.

For this problem, two exact methods were developed. One of them (Dolgui *et al*., 2004a) uses a mixed integer programming (MIP) approach. The second method (Dolgui *et al*., 2003b) is based on transformation the initial problem into a constrained shortest path problem. Some dominance rules are used to reduce the size of the obtained graph. The results of computational experiments with heuristic relaxations of these rules are reported in (Dolgui *et al*., 2003c). In this paper another heuristic approach is proposed.

The rest of the paper is organized as follows. Section 2 deals with the problem statement. In Section 3, graph and MIP models are described. Section 4 presents a combined heuristic approach. Section 5 is dedicated to experimental results.

2. PROBLEM STATEMENT

The following notation is used for modeling the design problem considered:
N is the set of all operations;
B is the set of blocks (spindle heads) which can be used for the line design;
m is the number of workstations in a design decision;
n_k is the number of blocks of workstation k;
C_1 is the basic cost of one workstation;
N_{kl} is the set of operations of block l of workstation k;
$Pred(N_{kl})$ is the set of operations which must be executed before any operation from N_{kl};
$C_2(N_{kl})$ is the cost of the block N_{kl};
$N_k=\{N_{k1},\ldots,N_{kn_k}\}$ is the set of blocks from **B** which are executed at the workstation k;
$P=<N_1,\ldots,N_m>$ is a design decision.

It is assumed also that the line cannot involve more than m_0 workstations and each workstation cannot include more than n_0 blocks.

The line cost for design decision P can be estimated as: $C(P)=C_1m+\sum_{k=1}^{m}\sum_{l=1}^{n_k}C_2(N_{kl})$.

The constraints introduced in Section 1 can be represented in the following way:

i) An order relation over the set **N** is represented by the acyclic digraph $G^{OR}=(\mathbf{N},D^{OR})$. An arc $(i,j)\in\mathbf{N}\times\mathbf{N}$ belongs to the set D^{OR} if and only if the operation j must be executed after the operation i.

ii) Since all blocks of the same workstation are executed simultaneously, the blocks with block time over the required line cycle time can be excluded from **B** before optimization. Therefore this constraint can be omitted after such transformation.

iii) Exclusion conditions for the blocks of the same workstation can be represented by the graph $G^{DS}=(\mathbf{B},E^{DS})$ in which a pair $(N', N'')\in\mathbf{B}\times\mathbf{B}$ belongs to the set E^{DS} if and only if blocks N' and N'' cannot be allocated to the same workstation.

iv) Inclusion conditions for the operations of the same workstation can be represented by the graph $G^{SS}=(\mathbf{N},E^{SS})$ such that a pair $(i,j)\in\mathbf{N}\times\mathrm{N}$ belongs to the set E^{SS} if and only if operation i and j must be allocated to the same workstation.

So, the design problem can be reduced to finding a collection $P=<\{N_{11},\ldots, N_{1n_1}\}, \ldots, \{N_{m1},\ldots, N_{mn_m}\}>$, $N_{kl}\in\mathbf{B}$, satisfying the conditions:

$$C(P)=C_1m+\sum_{k=1}^{m}\sum_{l=1}^{n_k}C_2(N_{kl})\rightarrow min; \quad (1)$$

$$\bigcup_{k=1}^{m}\bigcup_{l=1}^{n_k}N_{kl}=\mathbf{N}; \quad (2)$$

$$N_{k'l'}\cap N_{k''l''}=\varnothing,\ k'l'\neq k''l'',\ k',k''=1,\ldots,m,\ l'=1,\ldots,n_{k'},\ l''=1,\ldots,n_{k''}; \quad (3)$$

$$Pred(N_{kl})\subseteq\bigcup_{r=1}^{k-1}\bigcup_{q=1}^{n_r}N_{rq},\ k=1,\ldots,m,\ l=1,\ldots,n_k; \quad (4)$$

$$\bigcup_{l=1}^{n_k}N_{kl}\cap e\in\{\varnothing,e\},\ e\in E^{SS},\ k=1,\ldots,m; \quad (5)$$

$$(N_{kl'}, N_{kl''}) \notin E^{DS},\ k=1,\ldots,m,\ l',l''=1,\ldots,n_k; \quad (6)$$

$$n_k \le n_0,\ k=1,\ldots,m; \quad (7)$$

$$m \le m_0. \quad (8)$$

The objective function (1) is the line cost; constraints (2)-(3) provide assigning all the operations from **N** and choosing only one block for each operation; (4) define the precedence constraints over the set **N**; (5) – (6) are the inclusion and exclusion constraints, respectively; (7) – (8) limit the number of workstations at the line and the number of blocks for each workstation.

3. EXACT METHODS

In this section, two exact methods are described which can be used in a combined heuristic algorithm. The first approach is based on transformation of the problem (1)-(8) into a problem of finding a constrained shortest path in a special digraph. The second approach reformulates the problem (1) – (8) in terms of MIP model.

3.1 Graph approach

Let **P** be a set of collections $P=\langle N_1,\ldots,N_k,\ldots,N_m\rangle$, satisfying the constraints (2)-(7). The set $v_k=\bigcup_{r=1}^{k}\bigcup_{l=1}^{n_r} N_{rl}$ can be considered as a state of the part after machining it at the k-th workstation. Let V be the set of all states for all $P\in\mathbf{P}$, including the initial state $v_0=\varnothing$ and the final state $v_N=\mathbf{N}$. A new acyclic directed multi-graph $G=(V,D)$ can be constructed, in which an arc d from vertex v' to vertex v'' belongs to D if and only if $v'\subset v''$, and there exists a collection P that contains $(N_{k1},\ldots,N_{kn_k})$ such that $\bigcup_{r=1}^{k-1}\bigcup_{l=1}^{n_r} N_{rl}=v'$ and $\bigcup_{l=1}^{n_k} N_{kl}=v''\setminus v'$ (see Fig. 1). The cost $\Theta(d)=\sum_{l=1}^{n_k} C_2(N_{kl})+C_1$ is assigned to the arc $d\in D$ as well as a set $Q(d)=\{q_1(d),\ldots,q_{n_k}(d)\}$ of block indices where $N_{kl}=B_{q_l(d)}$ for $l=1,\ldots,n_k$. It is assumed that the set **B** is arbitrarily enumerated and B_q is a block from **B** with index q from the set **Q**.

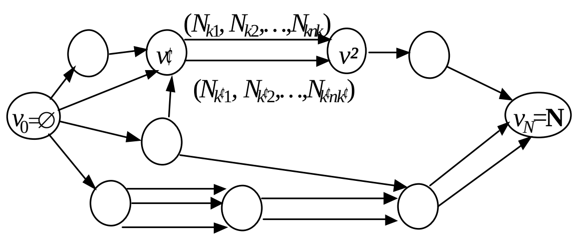

Fig. 1. Multigraph *G*.

The path $x(P)=(d_1(x(P)),\ldots,d_{m(x(P))}(x(P)))$ from the set $\mathbf{X}$ of all paths $x=(d_1(x),\ldots,d_k(x),\ldots,d_{m(x)}(x))$ in the digraph G from v_0 to v_N can be associated with each design decision $P\in\mathbf{P}$. On the other hand, each path $x\in\mathbf{X}$ corresponding to a collection $P(x)=\langle N(d_1(x)),\ldots,N(d_k(x)),\ldots,N(d_{m(x)}(x))\rangle$ where $N(d_k(x))=\{N(j_1(d_k(x))),\ldots,N(j_{n_k}(d_k(x)))\}$ satisfies constraints (2)-(7) but may violate constraint (8).

Thus the initial problem (1)-(8) can be transformed to a problem of finding the shortest path in multi-graph G with at most m_0 arcs. This problem is stated as follows:

$$\Theta(x)=\sum_{k=1}^{m(x)} Q(d_k(x)) \to \min, \quad (9)$$

$$x\in\mathbf{X}, \quad (10)$$

$$m(x)\le m_0. \quad (11)$$

3.2 MIP model

To formulate the problem (1) – (8) as a MIP problem, the following variables and additional parameters are introduced:

- binary variables x_{qk}, where $x_{qk}=1$ if the block $B_q\in\mathbf{B}$ is assigned to the station k and $x_{qk}=0$ otherwise, $k=1,\ldots,m_0$;
- a set of indices $Q(i)=\{q\in\mathbf{Q}|i\in B_q\}$ of blocks from **B** that include the operation $i\in\mathbf{N}$;
- a segment $KB(q)=[kb_{min}(q),kb_{max}(q)]$ of the station indices where the block $B_q\in\mathbf{B}$ can be assigned;
- a segment $KO(j)=[ko_{min}(j),ko_{max}(j)]$ of the station indices where the operation $j\in\mathbf{N}$ can be assigned;
- a lower bound m^* of the number of stations;
- variables $Y_k\in[0,1]$, $k=m^*+1,\ldots,m_0$, which indicate if the station k exists or not.

The objective function (12) provides the line cost minimization:

$$\sum_{k=m^*+1}^{m_0} C_1 Y_k + \sum_{k=1}^{m_0}\sum_{q\in\mathbf{Q}} C_2(B_q)x_{qk} \to \min \quad (12)$$

The following constraints ensure the execution of all the operations from **N** and each operation from **N** in one station only:

$$\sum_{q\in Q(i)}\ \sum_{k\in KB(q)} x_{qk}=1,\ i\in\mathbf{N} \quad (13)$$

The precedence constraints are not violated if:

$$\sum_{r\in Q(i)}\ \sum_{l=k_{\min}(r)}^{\min(k-1,k_{\max}(r))} x_{rl} \ge \sum_{q\in Q(j)} x_{qk},$$

$$(i,j)\in D^{OR},\ k\in KO(j) \quad (14)$$

The inclusion constraints for operations are satisfied if:

$$\sum_{r\in Q(i)} x_{rk} = \sum_{q\in Q(j)} x_{qk}\ ,\ (i,j)\in E^{SS}, k\in KO(j) \qquad (15)$$

The exclusion constraints for blocks are respected if:

$$x_{rk}+x_{qk}\le 1,\ (B_r,B_q)\in E^{DS},\ k\in KB(r)\cap KB(q) \qquad (16)$$

The number of blocks at each station is lower than n_0 if:

$$\sum_{r\in\{q\in \mathbf{Q}|k\in KB(q)\}} x_{rk} \le n_0,\ k=1,\dots,m_0 \qquad (17)$$

The following constraints define the existence of stations:

$$Y_k \geq x_{rk},\ k \ge m^*+1,\ r\in\{q\in \mathbf{Q}|k\in KB(q)\} \qquad (18)$$

$$\sum_{r\in\{q\in \mathbf{Q}|k\in KB(q)\}} x_{rk} \ge 1,\ k=1,\dots, m^* \qquad (19)$$

Station k can be created if the station (k-1) exists at the line:

$$Y_{k-1}-Y_k\ge 0,\ k=m^*+2,\dots,m_0. \qquad (20)$$

Segments $KB(q)$ for each block $B_q\in \mathbf{B}$ and segments $KO(j)$ for each operation $j\in \mathbf{N}$ as well as m^* can be calculated by the following algorithm. In this algorithm, $Pred(j)$ and $Succ(j)$ are the sets of immediate predecessors and successors for the operation $j\in \mathbf{N}$; *imp* calculates the number of improved bounds at the current iteration of the algorithm.

Algorithm 1.
Step 1. Set imp=0, $ko_{min}(j)$ =1 and $ko_{max}(j)= m_0$ for all $j\in \mathbf{N}$.

Step 2. For each $j\in \mathbf{N}$
a) compute $lj = \max\{ko_{min}(i)+1|i\in Pred(j)\}$ and $uj= \min\{ko_{max}(i)-1|\ i\in Succ(j)\}$;
b) if $lj > m_0$ or $uj < 1$ then stop (the problem has no feasible solutions);
c) If $lj>ko_{min}(j)$ then set $ko_{min}(j)=lj$ and $imp= imp+1$;
d) If $uj<ko_{max}(j)$ then set $ko_{max}(j)=uj$ and $imp=imp+1$.

Step 4. For each $(i,j)\in E^{SS}$ compute $lm=\max[ko_{min}(i), ko_{min}(j)]$, $um=\min[ko_{max}(i), ko_{max}(j)]$ and set $ko_{min}(i)= ko_{min}(j)=lm$, $ko_{max}(i)=ko_{max}(j)=um$.

Step 4. If $imp\_b > 0$ then set imp=0 and go to *Step* 2.

Step 5. Set $m^*=\max\{ko_{min}(j)|j\in \mathbf{N}\}$, $kb_{min}(q)= \max\{ko_{min}(j)|j\in B_q\}$ and $kb_{max}(q) =\min\{ko_{max}(j)|j\in B_q\}$ for each $B_q\in \mathbf{B}$.

4. COMBINED HEURISTIC APPROACH

The above exact algorithms are applicable for small and medium size. In this paper, a combined heuristic approach is proposed, which tries to improve a feasible solution obtained by an heuristics. It is based on decomposition of the initial problem into several sub-problems in accordance with a heuristic solution and solving the sub-problems by an exact method.

Let TR_{tot} be the current number of trials, TR_{nimp} be the number of trials that do not improve the current best solution, C_{min} be the cost of the best solution, C_{heur} be the cost of the current heuristic solution, *max_st_sub* and *max_op_sub* be the maximal allowable number of stations and operations in a sub-problem, respectively.

Algorithm 2.
Step 1. Set $C_{min} = \infty$, $TR_{tot} = 0$, $TR_{nimp} = 0$.

Step 2. A current solution P_{heur} of the initial problem with the cost C_{heur} is generated by an heuristics. If $C_{heur} < C_{min}$ then set $C_{min} = C_{heur}$ and $P_{min}= P_{heur}$.

Step 3. A series of sub-problems is generated based on the solution $P_{heur}= \langle N_1,\cdots,N_{k,}\cdots,N_{m_{heur}}\rangle$. The r-th sub-problem is to assign a set of operations $\mathbf{N}_r$ using a set of blocks $\mathbf{B}_r$. The set $\mathbf{N}_r$ includes the operations from m_r stations of P_{heur} beginning from the station $\sum_{i=1}^{r-1} m_i$ and the set $\mathbf{B}_r$ consist of blocks B from $\mathbf{B}$ that involve operations from $\mathbf{N}_r$ only. Value m_r is chosen at random within [1, *max_st_sub*] and then can be modified so that the total number of operations in a sub-problem does not exceed *max_op_sub* and the total sum of m_r is not greater than m_{heur}. For each sub-problem, constraints (2) – (6) are transformed by replacing the sets $\mathbf{N}$ and $\mathbf{B}$ with the subsets $\mathbf{N}_r$ and $\mathbf{B}_r$, and then removing those constraints which include operations from $\mathbf{N}\backslash\mathbf{N}_r$ or blocks from $\mathbf{B}\backslash\mathbf{B}_r$.

Step 4. A solution P_{comb} is composed by combining the solutions of sub-problems. If the cost C_{comb} of the obtained solution is less than C_{min} then set $C_{min}= C_{comb}$, $P_{min}= P_{comb}$, $TR_{nimp} = 0$ and keep the current solution as the best, set $TR_{nimp} = TR_{nimp} + 1$ otherwise.

Step 5. Set $TR_{tot} = TR_{tot} + 1$.

Step 6. Stop if one of the following conditions holds:

- a given solution time is exceeded;
- TR_{tot} is greater than the maximum number of iterations authorized;
- TR_{nimp} is greater than a given value;
- C_{min} is lower than a given cost value.

Go to *Step* 2 otherwise.

Algorithm 2 can be modified in such a way that several attempts of decomposition for a heuristic solution can be done; not all sub-problems are to be solved by an exact algorithm (an heuristic solution can be used); the type of the exact method can be chosen for each sub-problem in dependence of its parameters (number of operations, number of blocks, the order strength of precedence constraints). The order strength (Scholl, 1999) is defined as the density of the transitive closure of the precedence graph.

5. EXPERIMENTAL STUDY OF ALGORITHM

The purpose of this study is to compare the proposed algorithm on the quality of obtained solutions and running time with the exact methods and heuristics. The studied modifications of Algorithm 2 use the graph algorithm for solving sub-problems and differ in the number of decomposition attempts. Heuristic solutions were generated by the algorithm (Dolgui *et al*., 2004b). Experiments were carried out on HP Omnibook x86 Family 6 Model 8. The results are presented in Tables 1 - 4. They correspond to 12 series of 10 test instances which were generated in random way for different values of |**N**|, |**B**| and $p(G^{OR})$, $p(G^{DS})$, $p(G^{SS})$, where *p*(*G*¢) is the ratio between the number of generated edges (arcs) in *G*¢ and the number of edges (arcs) in the complete graph (digraph) with the same number of vertices. In all examples $p(G^{SS})$= 0.01. Choosing $p(G^{SS})$= 0.01 is justified since the graph G^{SS} should be coordinated with graphs G^{OR} and G^{DS} to provide feasible solutions. So, edges from the graph G^s are removed if they contradict graph G^{DS} or belong to a path in G^{OR}.

In these tables, the following abbreviations are used:

- TC (parameters $p(G^{OR})$, order strength, $p(G^{DS})$ of test instances);
- GA (exact graph algorithm);
- HA (heuristic algorithm);
- CHA1 (combined heuristic algorithm with one attempt for decomposition);
- CHA2 (combined heuristic algorithm with two attempts for decomposition);
- CHA3 (combined heuristic algorithm with three attempts for decomposition);
- PM (performance measures);
- RT (running time in seconds);
- SD (solution deviation of the obtained cost from the best known solution in percents).

Indices min, max, av for RT and SD mean the minimal, maximal and average values, respectively.

Table 1 Results for |**N**|=25, |**B**|=75

TC	PM	GA	HA	CHA1	CHA2	CHA3
0.05, 0.06, 0.05	RT_{min}	0.86	1.88	10	10	10
	RT_{max}	148.2	2.64	10	10	10
	RT_{av}	49.67	2.29	10	10	10
	SD_{min}	0	0.01	0	0	0
	SD_{max}	0	17.8	35.63	35.63	35.63
	SD_{av}	0	6.69	9.40	9.83	10.79
0.10, 0.18, 0.10	RT_{min}	0.1	1.91	4.27	7.17	6.97
	RT_{max}	13.26	3.12	10	10	10
	RT_{av}	2.202	2.546	7.93	9.27	9.63
	SD_{min}	0	0	0	0	0
	SD_{max}	0	14.74	8.71	8.71	13.33
	SD_{av}	0	6.99	2.46	2.85	3.78
0.15, 0.34, 0.15	RT_{min}	0.05	2.32	1.66	3.14	4.60
	RT_{max}	1.53	3.19	10	10	10
	RT_{av}	0.40	2.73	5.87	8.27	9.01
	SD_{min}	0	0.01	0	0	0
	SD_{max}	0	11.33	3.37	3.37	3.37
	SD_{av}	0	6.03	1.63	1.63	1.63

Table 2 Results for |**N**|=50, |**B**|=150

TC	PM	GA	HA	CHA1	CHA2	CHA3
0.05, 0.12, 0.05	RT_{min}		4.22	16.88	17.55	18.22
	RT_{max}		5.92	20	20	20
	RT_{av}		5.35	19.42	19.63	19.77
	SD_{min}		0	-11.51	-11.77	-11.77
	SD_{max}		0	2.2	2.18	2.18
	SD_{av}		0	-5.32	-5.72	-5.48
0.10, 0.33, 0.10	RT_{min}	1.20	4.21	11.09	13.62	12.15
	RT_{max}	68.66	8.32	20	20	20
	RT_{av}	17.25	6.33	16.70	18.74	19.044
	SD_{min}	0	1.96	0	0	0
	SD_{max}	0	19.19	10.57	10.57	10.57
	SD_{av}	0	8.87	3.09	3.09	3.29
0.15, 0.54, 0.15	RT_{min}	0.52	5.08	5.67	9.87	13.72
	RT_{max}	24.97	11.33	20	20	20
	RT_{av}	5.915	7.24	17.24	18.24	19.19
	SD_{min}	0	0	0	0	0
	SD_{max}	0	18.8	13.92	13.92	13.92
	SD_{av}	0	8.42	3.34	3.34	3.68

Table 3 Results for |**N**|=75, |**B**|=225

TC	PM	GA	HA	CHA1	CHA2	CHA3
0.05, 0.14, 0.05	RT_{min}		6.69	22.53	25.30	28.14
	RT_{max}		20.86	30	30	30
	RT_{av}		10.33	27.45	28.62	29.48
	SD_{min}		0	-5.42	-5.42	-5.42
	SD_{max}		0	0	0	0
	SD_{av}		0	-2.82	-2.82	-2.82
0.10, 0.45, 0.10	RT_{min}	3.45	5.69	16.99	17.22	17.46
	RT_{max}	273.8	11.76	30	30	30
	RT_{av}	73.19	8.787	26.02	26.14	26.26
	SD_{min}	0	4.1	0	0	0
	SD_{max}	0	24.69	18.41	18.41	18.41
	SD_{av}	0	11.89	6.54	6.41	6.41
0.15, 0.63, 0.15	RT_{min}	1.17	6.36	12.74	19.00	19.10
	RT_{max}	45.65	24.18	30	30	30
	RT_{av}	12.62	12.12	24.61	26.16	26.99
	SD_{min}	0	3.99	1.39	1.39	1.39
	SD_{max}	0	13.89	5.07	5.07	5.07
	SD_{av}	0	7.91	3.19	3.19	3.19

Table 4 Results for |**N**|=100, |**B**|=300

TC	PM	GA	HA	CHA1	CHA2	CHA3
0.05, 0.22, 0.05	RT_{min}		8.58	25.63	26.16	26.74
	RT_{max}		327.3	40	40	40
	RT_{av}		48.16	36.28	36.77	37.26
	SD_{min}		0	-16.95	-13.11	-13.11
	SD_{max}		0	0	0	0
	SD_{av}		0	-4.49	-4.06	-4.06
0.10, 0.55, 0.10	RT_{min}	11.72	9.96	29.70	30.31	30.96
	RT_{max}	595.6	24.50	40	40	40
	RT_{av}	150.4	16.17	38.29	38.40	38.51
	SD_{min}	0	3.17	0	0	0
	SD_{max}	0	15.11	8.84	8.84	8.84
	SD_{av}	0	10.11	5.15	5.15	5.15
0.15, 0.73, 0.15	RT_{min}	4.78	11.13	33.21	33.70	34.16
	RT_{max}	20.40	28.78	40	40	40
	RT_{av}	11.75	17.41	38.91	38.99	39.07
	SD_{min}	0	3.36	0.85	0.85	0.85
	SD_{max}	0	17.02	11.26	11.26	11.26
	SD_{av}	0	11.06	6.40	6.65	6.65

The maximal available time for CHA1, CHA2 and CHA3 was given as: 10 sec for test instances with |**N**|=25, |**B**|=75; 20 sec for |**N**|=50, |**B**|=150; 30 sec for |**N**|=75, |**B**|=225; 40 sec for |**N**|=100, |**B**|=300.

5. CONCLUSION

A combined heuristic approach has been presented to find a "good" design decision for balancing a transfer line with simultaneously activated spindle heads at workstations. It is supposed that spindle heads are to be chosen from a given set. The algorithm uses decomposition of the initial problem into several sub-problems in accordance with a known solution and tries to improve line balancing by composition of exact solutions of sub-problems.

The proposed algorithm is relatively efficient. This conclusion is based on its experimental comparison with an exact graph approach and other heuristics. For moderate size problems (less than 50 operations and 150 blocks) or when their order strength is relatively large, the exact graph approach is acceptable in terms of computation time, and thus the exact solutions have been obtained for this type of tests. For test instances, the algorithm was capable to improve the quality of obtained solutions in two times with regard to initial heuristic solutions. The experiments show that the heuristics performances depend on the problem characteristics (order strength, constraints of compatibility) as well on control parameters and available time for solution.

ACKNOWLEDGMENT

This work is financially supported by ISTC project B-986 and INTAS Project 03-51-5501.

REFERENCES

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

Dolgui, A., B. Finel, N. Guschinsky, G. Levin, and F. Vernadat (2003a). Some optimization approaches for transfer lines with blocks of parallel operations. In: *Preprints of the 7th IFAC Symposium on Intelligent Manufacturing Systems*, (L. Monostori, B. Kadar, G. Morel (Eds.)), Budapest, Hungary, pp. 261-266.

Dolgui, A., N. Guschinsky, N. and G. Levin (2003b). Balancing production lines composed by series of workstations with parallel operations blocks. In: *Proceedings of the 2003 IEEE International Symposium on Assembly and Task Planning*, Besançon, France, pp. 122-127.

Dolgui, A., N. Guschinsky, N. and G. Levin (2003c). Graph approach for transfer lines balancing: exact and heuristic methods. In: *Proceedings of the 3rd International Conference Research and Development in Mechanical Industry*, Herceg Novi, Serbia and Montenegro, CD-ROM, 9 p.

Dolgui, A., N. Guschinsky, G. Levin, M.-A. Louly and S. Belmokhtar (2004a). Balancing of transfer lines with simultaneously activated spindles. In: *Preprints proceedings of the 11th IFAC Symposium in Information Control Problem In Manufacturing*, Salvador de Bahia, Brazilia, CD-ROM, 6 p.

Dolgui, A., N. Guschinsky, N. and G. Levin (2004b). Heuristic algorithms for balancing transfer lines with simultaneously activated spindles. In: *Preprints proceedings of the IFAC Conference on Manufacturing, Modelling, Management and Control*, Athens, Greece, CD-ROM, 6 p.

Dolgui, A., B. Finel, N. Guschinsky, G. Levin and F. Vernadat (2005). A heuristic approach for transfer lines balancing. *Journal of Intelligent Manufacturing*, **16**(2), 159-171.

Erel, E. and S.C. Sarin (1998). A survey of the assembly line balancing procedures. *Production Planning and Control*, **9**(5), 414-34.

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

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

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

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

Scholl, A. (1999). *Balancing and sequencing of assembly lines*. Physica-Verlag, Heidelberg.