ELSEVIER

Int. J. Production Economics 68 (2000) 1–14

international journal of production economics

www.elsevier.com/locate/dsw

Heuristic methods for cost-oriented assembly line balancing: A survey

Matthias Amen*

Institut für Unternehmensrechnung und Controlling, University of Berne, Engehaldenstrasse 4, CH-3012 Bern, Switzerland

Received 17 March 1998; accepted 21 September 1998

Abstract

This paper is concerned with cost-oriented assembly line balancing. This problem occurs especially in the final assembly of automotives, consumer durables or personal computers, where production is still very labour-intensive, and where the wage rates depend on the requirements and qualifications to fulfil the work. First a short problem description is presented. After that a classification of existent and new heuristic methods for solving this problem is given. The heuristic methods presented in this paper are described in detail. A new priority rule called "*best change of idle cost*" is proposed. This priority rule differs from the existent priority rules because it is the only one which considers that production cost are the result of both, production time and cost rates. Furthermore a new sophisticated method called "*exact solution of sliding problem windows*" is presented. The solution process is illustrated by an example, showing how this metaheuristic works together with an exact method. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Assembly line balancing; Cost-oriented production planning; Heuristic methods

1. Introduction

In an *assembly line* the product units move with a constant transportation speed through the consecutive stations. The total work content to be performed by the production system has been split up into economical indivisible work elements which are called *tasks*. Among the set of tasks there exist *technological precedence relations*. The set of tasks to be performed in the same station is called an *operation* or a *station load*. The time to perform an operation is restricted by the *cycle time*. The *assembly line balancing problem* consists in allocating the tasks to the stations subject to the technological precedence relations, the cycle time restriction of the stations and the indivisibility of the tasks.

In literature the objective usually is to minimize the number of stations in a line for a fixed cycle time [1, p. 650]. In other words: The objective is to minimize the total idle time of the total capacity provided by the sum of the stations of the line [2, p. 911]. Therefore, this is called *time-oriented assembly line balancing*. The objective in *cost-oriented assembly line balancing* which is considered in this paper is *to minimize the total cost per product unit* [3–6].

In general, assembly is a labour-intensive kind of production. Therefore, in cost-oriented assembly

* Tel.: + 41-31-631-85-01; fax: + 41-31-631-37-80.
E-mail address: matthias.amen@iuc.unibe.ch (M. Amen)

line balancing, the *labour cost* has to be analyzed in detail: The wage rate of each operation depends on the maximal point value of the tasks assigned to a station. As the tasks differ in their difficulty, there are differences in the point values and thus in the corresponding wage rates. The worker has to be paid for the whole cycle time irrespective of the duration of the operation. So the total labour cost per product unit is the sum of the wage rates of all stations multiplied by the cycle time. Further it is assumed that the *cost of capital* depends on the overall line length and that all stations have the same length. All other costs (e.g. material cost) can be assumed to be independent of the division of labour and of the line length [3, p. 82]. Introducing the following symbols:

I	number of tasks ($-$)
i	index for the tasks, $i = 1(1)I$
M	number of stations ($-$)
m	index for the stations, $m = 1(1)M$
I^{s}_m	set of tasks, assigned to station m, $m = 1(1)M$
c	cycle time (TU/PU)
k	total cost per unit (MU/PU)
k^{s}_m	cost rate of station m, $m = 1(1)M$ (MU/TU)
k^{sr}	cost of capital per station (MU/PU)
k^{sw}_m	wage rate of station m, $m = 1(1)M$ (MU/TU)
k^{t}_i	cost rate of task i, $i = 1(1)I$ (MU/TU)
k^{tw}_i	wage rate of task i, $i = 1(1)I$ (MU/TU)

(Note: Dimensions: TU, time units; PU, product units; MU, monetary units; Labels: r, cost of capital; s, station; t, task; w, wage rate.)

The *total cost per product unit* can be calculated as

$$k = \left(\sum_{m=1}^{M} c k^{\mathrm{sw}}_m \right) + M k^{\mathrm{sr}}$$

with $k^{\mathrm{sw}}_m = \max\{k^{\mathrm{tw}}_i | i \in I^s_m\}$.

Since the conveyor speed is fixed by the cycle time, the cost of capital for a single station can be transformed into a cost rate per time unit. The total cost of capital can be formulated as $Mk^{\mathrm{sr}} = cM(k^{\mathrm{sr}}/c)$. Thus for the purpose of capacity balancing the relevant cost rates per time unit are made up by the sum of the capital cost per time unit and the task-depended wage rates: $k^{\mathrm{t}}_i = k^{\mathrm{tw}}_i + k^{\mathrm{sc}}/c$ and $k^{\mathrm{s}}_m = k^{\mathrm{sw}}_m + k^{\mathrm{sc}}/c$. This simplifies the calculation of the total cost per product unit to

$$k = \sum_{m=1}^{M} c k^{\mathrm{s}}_m = \sum_{m=1}^{M} c \max\{k^{\mathrm{t}}_i | i \in I^{\mathrm{s}}_m\}.$$

The total cost per product unit $k = \sum_{m=1}^{M} c k^{\mathrm{s}}_m$ differ from the term $\sum_{i=1}^{I} d^{\mathrm{t}}_i k^{\mathrm{t}}_i$ only by the idle cost per product unit which are caused by idle time and/or cost rate differences of the tasks assigned to the same station. Therefore, the objective to minimize the total cost per product unit is equivalent to minimize the total idle cost per product unit.

2. Classification of heuristic methods for solving the cost-oriented assembly line balancing problem

As the *NP*-complete bin-packing problem can be reduced to the time-oriented assembly line balancing problem in polynomial time, the time-oriented assembly line balancing problem is *NP*-hard [7, p. 56]. This also holds for the cost-oriented problem [4, p. 479]. Thus it is justified to develop heuristic solution methods.

There exist two different general approaches for assembly line balancing. In *the station-oriented approach* in each step of the solution process only one station is considered. Therefore we have to look for tasks which have to be assigned to the current station. In *the task-oriented approach* in each step of the solution process a selected task is considered. Therefore we have to look for the station to which the current task has to be assigned to [8, pp. 917–918; 9, pp. 182–184]. All of the methods described in this paper use the station-oriented approach. Fig. 1 gives an overview of the existent and new methods which can be classified as follows:

Most of the methods can be characterized as *random choice/priority rule methods*. In each step of the solution process all of the methods of this category but one choose one of alternative tasks for assignment. For determining the task either random choice or priority rules are used. The priority rule methods can be distinguished into methods that make use of one problem-oriented priority rule and methods that use several problem-oriented priority rules. Priority rules are called *problem-oriented* if they consider structural aspects of the

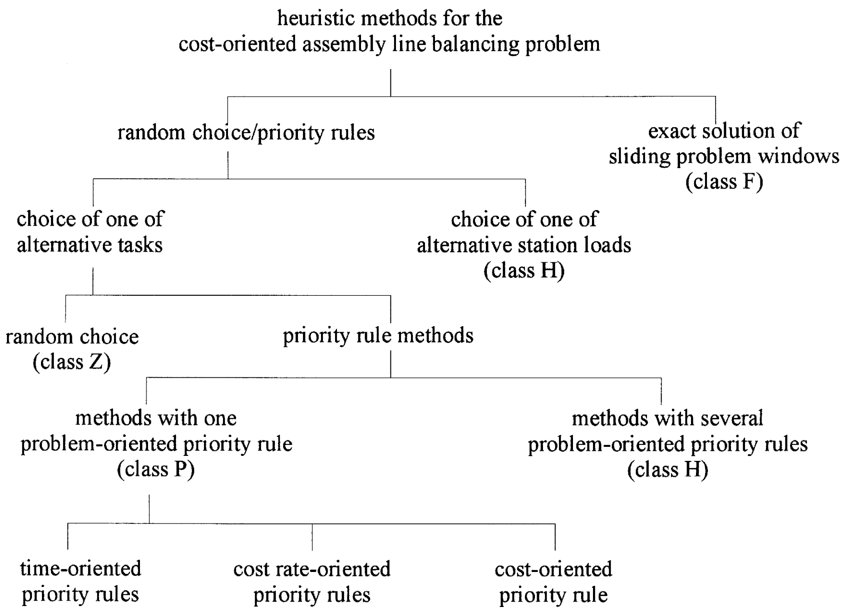
heuristic methods for the cost-oriented assembly line balancing problem
random choice/priority rules
exact solution of sliding problem windows (class F)
choice of one of alternative tasks
choice of one of alternative station loads (class H)
random choice (class Z)
priority rule methods
methods with one problem-oriented priority rule (class P)
methods with several problem-oriented priority rules (class H)
time-oriented priority rules
cost rate-oriented priority rules
cost-oriented priority rule

Fig. 1. Heuristic methods for cost-oriented assembly line balancing.

problem. The set of problem-oriented priority rules consists of three subclasses. *The time-oriented priority rules* consider only the durations and/or the precedence relations of the tasks, the *cost rate-oriented priority rules* consider only the cost rates of the tasks. Due to the fact that cost are the product of both, cost-rate and duration, the *new cost-oriented priority rule "best change of idle cost"* which will be presented in the next section takes both aspects into account. In methods with several problem-oriented priority rules the single rules are used in lexicographic order, i.e. problem-oriented priority rules are used as tie-breakers. All priority rule methods use a simple non-problem-oriented priority rule as a final tie-breaker.

One priority method differs from the other as in the major steps of the solution process one of several alternative station loads has to be chosen by the use of different priority rules which consider the situation in the alternative station loads. The alternative station loads are generated by the use of several problem-oriented priority rules to choose one of alternative tasks. Thus this method is a further development of the previous mentioned methods.

The new method called *"exact solution of sliding problem windows"* is quite different from the heuristic methods mentioned before because it is a heuristic version of an exact method. With respect to the similarities of the heuristic methods, the methods are grouped into the following classes:

- Z methods with random choice task assignment;
- P methods with one problem-oriented priority rule;
- H methods with several problem-oriented priority rules;
- F methods with exact solution of sliding problem windows;
- E exact method.

Class H contains both, the simple methods which choose one of alternative tasks and the more advanced method which makes use of these simple methods when generating alternative station loads. In the next section the methods of classes Z and P and the simple methods of class H are presented. Section 4 is concerned with the more advanced method of class H. Section 5 deals with the methods of class F. Class E consists of only one method. As exact methods are beyond the topic of this paper

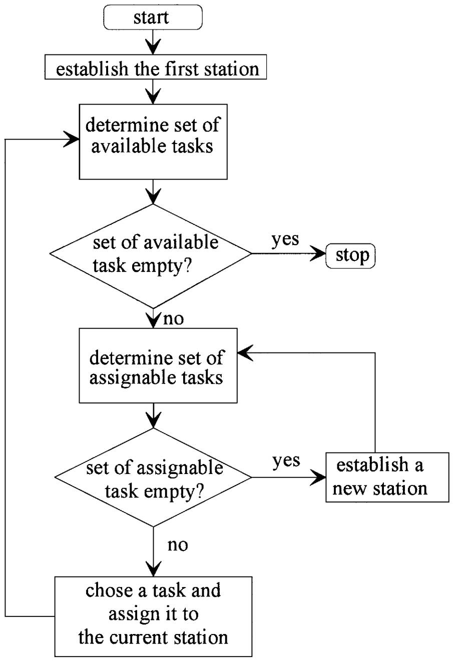
start
establish the first station
determine set of available tasks
set of available task empty?
yes
stop
no
determine set of assignable tasks
set of assignable task empty?
yes
establish a new station
no
chose a task and assign it to the current station

Fig. 2. Procedure for assigning tasks to stations.

this method is not presented in this paper (for further reading see [5,6]. The exact method works as a subalgorithm for the methods of class F.

3. Methods with choice of one of alternative tasks

All methods with choice of one of alternative tasks in each step of the solution process follow the same procedure when assigning tasks to stations. The procedure is shown in Fig. 2.

As the station loads are generated successively, in each step of the procedure only one station is considered (*station-oriented approach*). The tasks which have no predecessor task or only predecessor tasks already assigned to the current or to an earlier station are called *available*. The available tasks which have a duration not longer than the remaining idle time of the considered station are called *assignable*. Among the set of assignable tasks one task is chosen according to the method used. After assigning this task the sets of available and assignable tasks are updated immediately. If the set of assignable tasks is empty, the current station is *"maximally loaded"* [10, p. 266; 8, p. 918]. Then a new station is established and the set of assignable tasks has to be updated again. Each method stops after all tasks have been assigned to a station.

The *random choice method* (class Z) [11, pp. 25–35 and 44–46] can be used several times for solving the same problem instance. Among the set of randomly generated solutions for the same problem instance only the best one has to be considered for realization.

For a formal description of the priority rule methods, the following additional symbols are introduced:

I^{assign}	set of assignable tasks
d_i^{t}	duration of task i, $i = 1(1)I$ (TU/PU)
F_i	set of all technological immediate following tasks of task i, $i = 1(1)I$
Δk_i	change of idle cost if task i will be assigned to the currently considered station, $i \in I^{\text{assign}}$ (MU/PU)
r_i	ranked positional weight of task i, $i = 1(1)I$ (TU/PU)

With this symbols the following *problem-oriented priority rules* of class *P* can be defined:

- P-MaxD Maximal task duration [12, p. 728], i.e. $\max\{d_i^{\text{t}} | i \in I^{\text{assign}}\}$.
- P-MaxR Maximal ranked positional weight ([13, p. 395 in the modified version of Hahn], [14, pp. 44–46]), i.e. $\max\{r_i | i \in I^{\text{assign}}\}$, where

$$r_i = \begin{cases} d_i^{\text{t}} + \sum_{j \in F_i} r_j, & \text{if } F_i \neq \emptyset, \\ d_i^{\text{t}}, & \text{if } F_i = \emptyset. \end{cases}$$

- P-MaxF Maximal number of immediate followers [12, p. 728], i.e. $\max\{|F_i| \mid i \in I^{\text{assign}}\}$.
- P-MaxKt Maximal cost rate [4, p. 481], i.e. $\max\{k_i^{\text{t}} | i \in I^{\text{assign}}\}$.
- P-MinKt Minimal cost rate [3, pp. 106–107], i.e. $\min\{k_i^{\text{t}} | i \in I^{\text{assign}}\}$.
- P-MinKts Minimal absolute difference to the current station cost rate [3, pp. 106–107], i.e. $\min\{|k_i^{\text{t}} - k_m^{\text{s}}| \mid i \in I^{\text{assign}}\}$.

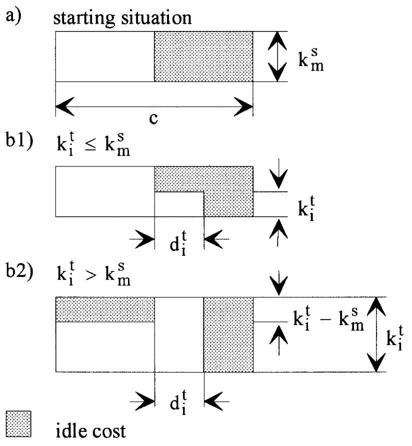

Fig. 3. Calculation of Δk_i.

- P-MinKI Best change of idle cost, i.e. $\min\{\Delta k_i | i \in I^{\text{assign}}\}$, where

$$\Delta k_i = \begin{cases} -d_i^t k_i^t, & \text{if } k_i^t \leqslant k_m^s, \\ (k_i^t - k_m^s)c - d_i^t k_i^t, & \text{if } k_i^t > k_m^s. \end{cases}$$

The non-problem-oriented priority rule

- P-MinI Minimal task number [12, p. 729], i.e. min $\{i | i \in I^{\text{assign}}\}$ (Usually the tasks are numbered topologically according to the precedence relations.)

is used as final tie breaker in all priority rule methods. While applying the new priority rule "*best change of idle cost*" (P-MinKI) both decreases and increases can occur, which has to be accepted. Not choosing an assignable task with $\Delta k_i > 0$ and establishing a new station in some cases may cause an extra station and therefore higher overall idle cost. The calculation of Δk_i is illustrated in Fig. 3.

P-MinKts and P-MinKI are dynamic rules because they depend on the current cost rate of the station. For these rules it is necessary to calculate not only the remaining idle time of the current station after each assignment, but also to update the cost rate of the station immediately. While P-MaxD, P-MaxR and P-MaxF are only time-oriented and P-MaxKt, P-MinKt and P-MinKts depend only on the cost rates, P-MinKI is the only priority rule which considers that cost results from both, the cost rates and the time needed by the workers of the line.

Within class H there exist three priority rule methods with the choice of one of alternative tasks in each step of the procedure. The lexicographic orders of priority rules of these heuristic methods are given below. For the heuristic methods of Steffen and Heizmann the lexicographic order differs according to the fact whether the station is just established (empty station) or the station has been established at an earlier step in the solution process (partially filled station).

- H-Stef Heuristic method of Steffen [3, pp. 105–110]:

Lexicographic order of priority rules in case of just established stations (station is empty):	Lexicographic order of priority rules in case of already established stations (station is partially filled with tasks):
1. P-MaxR	1. P-MinKts
2. P-MinI	2. P-MinKt
	3. P-MaxR
	4. P-MinI

- H-Heiz Heuristic method of Heizmann [15, pp. 110–124]:

Lexicographic order of priority rules in case of just established stations (station is empty):	Lexicographic order of priority rules in case of already established stations (station is partially filled with tasks):
1. P-MaxD	1. Identity of task cost rate and current station cost rate (i.e. $\lvert k_i^t - k_m^s \rvert = 0$ is required).
2. P-MinI	2. P-MaxD
	3. P-MinI

If there is no task which has the same cost rate as the current station cost rate in the case of an already established station (station is partially filled with tasks), then the priority rule P-MaxD is used together with the tie-breaking rule P-MinI.

- H-WR Heuristic method "wage rate" of Rosenberg/Ziegler [4, p. 481]:
  1. P-MaxKt
  2. P-MaxD
  3. P-MinI

  This heuristic method is a modification of the well-known "first-fit decreasing height"-method suggested for solving two-dimensional packing problems [16, p. 810]. The method is called "wage rate" because Rosenberg/Ziegler [4] consider only labour cost.

4. A method with choice of one of alternative station loads

A priority-rule method with choice of one of alternative station loads is the "wage rate smoothing" method (H-WRS) [4, pp. 481–484]. The method consists of a three-phase algorithm. In order to give a formal description of this method the following symbols are introduced:

I^{rest}	set of not yet finally assigned tasks
$I_m^{\text{s,pot,phase2}}$	set of tasks which defines the potential station load of station m in phase 2, $m = 1(1)M$
$I_m^{\text{s,pot,phase3}}$	set of tasks which defines the potential station load of station m in phase 3, $m = 1(1)M$
W	number of different cost rates ($-$)
w	index of the cost rate, $w \in \{1, \dots, W\}$
$I_w^{\text{assign},\ell}$	set of assignable tasks with the cost rate k_w^ℓ, $w = 1(1)W$
U_w	number of potential station loads for the currently considered task cost rate k_w^ℓ ($-$), $w = 1(1)W$
u	index of the potential station loads, $u \in \{1, \dots, U_w\}$
$I_{m,w,u}^{\text{s,k,pot}}$	set of tasks which defines the uth potential station load for the considered cost rate k_w^ℓ of the station m in the phases 1 and 3, $u = 1(1)U_w$, $w = 1(1)W$, $m = 1(1)M$
$I_{m,w,u}^{\text{s,k,pot,2}}$	set of tasks which defines the uth potential station load for the considered cost rate k_w^ℓ of the station m in the phase 2, $u = 1(1)U_w$, $w = 1(1)W$, $m = 1(1)M$
$I_{m,w,u}^{\text{assign,pot}}$	set of tasks with task cost rate k_w^ℓ that are assignable to station m if the current station load is defined by the set of tasks $I_{m,w,u}^{\text{s,k,pot}}$, $u = 1(1)U_w$, $w = 1(1)W$, $m = 1(1)M$
$d_{m,w,u}^{\text{s,pot}}$	work content of the current station load $I_{m,w,u}^{\text{s,k,pot}}$ (TU/PU), $u = 1(1)U_w$, $w = 1(1)W$, $m = 1(1)M$
$k_{m,w,u}^{\text{s,pot}}$	cost rate of the potential station load $I_{m,w,u}^{\text{s,k,pot}}$ (MU/TU), $u = 1(1)U_w$, $w = 1(1)W$, $m = 1(1)M$
fixed	variable which indicates in phase 1 whether phase 1 has to be restarted or whether phase 2 has to be started ($-$)

Fig. 4 shows the major steps of the 3-phase algorithm. The formal descriptions of the phases 1–3 are given in the appendix.

In phase 1 of the procedure, potential station loads are established. In phases 2 and 3 the potential station loads which have been calculated in phase 1 are taken and further tasks are assigned. In phase 1 each potential station load consists only of tasks which have identical cost rates. The phases 2 and 3 are completely separated from each other. In phase 2 a potential station load taken from phase 1 is filled further only with tasks which have the same cost rate as the station cost rate. In phase 3 a potential station load taken from phase 1 is filled further with tasks which could have cost rates different from the station cost rate of phase 1. In step 5 a station load has to be chosen from the two alternative station loads which have been obtained from the phases 2 and 3. After a potential station load is finally fixed, in step 5 the algorithm moves back to phase 1. Within phase 1 stopping-criteria are to be checked at two different steps.

5. A method with exact solution of sliding problem windows

A new heuristic method which is quite different from those described in the foregoing sections is the "exact solution of sliding problem windows" (class F). This heuristic is an application of the "*working*

1	$m := 1$, $I^{rest} := I$.
2	Phase 1 (appendix figure 8).
3	Phase 2 (appendix figure 9).
4	Phase 3 (appendix figure 10).
5	Choose the station load I^{s}_{m} from $I^{s,pot,phase2}_{m}$ and $I^{s,pot,phase3}_{m}$ by use of the following lexicographic order of priority rules: 1. maximal work content of station m 2. maximal number of tasks in station m 3. first considered potential station load (i.e. $I^{s,pot,phase2}_{m}$)
6	$I^{rest} := I^{rest} - I^{s}_{m}$.
7	$m := m+1$, go to 2.

Fig. 4. Wage rate smoothing.

forward technique" [17, pp. 315–316]. The key concept is to solve consecutive small problems by the use of an exact method. A part of the optimal solution of a small problem will be fixed finally. Then the next small problem will be defined and solved optimally. This process continues until the last small problem is solved. Similar methods for solving different assembly line balancing problems were suggested in [18–20].

The idea to develop a heuristic application of an exact method derives from the fact that all existent heuristic methods load the stations maximally, but that *the cost-oriented optimal solution could be missed* if the stations are filled in this way [5, p. 225; 6, 1998a]. The exact method used is the only existent method in which the stations are not necessarily loaded maximally.

5.1. Formal description of the method

For a formal description of the "exact solution of sliding problem windows" the following additional symbols are introduced:

I^{pw}	maximal number of tasks in a problem window (–)
I^{part}	set of tasks which are in a problem window
M^{kum}	number of the currently finally established stations (–)
M^{part}	number of stations which are needed in the solution of a problem window (–)
M^{fix}	number of stations which are needed in the solution of a problem window and which are finally established (–)
PartM	proportion of the maximal number of finally established stations to the number of stations which are needed in the solution of a problem window (–).

An important assumption for the formal description which is given in Fig. 5 is the *topological numbering* of the tasks, i.e. if $i \in F_h$ then $h < i$ holds. If the tasks are not numbered topologically then they have to be renumbered first.

The small problems which have to be solved by the use of an exact method are called "problem windows". A *problem window* is defined by the first I^{PW} lowest numbered tasks of the set of the not yet assigned tasks I^{rest}. If $|I^{rest}| \leqslant I^{PW}$ holds then the current problem window is the last one. If $|I^{rest}| < I^{PW}$ holds then the last problem window contains only $|I^{rest}|$ tasks. As the tasks are numbered topologically the precedence restrictions among the set of tasks are not violated by this simple approach for generating problem windows. From the M^{part} stations generated in a solution of a problem window the first M^{fix} stations with their corresponding station loads are finally established. The number M^{fix} is calculated by multiplying the M^{part} stations by *PartM*. Because of the indivisibility of stations we have to round the result (see step 3) ($\lfloor a \rfloor$ is the highest indivisible number not higher than a). To avoid resolving of the

1	$M^{kum} := 0,\ I^{part} := \{1, \ldots, I^{PW}\},\ I^{rest} := \{1, \ldots, I\}$. (Assumption: topological numbering of tasks)
2	*Use of an exact method* for balancing of I^{part}. Result: M^{part} stations with the station loads I^s_m, $m = M^{kum} + 1\ (1)\ M^{kum} + M^{part}$.
3	From the M^{part} stations the first $M^{fix} := \max\left\{1, \left\lfloor \text{PartlM} \cdot M^{part} \right\rfloor\right\}$ stations and their corresponding station loads are established finally. These are the stations $M^{kum} + 1, \ldots, M^{kum} + M^{fix}$ with the station loads I^s_m, $m = M^{kum} + 1\ (1)\ M^{kum} + M^{fix}$.
4	$I^{rest} := I^{rest} - \bigcup_{m=M^{kum}+1}^{M^{kum}+M^{fix}} I^s_m$.
5	$M^{kum} := M^{kum} + M^{fix}$.
6	If $\left\|I^{rest}\right\| > I^{PW}$ holds then determine I^{part} (the first I^{PW} tasks of I^{rest} (Assumption: topological numbering of tasks)) and go to 2.
7	*Use of an exact method* for balancing of I^{part}. Result: M^{part} stations with the station loads I^s_m, $m = M^{kum} + 1\ (1)\ M^{kum} + M^{part}$. $M := M^{kum} + M^{part}$. End of the algorithm.

Fig. 5. Exact solution of sliding problem windows.

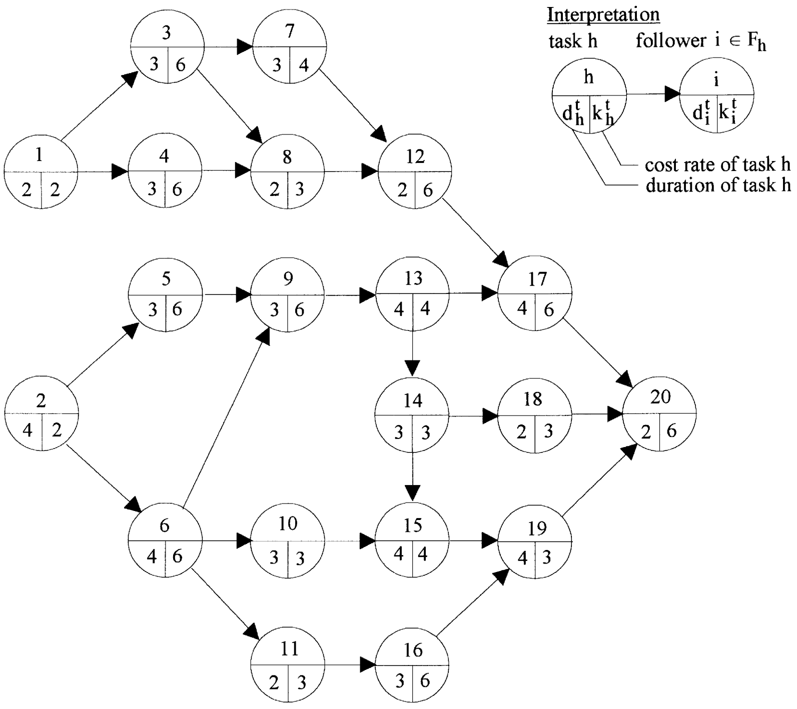

Fig. 6. Precedence graph for the example (cycle time $c = 10$ TU/PU).

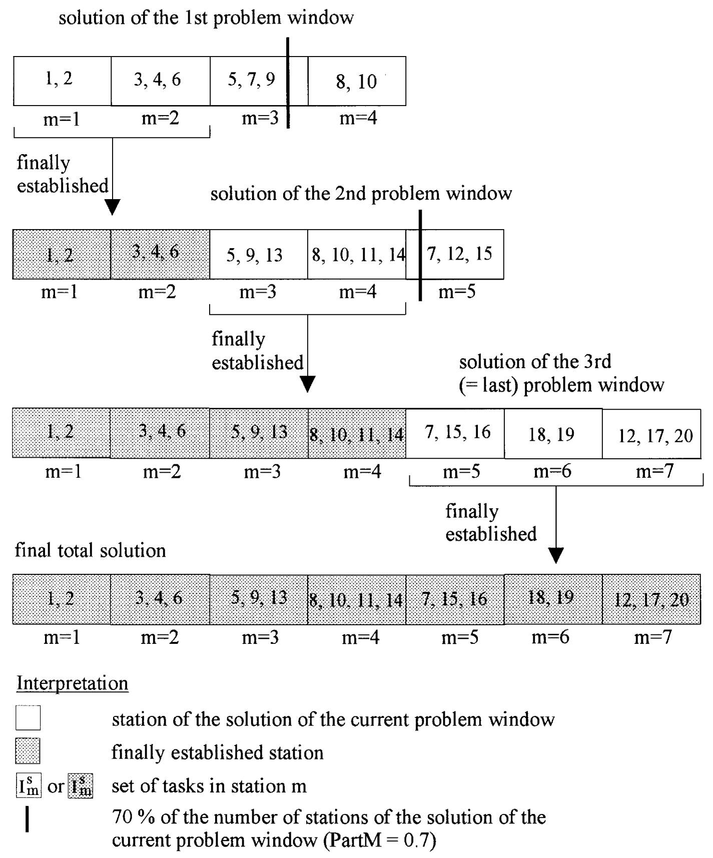
solution of the 1st problem window
1, 2
3, 4, 6
5, 7, 9
8, 10
m=1
m=2
m=3
m=4
finally established
solution of the 2nd problem window
1, 2
3, 4, 6
5, 9, 13
8, 10, 11, 14
7, 12, 15
m=5
finally established
solution of the 3rd (= last) problem window
7, 15, 16
18, 19
12, 17, 20
m=6
m=7
finally established
final total solution
Interpretation
station of the solution of the current problem window
finally established station
set of tasks in station m
70 % of the number of stations of the solution of the current problem window (PartM = 0.7)

Fig. 7. Illustration of the exact solution of sliding problem windows.

Table 1
Optimal solution of the 1st problem window

Station m	Station load I^s_m	Duration of the operation d^s_m (TU/PU)	Cost rate of the station k^s_m (MU/TU)
1	{1, 2}	6	2
2	{3, 4, 6}	10	6
3	{5, 7, 9}	9	6
4	{8, 10}	5	3

$M^{\text{part}} := 4$ stations were needed to perform the tasks 1–10. According to the formula $M^{\text{fix}} := \max\{1, \lfloor PartM \cdot M^{\text{part}} \rfloor\}$ we establish finally $M^{\text{fix}} := \max\{1, \lfloor 0.7 \cdot 4 \rfloor\} = \max\{1, \lfloor 2.8 \rfloor\} = 2$ stations. The tasks 1 and 2 are finally assigned to station 1, the tasks 3, 4 and 6 are finally assigned to station 2. The assignments of tasks to the stations 3 and 4 are abolished.

Table 2
Optimal solution of the 2nd problem window

Station m	Station load I^s_m	Duration of the operation d^s_m (TU/PU)	Cost rate of the station k^s_m (MU/TU)
1	{5, 9, 13}	10	6
4	{8, 10, 11, 14}	10	3
5	{7, 12, 15}	9	6

$M^{\text{part}} := 3$ stations were needed to perform the set of tasks in this problem window I^{part}. According to the formula $M^{\text{fix}} := \max\{1, \lfloor PartM \cdot M^{\text{part}} \rfloor\}$ we establish finally $M^{\text{fix}} := \max\{1, \lfloor 0.7 \cdot 3 \rfloor\} = \max\{1, \lfloor 2.1 \rfloor\} = 2$ stations. The tasks 5, 9 and 13 are finally assigned to station 3, the tasks 8, 10, 11 and 14 are finally assigned to station 4. The assignments of tasks to the station 5 are abolished.

Table 3
Optimal solution of the 3rd problem window

Station m	Station load I^s_m	Duration of the operation d^s_m (TU/PU)	Cost rate of the station k^s_m (MU/TU)
5	{7, 15, 16}	10	6
6	{18, 19}	6	3
7	{12, 17, 20}	8	6

$M^{\text{part}} := 3$ stations were needed to perform the set of tasks in this last problem window I^{part}. Because this is the last problem window we establish finally all M^{part} stations. The tasks 7, 15 and 16 are finally assigned to station 5, the tasks 18 and 19 are finally assigned to station 6, the tasks 12, 17 and 20 are finally assigned to station 7.

same problem window, which otherwise may occur according to the parameter constellations, we have to ensure that at least one station of the solution is established finally. Therefore we set $M^{\text{fix}} := \max\{1, \lfloor PartM \cdot M^{\text{part}} \rfloor\}$. The tasks assignments of the M^{part}–M^{fix} last stations of a solution of a problem window are abolished. In the next problem window a balance for the I^{PW} lowest numbered not yet assigned tasks has to be generated and the first M^{fix} stations have to be finally established. This procedure continues until all I tasks are finally assigned to a station.

5.2. *An example of the solution process*

The method described in Section 5.1 will now be illustrated by an example. Fig. 6 shows the precedence graph of the problem instance together with the durations and the cost rates of the tasks. A *precedence graph* $G = (\{1, \dots, I\}, R)$ consists of the set of tasks $\{1, \dots, I\}$ and the set of precedence relations R. A precedence relation with i as technological immediate follower of h, $i \in F_h$, is illustrated by an arrow (h, i) which is directed from h to i [21, p. 685]. The cycle time for this instance is given by $c = 10$ TU/PU. Note that the tasks of the instance are numbered topologically.

The 20-task-instance will be solved by a version of the method which works with a maximum of $I^{\text{PW}} = 10$ tasks in each problem window. From a solution of a problem window about 70% of the station loads generated were finally established, i.e. $PartM = 0.7$.

In our implementation of this method we used the *exact backtracking method* suggested by Amen [5,6] — this is the only existent method to generate an optimal solution for a cost-oriented assembly line balancing instance. The initial upper bound for the minimum cost per product unit which is needed in this exact method was obtained from the heuristic solution of the problem window by the use of the new priority rule “best change of idle cost” (P-MinKI).

Table 4
Final total solution

Station m	Station load I^s_m	Duration of the operation d^s_m (TU/PU)	Cost rate of the station k^s_m (MU/TU)	From solution of problem window no.
1	{1, 2}	6	2	1
2	{3, 4, 6}	10	6	1
3	{5, 9, 13}	10	6	2
4	{8, 10, 11, 14}	10	3	2
5	{7, 15, 16}	10	6	3
6	{18, 19}	6	3	3
7	{12, 17, 20}	8	6	3
Sum		60	32	
Cost	k	$10 \cdot 32 =$	320	MU/PU
Stations	M		7	

0	If $\sum_{i \in I^{rest}} d_i^t \le c$ holds then set $I_m^s := I^{rest}$ and stop (End of the algorithm).
1	Calculate I^{assign}.
2	Decompose I^{assign} into subsets $I_w^{assign,\ell}$, $w=1(1)W$.
3	$w := 1$.
4	If $\sum_{i \in I_w^{assign,\ell}} d_i^v \le c$ holds then establish a potential station load with the set of tasks $I_{m,w,1}^{s,k,pot} := I_w^{assign,\ell}$.
5	If $\sum_{i \in I_w^{assign,\ell}} d_i^t > c$ holds then establish U_w potential station loads $I_{m,w,u}^{s,k,pot}$ $(u=1(1)U_w)$. These potential station loads are the result of a fictitious balancing of the tasks which are in the set $I_w^{assign,\ell}$. In the fictitious balancing the tasks are assigned according to the priority rule P-MaxD and the tie-breaking priority rule P-MinI. (Note: U_w is the result of the fictitious balancing.)
6	If $w < W$ holds then set $w := w + 1$ and go to 4.
7	$w := 1$, $u := 1$, fixed $:= 0$.
8	If $\sum_{i \in I_{m,w,u}^{s,k,pot}} d_i^t + \min\left\{ d_j^t \mid j \in \left(I^{rest} - I_{m,w,u}^{s,k,pot}\right) \right\} > c$ holds then the potential station load is established finally. In this case set $I_m^s := I_{m,w,u}^{s,k,pot}$, $I^{rest} := I^{rest} - I_m^s$, $m := m+1$, fixed $:= 1$.
9	If $I^{rest} = \varnothing$ holds then stop (End of the algorithm).
10	If $u < U_w$ holds then set $u := u+1$ and go to 8.
11	If $w < W$ holds then set $u := 1$, $w := w + 1$ and go to 8.
12	If fixed $=1$ holds then go to 0 (Restart of phase 1). If fixed $=0$ holds then stop phase 1 (End of phase 1).

Fig. 8. Phase 1 of wage rate smoothing.

Fig. 7 gives a comprehensive illustration of the solution process for this example. As the exact method is beyond the topic of this paper the enumeration process and the dominance criteria of this backtracking-procedure are not explained here. In fact even for the solution of the first problem window with 10 task approximately 100 iterations were needed. Because to give detailed information on the exact method is much space- and time-consuming, it is left for further reading (see [5,6]).

(1) Definition and solution of the first problem window. The first problem window consists of the first $I^{PA} = 10$ lowest numbered (not yet finally assigned) tasks. These are the tasks 1–10. Using the exact method we get the solution shown in Table 1.

(2) Definition and solution of the second problem window. First we have to calculate the set of not yet finally assigned tasks I^{rest}. These are the tasks 5 and 7–20. The first $I^{PA} = 10$ lowest numbered task among the set I^{rest} are taken into the set of tasks of this problem window I^{part}. Therefore, we get $I^{part} := \{5, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$. Again, we solve this problem window with the exact method and calculate the solution shown in Table 2.

(3) Definition and solution of the third problem window. First we have to calculate the set of not yet finally assigned tasks I^{rest}. These are the tasks 7, 12 and 15–20. As the number of not finally assigned task $|I^{rest}| = 8$ is not higher than $I^{PA} = 10$, the third problem window is the last one. Therefore, we get $I^{part} := I^{rest} = \{7, 12, 15, 16, 17, 18, 19, 20\}$. Again, we solve this problem window with the

0	Set $I^{s,k,pot,2}_{m,w,u} := I^{s,k,pot}_{m,w,u}$, $u=1(1)U_w$, $w=1(1)W$.
1	$w := 1$, $u := 1$.
2	Calculate $I^{assign,pot}_{m,w,u}$ with $\left\{ i \mid i \in \left(I^{rest} - I^{s,k,pot,2}_{m,w,u} \right) \wedge k^t_i = k^{\ell}_w \right\}$.
3	If $I^{zuord,pot}_{m,w,u} = \emptyset$ holds then go to 9.
4	If $d^{s,pot}_{m,w,u} + \sum_{i \in I^{assign,pot}_{m,w,u}} d^t_i \le c$ holds then set $I^{s,k,pot,2}_{m,w,u} := I^{s,k,pot,2}_{m,w,u} \cup I^{zuord,pot}_{m,w,u}$ and go to 6.
5	Assign the tasks of $I^{assign,pot}_{m,w,u}$ to the potential station load $I^{s,k,pot,2}_{m,w,u}$. For this assignment use the priority rule P-MaxD and the tie-breaker P-MinI.
6	If $\sum_{i \in I^{s,k,pot,2}_{m,w,u}} d^t_i + \min \left\{ d^t_j \mid j \in \left(I^{rest} - I^{s,k,pot,2}_{m,w,u} \right) \right\} > c$ holds then the potential station load is established finally. In this case set $I^s_m := I^{s,k,pot,2}_{m,w,u}$, $I^{rest} := I^{rest} - I^s_m$, $m := m+1$, stop phase 2 and go to phase 1.
7	Calculate $I^{assign,pot}_{m,w,u}$ with $\left\{ i \mid i \in \left(I^{rest} - I^{s,k,pot,2}_{m,w,u} \right) \wedge k^t_i = k^{\ell}_w \right\}$.
8	If $I^{assign,pot}_{m,w,u} \neq \emptyset$ holds then go to 4.
9	If $u < U_w$ holds then set $u := u + 1$ and go to 2.
10	If $w < W$ holds then set $w := w+1$, $u := 1$ and go to 2.
11	From the calculated $I^{s,k,pot,2}_{m,w,u}$ choose the potential station load $I^{s,pot,phase2}_m$ according to the following lexicographic order of priority rules: 1. maximal work content of station m 2. maximal number of tasks in station m 3. maximal cost rate of station m 4. first considered potential station load End of phase 2.

Fig. 9. Phase 2 of wage rate smoothing.

exact method and obtain the solution shown in Table 3.

(4) Final total solution. Combining the final assignments which are calculated in the solutions of the problem windows we get the final total solution (Table 4).

As the minimum cost is 310 MU/PU, the solution generated by this method has the second lowest possible cost (320 MU/PU). To give additional information we have found that there exist 18 optimal solutions for this problem instance. In each of the optimal solutions 7 stations were needed. The lowest realizable number of stations is 6 (94 solutions). With 6 stations a minimum of 330 MU/PU for the cost is calculated. Note that the stations 1 and 6 are not loaded maximally. This is another evidence that loading the stations maximally could prevent one from generating low-cost solutions [5, p. 225; 6].

6. Summary and outlook

This paper is concerned with the existent and new heuristic methods for solving the cost-oriented assembly line balancing problem. The methods are classified and described in detail. Two new heuristic methods are presented: The priority rule "*best change of idle cost*" and the more sophisticated method "*exact solution of sliding problem windows*". The sliding problem window technique is a new class of its own. It can be taken as a *metaheuristic* because its possible application is not only for the assembly line balancing, but also for some other problems with precedence relations. In a further

0	Take the potential station loads $I^{s,k,pot}_{m,w,u}$ of phase 1, u=1(1) U_w, w=1(1)W.
1	$w := 1$, $u := 1$, $I^{rest}_{w,u} := I^{rest}$.
2	Calculate $I^{assign,pot}_{m,w,u}$ with $\left\{ i \mid i \in \left(I^{rest}_{w,u} - I^{s,k,pot}_{m,w,u} \right) \right\}$.
3	If $I^{assign,pot}_{m,w,u} = \varnothing$ holds then go to 8.
4	Choose the task i* from the set $I^{assign,pot}_{m,w,u}$ according to the following lexicographic ordered priority rules: 1. P-MinKts 2. if $d^{s,pot}_{m,w,u} < c/2$: P-MaxKt if $d^{s,pot}_{m,w,u} \geq c/2$: P-MinKt 3. P-MaxD 4. P-MinI
5	Check for the task i* in the following order: (1) $k^t_{i^*} \leq k^{s,pot}_{m,w,u}$ (2) $\sum_{i \in I^{rest} - I^{s,pot}_{m,w,u}} d^t_i > c$ (3) $k^{s,pot}_{m,w,u} > \max \left\{ k^t_i \mid i \in \left(I^{rest} - \left(I^{s,pot}_{m,w,u} \cup \{i^*\} \right) \right) \right\}$ As soon as a criterion is fulfilled assign the task i* to the potential station load, set $I^{s,k,pot}_{m,w,u} := I^{s,k,pot}_{m,w,u} \cup \{i^*\}$ and update the station cost rate. If none of these criteria is fulfilled then set $I^{rest}_{w,u} := I^{rest}_{w,u} - \{i^*\}$.
6	Calculate $I^{assign,pot}_{m,w,u}$ with $\left\{ i \mid i \in \left(I^{rest}_{w,u} - I^{s,k,pot}_{m,w,u} \right) \right\}$.
7	If $I^{assign,pot}_{m,w,u} \neq \varnothing$ holds then go to 4.
8	If $u < U_w$ holds then set $u := u + 1$, $I^{rest}_{w,u} := I^{rest}$ and go to 2.
9	If $w < W$ holds then set $w := w+1$, $u := 1$, $I^{rest}_{w,u} := I^{rest}$ and go to 2.
10	From the calculated $I^{s,k,pot}_{m,w,u}$ choose the potential station load $I^{s,pot,phase3}_m$ according to the following lexicographic order of priority rules:: 1. maximal work content of station m 2. maximal number of tasks in station m 3. first considered potential station load End of phase 3.

Fig. 10. Phase 3 of wage rate smoothing.

paper a comparison of the heuristic methods according to the solution quality and the computing time is presented [22].

Appendix

The formal discriptions of the phases 1–3 are given in Figs. 8–10.

References

[1] S. Ghosh, R.J. Gagnon, A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems, International Journal of Operations Research 27 (1989) 637–670.

[2] I. Baybars, A survey of exact algorithms for the simple assembly line balancing problem, Management Science 32 (1986) 909–932.

[3] R. Steffen, Produktionsplanung bei Fließbandfertigung, Gabler, Wiesbaden, 1977.

[4] O. Rosenberg, H. Ziegler, A comparison of heuristic algorithms for cost-oriented assembly line balancing, Zeitschrift für Operations Research 36 (1992) 477–495.

[5] M. Amen, Ein exaktes Verfahren zur kostenorientierten Fließbandabstimmung, in: U. Zimmermann et al., (Eds.), Operations Research Proceedings 1996, Springer, Berlin, 1997, pp. 224–229.

[6] M. Amen, An exact method for cost-oriented assembly line balancing, International Journal of Production Economics 64 (2000) 187–195.

[7] T.S. Wee, M.J. Magazine, Assembly line balancing as generalized bin packing, Operations Research Letters 1 (1982) 56–58.

[8] St.T. Hackman, H.J. Magazine, T.S. Wee, Fast, effective algorithms for simple assembly line balancing problems, Operations Research 37 (1989) 916–924.

[9] A. Scholl, Balancing and Sequencing of Assembly Lines, 2nd ed., Physica-Verlag, Heidelberg, 1999.

[10] J.R. Jackson, A computing procedure for a line balancing problem, Management Science 2 (1956) 261–271.

[11] F.N. Silverman, The effects of stochastic work times on the assembly line balancing problem, Ph.D. Thesis, Columbia University, New York, 1974.

[12] F.M. Tonge, Assembly line balancing using probabilistic combinations of heuristics, Management Science 11 (1965) 727–735.

[13] W.B. Helgeson, D.P. Birnie, Assembly line balancing using the ranked positional weight technique, Journal of Industrial Engineering 12 (1961) 394–398.

[14] R. Hahn, Produktionsplanung bei Linienfertigung, Walter de Gruyter, Berlin, 1972.

[15] Heizmann, J., Soziotechnologische Ablaufplanung verketteter Fertigungsnester zur Erhöhung der Flexibilität von Montage-Fließlinien, Jochem Heizmann Verlag, Karlsruhe, 1981.

[16] E.G. Coffman et al., Performance bounds for level-oriented two-dimensional packing algorithms, SIAM Journal on Computing 9 (1980) 808–826.

[17] C. Imboden, A. Leibundgut, P. Siegenthaler, Klassifikation heuristischer Prinzipien: Ein methodologischer Beitrag zur Entwicklung von heuristischen Verfahren, Die Unternehmung 32 (1978) 295–330.

[18] L.B.J.M. Sturm, An improved method for balancing assembly lines, Working paper R 70/1, Interfaculty for Graduate Studies in Management, Rotterdam, 1970.

[19] W.H. Thomas, N.R. Reeve, Balancing continuous stochastic assembly lines, in: American Institute of Industrial Engineers, Technical Papers, 23rd Annual Conference and Convention of the American Institute of Industrial Engineers, Anaheim, California, 1972, pp. 409–419.

[20] N.R. Reeve, W.H. Thomas, Balancing stochastic assembly lines, AIIE [American Institute of Industrial Engineers] Transactions 5 (1973) 223–229.

[21] R. Reiter, On assembly-line balancing problems, Operations Research 17 (1969) 685–700.

[22] M. Amen, Heuristic methods for cost-oriented assembly line balancing: A comparison on solution quality and computing time, International Journal of Production Economics 69 (2001) Forthcoming.