



HAL
open science

Balancing large-scale machining lines with multi-spindle heads using decomposition

Alexandre Dolgui, Brigitte Finel, Olga Battaïa, Nikolay Guschinsky, Genrikh Levin, Francois B. Vernadat

► **To cite this version:**

Alexandre Dolgui, Brigitte Finel, Olga Battaïa, Nikolay Guschinsky, Genrikh Levin, et al.. Balancing large-scale machining lines with multi-spindle heads using decomposition. *International Journal of Production Research*, 2006, 44 (18-19), pp.4105-4120. 10.1080/00207540600632232 . hal-00512895

HAL Id: hal-00512895

<https://hal.science/hal-00512895v1>

Submitted on 1 Sep 2010

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

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



(18th ICPR) Balancing large-scale machining lines with multi-spindle heads using decomposition

Journal:	<i>International Journal of Production Research</i>
Manuscript ID:	TPRS-2005-IJPR-0543
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	10-Dec-2005
Complete List of Authors:	Dolgui, Alexandre; Ecole des Mines de St Etienne, Division for Indl Engineering and Computer Sciences Finel, Brigitte; ENIM, LGIPM Guschinskaya, Olga; Ecole des Mines de ST Etienne, Division for Indl Engineering and Computer Sciences Guschinsky, Nikolay; Academy of Science, Operational Research Levin, Genrikh; Academy of Science, Operational Research Vernadat, Francois; ENIM, LGIPM
Keywords:	LINE BALANCING, DESIGN OF PRODUCTION SYSTEMS, DISCRETE OPTIMISATION, FLOW LINES, INTEGER PROGRAMMING, MACHINING PROCESSES, PACED LINES, TRANSFER LINES
Keywords (user):	



Balancing large-scale machining lines with multi-spindle heads using decomposition

A. DOLGUI¹, B. FINEL², O. GUSCHINSKAYA¹, N. GUSCHINSKY³, G. LEVIN³ and
F. VERNADAT²

The paper deals with optimal logical layout design for a type of machining lines. Such lines are made of many machine-tools (workstations) located in sequence. On each workstation there are several multi-spindle heads. A spindle-head does not execute one operation but a block of machining operations simultaneously. The problem studied in this paper consists in finding the best partition of the set of all operations to be executed on the line into blocks and workstations. The objective is to minimize the number of blocks and workstations. An optimal decision must satisfy a desired productivity rate (cycle time) and precedence and compatibility constraints for machining operations. A heuristic approach based on decomposition of a Mixed Integer Programming (MIP) model is developed. Two ways of forming sub-problems are proposed. One of them treats the obtained subsets independently. The second one aggregates the solution of the previous sub-problems. Results of their computational evaluation are reported.

Key words: Machining lines, Line balancing, Optimization, Decomposition, Heuristics.

Submitted to IJPR, December 9, 2005

¹ Ecole des Mines de Saint Etienne, France
{dolgui, guschinskaya}@emse.fr

² ENIM/University of Metz, France
{finel, vernadat}@anim.fr

³ United Institute of Informatics Problems of National Academy of Sciences, Belarus
{gyshin,levin}@newman.bas-net.by

1. Introduction

Machining lines are widely used in industry (Groover 1987, Askin and Standridge 1993, Hitomi 1996, Dashchenko 2003). Usually, the design of this type of production systems involves the following three interconnected steps: (1) the selection of necessary operations to be executed at the line; (2) the logical synthesis of the manufacturing process, which consists in grouping the operations into blocks and the blocks into stations (*logical layout*); and (3) the physical layout and design of equipment (tools, spindle-heads, etc.) depending on the logical layout contained.

This paper concentrates on the logical layout design stage. At this stage, all machining operations as well as essential constraints of physical layout are known. It is necessary to define a partition of all the operations into subsets in order to minimize the number of workstations and spindle heads (pieces of equipment). Such *spindle heads* and *workstations* represent a significant investment cost and are used for a long exploitation period. Thus finding a good (and if possible the best) logical layout design decision *minimizing their number* is a crucial problem.

To minimize the number of workstations and spindle heads as well as the occupied area for the considered line, the operations are grouped into *blocks*. All operations of the same block are executed simultaneously by one spindle head. All blocks (spindle heads) of the same station are executed (activated) sequentially; all stations are linearly ordered. There are no buffers between stations. The assignment of blocks to a station defines at the same time the order of their activation on the station.

The processing time of each spindle head depends on the parameters (*working stroke length* and *feed rate*) of operations assigned to the corresponding block. The station processing time is equal to the sum of *block times* (for all blocks assigned to this station). The bottleneck station defines the machining *line cycle time*.

In this paper, the objective function is as follows:

$$\text{Min } m W_1 + n W_2, \quad (1)$$

1
2 where

3
4
5 m is the number of stations (multiple spindle machine-tools),

6
7
8 n is the total number of blocks (spindle heads),

9
10
11 W_1 and W_2 are the weights (relative costs) of one station and of one spindle head in a design
12 decision, respectively.

13
14
15
16 The problem considered in this paper extends the well-known *Simple Assembly Line Balancing*
17 *Problem* (SALBP). Techniques used to solve SALBP can be found in literature, *e.g.* (Baybars 1986,
18 Ghosh and Gagnon 1989, Erel and Sarin 1998, Scholl and Klein 1998, Sprecher 1999, Gadinov and
19 Wilhelm 2000, Rekiek *et al.* 2002).

20
21
22
23
24
25
26 For the investigated problem, which cannot be directly solved by SALBP methods due to some
27 specificities (groups of operations are executed simultaneously, operations times are not known in
28 advance, criterion include the number of blocks and stations, etc.), two exact optimization methods
29 based on Mixed Integer Programming (MIP) and graph approaches, respectively, were suggested in
30 (Dolgui *et al.* 1999, Dolgui *et al.* 2000, Dolgui *et al.* 2006a, Dolgui *et al.* 2006b). These methods
31 can be used only for small size problems. For large-scale problems, two heuristic algorithms RAP
32 and FSIC were developed in (Dolgui *et al.* 2005), these algorithms are based on generalizations of
33 the COMSOAL method (Arcus 1966).

34
35
36
37
38
39
40
41
42
43
44
45 In this paper, a new heuristic procedure is suggested for large-scale problems. It is based on
46 decomposition of the whole set of operations into several subsets. On the basis of these subsets MIP
47 models are formulated and then solved by a MIP solver. Two ways of MIP models formulation are
48 investigated. One of them treats the obtained subsets independently. The second one aggregates the
49 solution of the previous sub-problems. The experimental results are reported.

50
51
52
53
54
55
56
57
58
59
60
The paper is organized as follows. Section 2 introduces some notations and the problem. In
Section 3, the decomposition procedure is suggested. In Section 4, experimental results are
reported. Conclusion remarks are given in Section 5.

2. Notations and problem statement

2.1 Objective and constraints

An illustration of the type of lines studied in this paper is given in Figure 1. The parts are moved from station to station. On each station there is at least one multiple spindle head with different tools. The time spends on a multi-spindle head depends on the common parameters of their operations (tools). The following parameters of the operations are given before optimization: the working stroke length and the feed rate. The spindle heads of the same station are activated sequentially.

[Insert Fig. 1 about here]

The objective of this paper is to investigate a new method for optimal design of logical layout for such lines (several hundreds of operations), *i.e.* to assign the operations of a given set to subsets corresponding to stations (machines) and to blocks (spindle heads) in such a way that:

- The objective function (1) is as small as possible;
- The specified line cycle time is not exceeded (*i.e.*, the obtained line cycle time is not greater than the specified one);
- All the constraints for machining operations are satisfied.

The following precedence and compatibility constraints are taken into account:

- A *partial order relation* on the set of all operations to be machined. This defines a set of possible operation sequences (precedence constraints). In this problem when an operation i precedes an operation j , the operation i can be executed before the operation j , or in the same block with j , but the operation j cannot be executed before the operation i .
- *Inclusion constraints* for blocks and stations, which oblige to perform some groups of operations in the same block or at the same station (*e.g.*, because of a required machining tolerance);

- *Exclusion constraints* for blocks and stations, which forbid execution of some groups of operations within one block or at the same station (*e.g.*, due to design rules or because of manufacturing incompatibility of operations).

2.2 Notations

The following notations will be used throughout this paper:

\mathbf{N}	given set of operations to be machined;
T_0	desired machining line cycle time;
m_0	maximal authorized number of stations;
n_0	maximal authorized number of blocks at one station;
l_j	processing time of the operation $j \in \mathbf{N}$;
s_j	feed rate of the operation $j \in \mathbf{N}$;
τ^b	an additional auxiliary time for execution of a block;
τ^s	an additional auxiliary time for execution of all blocks of a station;
t_j	execution time of the operation $j \in \mathbf{N}$;
t_{ij}	execution time of the operations $i, j \in \mathbf{N}$ when they are grouped in one block;
$Pred(j)$	set of direct predecessors of the operation $j \in \mathbf{N}$;
\overline{EB}	collection of operation sets representing the exclusion (<i>i.e.</i> , impossibility) constraints for blocks. Operations of the same set cannot be assigned to the same block;
\overline{ES}	collection of operation sets representing the exclusion constraints for stations. Operations of the same set cannot be assigned to the same station;
EB	collection of operation sets representing the inclusion (<i>i.e.</i> , necessity) constraints for the blocks. All operations of the same set must be assigned to the same block;

- 1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
- ES collection of operation sets representing the inclusion (*i.e.*, necessity) constraints for the stations. All operations of the same set must be assigned to the same station.
- k station number, $k=1, 2, \dots, m$;
- q index of the block, *e.g.*, $q=(k-1)n_0+l$ for the block l of the station k ;
- q_0 maximal possible value of q , $q_0=m_0n_0$;
- $S(k)$ set of block indices for the station k ;
- n_k number of blocks at the station k ;
- $C(k)$ cost of the station k with all its blocks, *i.e.* $C(k) = W_1 + W_2n_k$;
- B_q set of operations of the block q ;
- $Q(j)$ interval of indices q (blocks) where operation j can be potentially assigned ($Q(j)=[q^-(j), q^+(j)]$);
- $K(j)$ interval of indices k (stations) where operation j can be potentially assigned ($K(j)=[k^-(j), k^+(j)]$);
- e set of operations, element of collection EB, ES, \overline{ES} or \overline{EB} ;
- X_{jq} a binary decision variable (1 if operation j is assigned to block q and 0 otherwise);
- F_q an auxiliary variable for determining the time of block q ($F_q \geq 0$);
- Y_q an auxiliary variable that indicates if the block q exists or not ($Y_q \in \{0, 1\}$).

The time $t^b(N)$ of a block $N \subseteq \mathbf{N}$ is determined by the working stroke length $L(N)$ of the corresponding spindle head, and by its feed per minute $S(N)$ and is equal to $t^b(N)=L(N)/S(N)+\tau^b$ where $L(N)=\max\{l_j|j \in N\}$ and $S(N)=\min\{s_j|j \in N\}$. The time of block N can be also calculated as the maximal value among t_{ij} for all pairs of operations from N (Dolgui *et al.* 2000): $t^b(N)=\max_{i,j \in N}\{t_{ij}\}$ where $t_{ij}=\max\{l_i, l_j\} / \min\{s_i, s_j\}+\tau^b$.

The time $t^s(N_k)$ of station k is equal to $t^s(N_k) = \sum_{l=1}^{n_k} t^b(N_{kl}) + \tau^s$.

2.2 MIP model

Taking into account that station k exists if at least one operation is assigned to block $q = (k-1)n_0+1$, the considered problem can be formulated as the following Mixed Integer Program (MIP) and solved using a MIP solver (ILOG Cplex, Xpress MP, ...):

$$\text{Min } W_1 \sum_{k=1}^{m_0} Y_{(k-1)n_0+1} + W_2 \sum_{q=1}^{q_0} Y_q \quad (2)$$

Subject to:

$$\sum_{q \in Q(j)} X_{jq} = 1 \quad (3)$$

$$\sum_{i \in \text{Pred}(j)} \sum_{q' \leq q, q' \in Q(i)} X_{iq'} \geq X_{jq} |\text{Pred}(j)|, \quad q \in Q(j) \quad (4)$$

$$X_{jq} = X_{iq}, \quad e \in EB, \quad i, j \in e, \quad q \in Q(j) \quad (5)$$

$$\sum_{q \in S(k) \cap Q(j)} X_{jq} = \sum_{q \in S(k) \cap Q(i)} X_{iq}, \quad e \in ES, \quad i, j \in e, \quad k \in K(j) \quad (6)$$

$$\sum_{j \in e} X_{jq} \leq |e| - 1, \quad e \in \overline{EB}, \quad q \in \bigcap_{j \in e} Q(j) \quad (7)$$

$$\sum_{j \in e} \sum_{q \in S(k) \cap Q(j)} X_{jq} \leq |e| - 1, \quad e \in \overline{ES}, \quad k \in \bigcap_{j \in e} K(j) \quad (8)$$

$$F_q \geq t_j X_{jq}, \quad q \in Q(j) \quad (9)$$

$$F_q \geq t_{ij} (X_{iq} + X_{jq} - 1), \quad i < j, \quad q \in Q(i) \cap Q(j) \quad (10)$$

$$\tau^s + \sum_{q \in S(k)} F_q \leq T_0, \quad k=1, 2, \dots, m_0 \quad (11)$$

$$Y_q \geq X_{jq}, \quad q \in Q(j) \quad (12)$$

$$Y_{q-1} - Y_q \geq 0, \quad q \in S(k) \setminus \{(k-1)n_0+1\}, \quad k=1, 2, \dots, m_0 \quad (13)$$

$$Y_{(k-1)n_0+1} - Y_{k*n_0+1} \geq 0, \quad k=2, 3, \dots, m_0 \quad (14)$$

where $i, j \in \mathbf{N}$.

According to equation (3), each operation of \mathbf{N} is assigned to one block only. Inequalities (4) provide that all the predecessors of operation j are assigned either before operation j or in the same block with operation j . Equations (5) and (6) enforce assignment of any pair of operations i and j of set $e \in EB$ ($e \in ES$) to the same block (station), respectively. In accordance with equation (7), at most $|e|-1$ operations of set $e \in \overline{EB}$ may be assigned to the same block. Expressions (8) represent the exclusion constraints for assignment of operations to stations. Equations (9) and (10) provide that F_q is not less than the time of execution of all the operations assigned to the block q . Inequalities (11) provide the desired productivity level, *i.e.*, for each workstation of the line, the station time $\tau^s + \sum_{q \in S(k)} F_q$ does not exceed the line cycle time T_0 . In accordance with equation (12), the block q does not exist in the line if no operations are assigned to it. Equations (13) prevent creation of block q for station k if block $q-1$ does not exist for this station. Equations (14) provide that station k can be created only if station $k-1$ exists in the line.

This model is an improvement of the models suggested in (Dolgui *et al.* 2000, Dolgui *et al.* 2006b). To calculate $q^-(j)$, $q^+(j)$, $k^-(j)$, and $k^+(j)$, a pre-processing procedure has developed in (Dolgui *et al.* 2000, Dolgui *et al.* 2006b).

2.3 An industrial example

The problem can be illustrated by the following *industrial example*.

The part to be machined is shown in Figure 2. It is necessary to machine a plane surface (F) and 8 holes (H1 – H8). The set \mathbf{N} consists of 30 operations. The operation parameters are given in Table 1. The precedence graph is given in Figure 3. The desired line cycle time is $T_0 = 2.8$ min, $\tau^b = 0.2$, $\tau^s = 0.4$ min, the maximum number of stations on the line is $m_0 = 5$ and the maximum number of spindle-heads at a station is $n_0 = 4$.

[Insert Fig. 2 about here]

[Insert Table 1 about here]

[Insert Fig. 3 about here]

The compatibility constraints are the following. Operations 1 and 2 are not compatible in one position with any other operation from N since they should be executed by a milling machine. All operations for machining H1 are not compatible with operations for H7 due to constraints on minimal center distances. The same type of constraints exists between operations for H2 and H7, H3 and H8, and H4 and H8. Operations 22, 23, 29 and 30 cannot be grouped in one block with any operation since the minimal center distance for these operations is 125 mm. Operations 13 and 16 as well as 18 and 25 have to be executed in one block due to tolerance requirements. Operations 13, 16, 23 and 30 should be executed in the station since they are finishing operations. All the constraints are represented by:

$$EB = \{\{13, 16\}, \{18, 25\}\},$$

$$ES = \{\{13, 16, 18, 25, 23, 30\}\},$$

$$\overline{ES} = \{\{2, 3\}, \{2, 5\}, \{2, 7\}, \{2, 9\}, \{2, 11\}, \{2, 14\}, \{2, 17\}, \{2, 24\}\}, \text{ and}$$

$$\overline{EB} = \{\{2, 1\}, \{3, 2\}, \{4, 3\}, \{5, 2\}, \{6, 5\}, \{7, 2\}, \{8, 7\}, \{9, 2\}, \{10, 9\}, \{11, 2\}, \{12, 11\}, \{13, 12\}, \{14, 2\}, \{15, 14\}, \{16, 15\}, \{17, 2\}, \{18, 20\}, \{19, 17\}, \{20, 22\}, \{21, 4\}, \{21, 6\}, \{21, 8\}, \{21, 10\}, \{21, 13\}, \{21, 16\}, \{21, 23\}, \{21, 30\}, \{22, 19\}, \{23, 18\}, \{24, 2\}, \{25, 27\}, \{26, 24\}, \{27, 29\}, \{28, 4\}, \{28, 6\}, \{28, 8\}, \{28, 10\}, \{28, 13\}, \{28, 16\}, \{28, 23\}, \{28, 30\}, \{29, 26\}, \{30, 25\}, \{11, 4\}, \{11, 8\}, \{12, 4\}, \{12, 8\}, \{13, 4\}, \{13, 8\}, \{14, 6\}, \{14, 10\}, \{15, 6\}, \{15, 10\}, \{16, 6\}, \{16, 10\}, \{17, 4\}, \{17, 6\}, \{18, 4\}, \{18, 6\}, \{19, 4\}, \{19, 6\}, \{19, 17\}, \{19, 18\}, \{20, 3\}, \{20, 4\}, \{20, 5\}, \{20, 6\}, \{20, 17\}, \{20, 18\}, \{21, 3\}, \{21, 4\}, \{21, 5\}, \{21, 6\}, \{21, 11\}, \{21, 12\}, \{21, 13\}, \{21, 14\}, \{21, 15\}, \{21, 16\}, \{21, 17\}, \{21, 18\}, \{22, 3\}, \{22, 4\}, \{22, 5\}, \{22, 6\}, \{22, 7\}, \{22, 8\}, \{22, 9\}, \{22, 10\}, \{22, 11\}, \{22, 12\}, \{22, 13\}, \{22, 14\}, \{22, 15\}, \{22, 16\}, \{22, 17\}, \{22, 18\}, \{22, 19\}, \{22, 20\}, \{22, 21\}, \{23, 3\}, \{23, 4\}, \{23, 5\}, \{23, 6\}, \{23, 11\}, \{23, 12\}, \{23, 13\}, \{23, 14\}, \{23, 15\}, \{23, 16\}, \{23, 17\}, \{23, 18\}, \{23, 19\}, \{23, 20\}, \{23, 21\}, \{24, 8\}, \{24, 10\}, \{24, 22\}, \{25, 8\}, \{25, 10\}, \{25, 22\}, \{26, 8\}, \{26, 10\}, \{26, 22\}, \{26, 24\}, \{26, 25\}, \{27, 7\},$$

1
2 {27, 8}, {27, 9}, {27, 10}, {27, 22}, {27, 24}, {27, 25}, {28, 7}, {28, 8}, {28, 9}, {28, 10}, {28,
3
4 11}, {28, 12}, {28, 13}, {28, 14}, {28, 15}, {28, 16}, {28, 22}, {28, 24}, {28, 25}, {29, 3}, {29,
5
6 4}, {29, 5}, {29, 6}, {29, 7}, {29, 8}, {29, 9}, {29, 10}, {29, 11}, {29, 12}, {29, 13}, {29, 14},
7
8 {29, 15}, {29, 16}, {29, 17}, {29, 18}, {29, 19}, {29, 20}, {29, 21}, {29, 22}, {29, 23}, {29, 24},
9
10 {29, 25}, {29, 26}, {29, 27}, {29, 28}, {30, 7}, {30, 8}, {30, 9}, {30, 10}, {30, 11}, {30, 12}, {30,
11
12 13}, {30, 14}, {30, 15}, {30, 16}, {30, 22}, {30, 24}, {30, 25}, {30, 26}, {30, 27}, {30, 28}}.

13
14
15
16 An optimal solution for the assignment of the set \mathbf{N} to blocks and stations (12 blocks and 4
17
18 workstations) is shown in Table 2. In Table 3 an optimal solution is given for $T_0=2.4$, which
19
20 consists of 6 workstations and 12 blocks. This solution remains optimal until $T_0 \geq 2.02$.

21
22
23 [Insert Table 2 about here]

24
25
26 [Insert Table 3 about here]

27 28 29 30 31 32 3. Decomposition procedure

33
34
35 When (2)–(14) is a large-scale problem (*i.e.* more than 70 operations), it is impossible to use
36
37 MIP solvers directly. A possible solution is to use a decomposition technique that divides the initial
38
39 set \mathbf{N} of operations into several (w) disjoint subsets (Dolgui *et al.* 2000, Dolgui *et al.* 2006b) and
40
41 then apply MIP solver in sequential order following the decomposition $\mathbf{N}^1, \mathbf{N}^2, \dots, \mathbf{N}^w$. The global
42
43 result for the initial problem is obtained at the end of the algorithm after w MIP problems have been
44
45 solved.
46
47
48
49

50
51 The subset \mathbf{N}^u is chosen in such a way that all the constraints are not violated. To respect all the
52
53 precedence constraints, all the predecessors of the operations of the subset \mathbf{N}^u treated by MIP have
54
55 to be in this subset or already assigned (during a treatment of an anterior subset):
56
57

$$58 \text{Pred}(j) \subseteq \bigcup_{r=1}^u \mathbf{N}^r \text{ for any } j \in \mathbf{N}^u.$$

59
60

1
2 If an operation of \mathbf{N}^u is also in a subset $e \in ES$ or in $e \in EB$, then all the operations of e have to be
3
4 in \mathbf{N}^u , *i.e.*:

$$e \cap \mathbf{N}^u = \emptyset \text{ or } e \subseteq \mathbf{N}^u \text{ for any } e \in ES \text{ and any } e \in EB.$$

5
6
7
8
9
10
11 An exclusion constraint represented by a subset $e \in \overline{EB}$ (or $e \in \overline{ES}$) is maintained only if all the
12
13 operations of e are in \mathbf{N}^u ; otherwise, the constraint is not taken into account for \mathbf{N}^u .

14
15
16 For the *decomposition*, the partitioning of the set \mathbf{N} is done by a cut of the precedence graph.
17
18 First a desired size of each subset (while analyzing the behavior of the MIP model for this kind of
19
20 problem) is fixed. This choice depends on the initial problem size and on the available time.
21
22

23
24
25 The smaller are the subsets, the easier it is to get a reasonable computation time, but more
26
27 chances exist to reach a solution far from the global optimum.
28
29

30 Two ways are possible to form sub-problems in accordance with the partition $\mathbf{N}^1, \mathbf{N}^2, \dots, \mathbf{N}^w$:

- 31
32
33 1) Form MIP model for each set \mathbf{N}^u separately;
34
35
36 2) Aggregate in MIP model for each set \mathbf{N}^u the solutions obtained by MIP solver for the
37
38 previous problems.
39
40

41 The implementation of the first approach is relatively simple: all the constraints are adjusted
42
43 with regard to the subset \mathbf{N}^u , *i.e.*, a set e is removed from the collections EB, ES, \overline{ES} and \overline{EB} if it
44
45 includes an operation $i \notin \mathbf{N}^u$; operations from $\mathbf{N} \setminus \mathbf{N}^u$ are deleted from each set $Pred(j), j \in \mathbf{N}^u$. The
46
47 solution of the initial problem is obtained then by grouping together all the stations and blocks
48
49 constructed during all the stages.
50
51
52

53
54 For the second approach, the same procedure is applied to the subset $\bigcup_{r=1}^u \mathbf{N}^r$ and the solution
55
56 at the last stage is the solution of the problem (2)-(14). To aggregate the solution of the previous
57
58 problems, additional sets with operations constituting obtained blocks are included into the
59
60 collection EB . In such a situation a reassignment of old blocks to other stations is possible. Another

way of aggregation is to fix intervals $Q(j)$ and $K(j)$, i.e., $Q(j)=[q^*(j), q^*(j)]$ and $K(j)=[\lceil q^*(j)/n_0 \rceil, \lceil q^*(j)/n_0 \rceil]$ where $q^*(j)$ is the optimal assignment of the operation j in the solution and $\lceil s \rceil$ is the smallest integer value greater than or equal to s . In this case, only assignment of operations from \mathbf{N}^u is allowable.

Obviously, if the problem (2)-(14) has a feasible solution then there is a partition $\mathbf{N}^1, \mathbf{N}^2, \dots, \mathbf{N}^w$ such that both approaches can produce an optimal solution. In most cases the first approach allows finding a feasible solution quickly than the second, since in the first case the dimension of solved sub-problems is smaller. On the other hand, the second approach may produce a solution of higher quality.

The idea of the proposed decomposition algorithm is based on the decomposition of (Kilbridge et Wester 1961), but taking into account the specificity of the problem treated and on the objectives of the decomposition. The decomposition procedure uses parameters *MinOp* and *MaxOp* that are the minimum number and the maximum number of operations authorized in a subset (the number of operations must be between *MinOp* and *MaxOp*).

First, each operation range is computed based on the precedence graph. The range of the operations which have no direct predecessor is equal to 1. The range operation i is equal to: $Rang(i) = \max (Rang(j), j \in Pred(i)) + 1$, where $Pred(i)$ is the set of direct predecessors of the operation i . Then, the operations of each range are analyzed one by one beginning from the smaller range and they are added to \mathbf{N}^1 .

For each added operation the inclusion constraints must be respected. If some of them are not respected, the corresponding operations and the non assigned predecessors are added even if their range is different. When all the operations of a range are treated, the following range is studied. As long as the number of operations in \mathbf{N}^1 is less than *MinOp* the procedure is continued. When the decomposition procedure is stopped, the subset \mathbf{N}^1 is reduced taking into account the parameter *MaxOp* and the constraints are modeled for the corresponding MIP model (for \mathbf{N}^1). Then, MIP is

1 applied to this first subset \mathbf{N}^1 , and so on for $\mathbf{N}^2, \dots, \mathbf{N}^w$. To add randomness to the decomposition
 2
 3 procedure, values $MinOpCur$ and $MaxOpCur$ can be used instead $MinOp$ and $MaxOp$ for generation
 4
 5 a subset \mathbf{N}^u , which are randomly chosen within $[MinOp, MaxOp]$.
 6
 7
 8

9
 10 The performance of the MIP model depends on the parameter m_0 and intervals $Q(j)$. They can
 11 be modified in the following way. A lower bound on the objective function of the problem (2)-(14)
 12 is equal to $W_1 \underline{m} + W_2 \underline{n}$, where \underline{m} and \underline{n} are lower bounds on m and $\sum_{k=1}^m n_k$, respectively. The bound \underline{m}
 13
 14 can be calculated as $\max\{k(j) | j \in \mathbf{N}\}$ and the bound \underline{n} can be determined as $\max\{k'(j) | j \in \mathbf{N}\}$
 15
 16 when values $k'(j)$ are obtained in the same manner as $k(j)$ but using $n_0=1$ and without inclusion
 17
 18 constraints ES . If a feasible solution of the problem (2)-(14) is known and UB is the corresponding
 19
 20 value of the objective function, then m_0 can be refined as $m_0 = \lfloor (UB - W_2 \underline{n}) / W_1 \rfloor$, where $\lfloor s \rfloor$ is the
 21
 22 greatest integer value less than or equal to s . As a consequence, intervals $Q(j)$ and $K(j)$ may be
 23
 24 tightened.
 25
 26
 27
 28
 29
 30
 31
 32

33 The decomposition procedure can be applied several times and the best solution is kept.

34 In this algorithm, the FSIC heuristic (Dolgui *et al.* 2005) based on a generalization of the
 35
 36 COMSOAL method is used to search a feasible initial solution.
 37
 38
 39
 40

41 The following notations are used:

- 42
 43
 44 TR_{tot} the current number of trials;
 45
 46
 47 TR_{nimp} the number of trials that do not improve the current best solution;
 48
 49
 50 C_{min} the cost of the best obtained solution;
 51
 52
 53 C_{cur} the cost of the current solution;
 54
 55
 56 T_{heur} the total available time for solving a sub-problem by a heuristics;
 57
 58
 59 T_{MIP} the total available time for solving a sub-problem by MIP solver;

60 **Algorithm:**

Step1. Set $C_{min} = \infty$, $TR_{tot} = 0$, $TR_{nimp} = 0$.

1
2 Step 2. Set $w=1$.

3
4 Step 3. Generate subset N^w and form problem (2)-(14) for the subset N^w in accordance with the
5
6 parameters $MinOp$ and $MaxOp$.

7
8
9 Step 4. Run the FSIC heuristics with the available time T_{heur} . If a feasible solution is found then
10
11 modify m_0 , $Q(j)$ and $K(j)$.

12
13 Step 5. Run MIP model. If MIP does not give any solution after T_{MIP} then MIP is stopped. The
14
15 current MIP solution is compared with the heuristic solution and the best is kept.

16
17 Step 6. If there are still some non assigned operations then set $w=w+1$ and go to Step 3.

18
19 Step 7. If $C_{cur} < C_{min}$ then set $C_{min} = C_{cur}$, $TR_{nimp} = 0$. Otherwise set $TR_{nimp} = TR_{nimp} + 1$.

20
21 Step 8. Set $TR_{tot} = TR_{tot} + 1$.

22
23 Step 9. Stop if one of the following conditions holds:

24
25 a given solution time is exceeded;

26
27 TR_{tot} is greater than the maximum number of iterations authorized;

28
29 TR_{nimp} is greater than a given value;

30
31 C_{min} is lower than a given cost value.

32
33 Otherwise go to Step 3.

34 35 36 37 38 39 40 41 42 **4. Tests and comparisons**

43
44 In order to study the performance of the decomposition procedures described above, 8 data sets
45
46 were created, each of which contains 10 machining line balancing problems (i.e. 80 problems were
47
48 tested). The data sets differ by two parameters: the number of operations $|N|$ and the order strength
49
50 OS of the precedence graph.

51
52 Note: the order strength OS is defined as the density of the transitive closure of the precedence
53
54 graph (Scholl and Klein, 1998).

55
56 Each unique problem was generated at random. The operation times are derived from the
57
58 uniform distribution ($a=10$, $b=20$). The precedence graph is generated at random by adding arcs
59
60 between randomly chosen vertexes until getting desired order strength. All data sets were solved by

decomposition algorithms with the following parameters:

- 1) IS1, independent solving with $MinOp = 10$ and $MaxOp = 20$;
- 2) IS2, independent solving with $MinOp = 15$ and $MaxOp = 25$;
- 3) AS1, aggregate solving with $MinOp = 10$ and $MaxOp = 20$;
- 4) AS2, aggregate solving with $MinOp = 15$ and $MaxOp = 25$.

Experiments were carried out on Pentium IV, 3GHZ. The obtained results are reported in Table

4. For the presentation, the following notations are used: NO – number of obtained optimums;

Δ_{max} , Δ_{av} , Δ_{min} – maximal, average and minimal deviation of the criterion obtained by a method from the best value, respectively; T_{com} – computational time in seconds.

[Insert Table 4 about here]

For the problems with 200 operations, IS method found only some solutions (< 3). So, only the results obtained by AS method are reported in Table 4 for this data set. For all data sets, the decomposition method with independent solving was surpassed by the method with aggregate solving. In the same time, it is impossible to conclude that one of the sets of parameters for aggregate decomposition (AS1 or AS2) is better from the quality of the obtained results. The experiments show that the method performance depends on the characteristics of the problem.

6. Conclusion

A line balancing problem in machining environment has been discussed. It concerns the machining lines with multi-spindle heads. Each multi-spindle head executes simultaneously a block of operations. In comparison with the standard assembly line balancing problems, the problem addressed has many additional properties and constraints. In this paper, an industrial example of this problem is given.

For small problem of this type, there are in literature exact optimization models based on Mixed Integer Programming (MIP) and graph theory, but these models are time consuming. Therefore, the generation of “good” design decision for large-scale machining lines (one or several hundred of operations) needs the development of efficient heuristic algorithms.

1
2 In this paper, an approach for solving this problem has been tested. This approach is based on
3
4 an improved MIP model and a new heuristic of its decomposition. Using this algorithm, the initial
5
6 problem is decomposed into several sub-problems and solving them by means ILOG Cplex solver.
7
8 Two algorithms of decomposition are tested for randomly generated series of examples from 50
9
10 operations to 200 operations. The results are promising.
11
12

13
14 Future research will concern development of heuristics to find “good” partitions in order to
15
16 decrease the calculation time. A possible way is the use of the COMSOAL based heuristics as FSIC
17
18 or RAP for a not deterministic decomposition. Another direction is the development of a pre-
19
20 processing procedure for the aggregation of operations into macro-operations before optimization.
21
22
23

24 25 26 **Acknowledgements**

27
28 This work is partially supported by ISTC project B-986, INTAS project 03-51-5501 and European
29
30 coordination action CODESNET.
31
32

33 34 **References**

- 35
36
37 ARCUS, A.L., 1966. COMSOAL: A computer method of sequencing operations for assembly lines.
38
39 *International Journal of Production Research*, **4**, 259-277.
40
41
42 ASKIN, R.G. and STANDRIDGE, C.R., 1993. *Modeling and analysis of manufacturing systems* (John
43
44 Wiley & Sons, Inc)
45
46
47 BAYBARS, I. 1986. A survey of exact algorithms for the simple assembly line balancing.
48
49 *Management Science*, **32**, 909-932.
50
51
52 DASHCHENKO, A. I. (Ed), 2003. *Manufacturing Technologies for Machines of the Future 21st*
53
54 *Century Technologies*, Springer.
55
56
57 DOLGUI, A., GUSCHINSKY, N. and LEVIN, G., 1999. On Problem of Optimal Design of Transfer
58
59 Lines with Parallel and Sequential Operations. In: *Proceedings of the 7th IEEE International*
60
Conference on Emerging Technologies and Factory Automation (ETFA'99), October 18-21,
1999, Barcelona, Spain. J.M. Fuertes (Ed.), Vol. 1, pp. 329-334.

- 1
2 DOLGUI, A., GUSCHINSKY, N. and LEVIN, G., 2000. *Approaches to balancing of transfer line with*
3
4 *block of parallel operations*, Institute of Engineering Cybernetics/University of Technology
5
6 of Troyes, Minsk, Preprint No. 8, 42 pages.
7
8
9 DOLGUI, A., GUSCHINSKY, N. and LEVIN, G. 2006a. A Special case of transfer lines balancing by
10
11 graph approach, *European Journal of Operational Research*, **168**, 732-746, Available online
12
13 11 September 2004.
14
15
16 DOLGUI, A., FINEL, B., GUSCHINSKY, N., LEVIN, G. and VERNADAT, F., 2006b. MIP approach to
17
18 balancing transfer lines with blocks of parallel operations, *IIE Transactions*, (accepted, in
19
20 Press).
21
22
23 DOLGUI, A., FINEL, B., GUSCHINSKY, N., LEVIN, G., and VERNADAT, F., 2005. A Heuristic
24
25 Approach for Transfer Lines Balancing, *Journal of Intelligent Manufacturing*, **16**, 159-172.
26
27
28 EREL, E., and SARIN, S.C., 1998. A survey of the assembly line balancing procedures. *Production*
29
30 *Planning and Control*, **9**, 414-434.
31
32
33 GADINOV, R., WILHELM, W., 2000. A cutting plane approach for single-product assembly system
34
35 design problem, *International Journal of Production Research*, **38**, 1731-1754.
36
37
38 GHOSH, S., and GAGNON, R., 1989. A comprehensive literature review and analysis of the design,
39
40 balancing and scheduling of assembly lines. *International Journal Production Research*, **27**,
41
42 637-670.
43
44
45 GROOVER, M. P., 1987. *Automation, Production Systems and Computer Integrated Manufacturing*,
46
47 (Prentice Hall, Englewood Cliffs, New Jersey)
48
49
50 HITOMI, K., 1996. *Manufacturing Systems Engineering*, (Taylor & Francis)
51
52
53 KILBRIDGE, M.D. and WESTER, L., 1961. A heuristic method of assembly line balancing, *Journal of*
54
55 *Industrial Engineering*, **12**, 292-298.
56
57
58 REKIEK, B., DOLGUI, A., DELCHAMBRE, A., and BRATCU, A. 2002. State of the art of assembly lines
59
60 design optimisation. *Annual Reviews in Control*, **26**, 163-174.

- 1
2 SCHOLL, A. and KLEIN, R., 1998. Balancing assembly lines effectively: a computational
3
4 comparison. *European Journal of Operational Research*, **114**, 51-60.
5
6
7 SPRECHER, A., 1999. A competitive branch-and-bound algorithm for the simple assembly line
8
9 balancing problem. *International Journal of Production Research*, **37**, 1787-1816.
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

For Peer Review Only

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

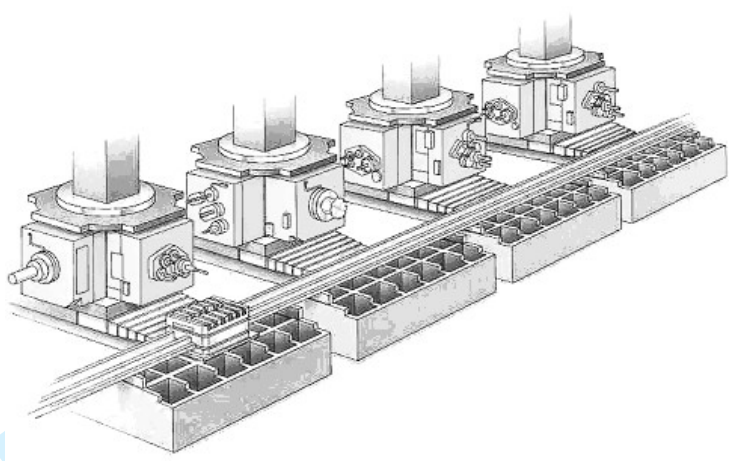


Figure 1. A machining line with multi-spindle heads

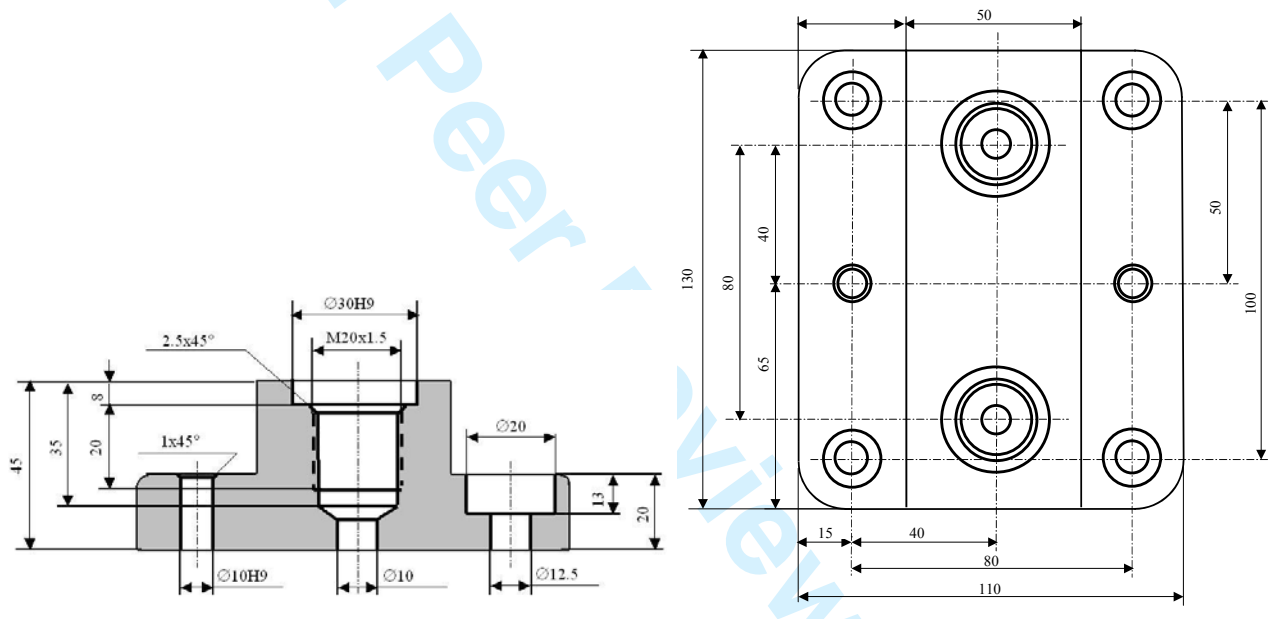


Figure 2. Part machined

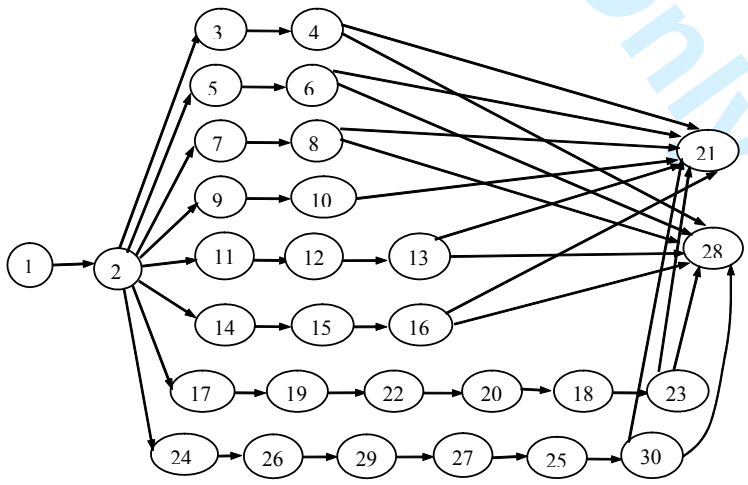


Figure 3. Precedence graph

Table 1. Operations and their parameters

Machined elements	Operation name	Operation number	Working stroke length, mm	Feed rate, mm/min	Execution time, min
Plane P	Mill roughly	1	155	167.5	0.93
	Mill completely	2	155	146.5	1.06
Hole H1	Drill Ø12.5	3	35.8	52.6	0.68
	Spotface Ø20	4	20	127.6	0.16
Hole H2	Drill Ø12.5	5	35.8	52.6	0.68
	Spotface Ø20	6	20	127.6	0.16
Hole H3	Drill Ø12.5	7	35.8	52.6	0.68
	Spotface Ø20	8	20	127.6	0.16
Hole H4	Drill Ø12.5	9	35.8	52.6	0.68
	Spotface Ø20	10	20	127.6	0.16
Hole H5	Drill Ø9.8	11	29.9	57.4	0.52
	Drill out facet Ø12	12	8	68.4	0.12
	Ream a hole Ø10	13	24.1	139.3	0.17
	Drill Ø9.8	14	29.9	57.4	0.52
Hole H6	Drill out facet Ø12	15	8	68.4	0.12
	Ream a hole Ø10	16	24.1	139.3	0.17
	Drill Ø10	17	57.7	57.4	1
Hole H7	Enlarge a hole Ø18	18	53.3	68.4	0.78
	Drill out Ø17	19	44.1	68.4	0.64
	Drill out facet Ø22	20	6	68.4	0.09
	Tap a hole	21	24	143.6	0.17
	Bore out roughly Ø29.5	22	24	259.5	0.09
	Bore out completely Ø30	23	12.5	106.3	0.12
	Drill Ø10	24	57.7	57.4	1
Hole H8	Enlarge a hole Ø18	25	53.3	68.4	0.78
	Drill out Ø17	26	44.1	68.4	0.64
	Drill out facet Ø22	27	6	68.4	0.09
	Tap a hole	28	24	143.6	0.17
	Bore out roughly Ø29.5	29	24	259.5	0.09
	Bore out completely Ø30	30	12.5	106.3	0.12

Table 2. An optimal solution for $T_0=2.8$

Station number k	1		2		3		4					
Block number l	1	2	1	2	1	2	3	4	1	2	3	4
Operations $i \in N_{kl}$	1	2	17,24	19,26	22	29	11,14, 3,5,7,9	4,6, 8,10	20,27, 12,15	18,25,13, 16	23,30	21,28
$L(N_{kl})$	155	155	57.7	44.1	24	24	35.8	20	8	53.3	12.5	24
$S(N_{kl})$	167.5	146.5	57.4	68.4	259.5	259.5	52.6	127.6	68.4	68.4	106.3	143.6
Block time $t^b(N_{kl})$	1.13	1.26	1.21	0.84	0.29	0.29	0.88	0.36	0.32	0.98	0.32	0.38
Station time $t^s(N_k)$	2.79		2.45		2.22		2.40					

Table 3. An optimal solution for $T_0=2.4$

Station number k	1	2	3	4		5			6			
Block number l	1	1	1	1	2	3	1	2	3	1	2	3
Operations $i \in N_{kl}$	1	2	17,24	19,26,11, 14	22	29	20,27, 12,15	18,25,13,16	23,30	3,5,7,9	4,6,8,10	21,28
$L(N_{kl})$	155	155	57.7	44.1	24	24	8	53.3	12.5	35.8	20	24
$S(N_{kl})$	167.5	146.5	57.4	57.4	259.5	259.5	68.4	68.4	106.3	52.6	127.6	143.6
Block time $t^b(N_{kl})$	1.13	1.26	1.21	0.97	0.29	0.29	0.32	0.98	0.32	0.88	0.36	0.38
Station time $t^s(N_k)$	1.53	1.66	1.61	1.96		2.02			2.02			

Table 4. Results of tests

Method	$ N , OS, T_{com}$	NO	Δ_{min} %	Δ_{av} %	Δ_{max} %	$ N , OS, T_{com}$	NO	Δ_{min} %	Δ_{av} %	Δ_{max} %
IS1	50, 25, 60	0	2.6	23.1	35.3	50, 45, 60	1	0	22.2	72.7
IS2		0	7.5	20.7	35.3		1	0	16.1	36.4
AS1		6	0	2.5	12.8		6	0	1.2	4.8
AS2		7	0	1.5	9.5		5	0	3.7	17.1
IS1	75, 25, 120	0	3	31.2	49.1	75, 45, 120	0	7.6	22.3	49
IS2		0	8	24.7	42.1		0	1.3	19.3	47.1
AS1		7	0	2.6	11.4		6	0	2	8.2
AS2		5	0	2.6	10.5		10	0	0	0
IS1	100, 25, 300	0	2.6	22.3	35.3	100, 45, 300	1	0	22.2	72.7
IS2		0	5.1	18.7	35.3		1	0	16.9	36.4
AS1		8	0	0.5	2.5		6	0	1.3	4.9
AS2		10	0	0	0		6	0	1	2.6
IS1	150, 45, 600	0	11.9	19.9	35	200, 45, 600	-	-	-	-
IS2		0	6.25	15.7	28.5		-	-	-	-
AS1		4	0	2.5	9.4		5	0	1.7	4.5
AS2		7	0	1.6	9		6	0	2.1	10.3