ELSEVIER

Int. J. Production Economics 69 (2001) 255–264

international journal of
production economics

www.elsevier.com/locate/dsw

Heuristic methods for cost-oriented assembly line balancing: A comparison on solution quality and computing time

Matthias Amen*

Institut für Unternehmensrechnung und Controlling, Universität Bern, Engehaldenstrasse 4, CH-3012 Bern, Switzerland

Abstract

This paper is focused on the solution quality and computing time requirements of heuristic methods for cost-oriented assembly line balancing. It is based on a recent paper (Amen, International Journal of Production Economics 68 (2000), which describes in detail the solution process of existent and two new heuristics. After a short review of the historic origin and the wideness of assembly line systems in present day industry, the paper emphasizes the economic view of production in order to cut down production cost. Results of a worst-case analysis concerning the solution quality and the computing time are presented. An interval for the worst-case-solution quality for most heuristic methods is given. The results of an experimental investigation show that the new priority rule "*best change of idle cost*" (Amen) achieves significantly better solutions than the existent priority rules. Furthermore, the new method "*exact solution of sliding problem windows*" (Amen) has been found to be the best heuristic method known so far.

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

1. Introduction

Today in mass production a huge number of units of the same product is produced. This is only possible with a high degree of division of labour. Since Adam Smith (1776) [1] it has been known that division of labour will train the required skills of the workers and will increase the productivity to a maximum. The maximum degree of division of labour is obtained by organizing production as an assembly line system.

Even in the early days of the industrial revolution mass production was already organized in assembly line systems. According to Salveson [[2], p. 18] the first assembly line was introduced by Eli Whitney during the French Revolution for the manufacturing of muskets (see also [[3], pp. 24–27]). The most popular example is the introduction of the assembly line on 1 April 1913, in the "John R-Street" of Henry Ford's "Highland-Park" production plant (cf. [[4], pp. 93–96]), where components of the famous "T-model" were manufactured. The decrease of production cost per unit is derived from the consequent organization of work and the application of Taylor's [5] "principles of scientific management" (cf. [6,7]).

Today assembly line production systems are still "up to date" because the principle to increase productivity by division of labour is timeless. The most known example is the final assembly in *automotive*

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

0925-5273/01/$ - see front matter
PII: S0925-5273(99)00096-1

industry. But *nearly all goods of daily life* are made by mass production which at its later stages is organized in *assembly line production systems*. For example the final assembly of consumer durables, like coffee machines, toasters, washing machines, refrigerators or products of the electrical industry like radio and TV or even personal computers is organized in assembly line systems.

The *characteristic problem* in assembly line systems is how to split up the total work to be done by the total system among the single stations of the line. This problem is called "*assembly line balancing*" because we have to find a "balance" of the work loads of the stations. First of all we have to determine the set of single tasks which have to be performed in the whole production system and the *technological precedence relations* among them. The *work load* of each station (also: *set of task, station load, operation*) is restricted by the cycle time, which depends on the fixed speed of the conveyor and the length of the stations. The *cycle time* is defined as the time between the entering of two consecutive product units in a station. A *task* is an economical indivisible work element. Because of the indivisibility of the task, the precedence relations among the tasks, and the fact that the tasks differ in their durations the allocation of task to stations – assembly line balancing – is a *combinatorial problem*.

In the literature usually the objective is to minimize the number of stations in a line for a given cycle time. This is called *time-oriented assembly line balancing*. As in recent years the industry was faced with sharp competitiveness the production cost has become more relevant. Even in such successful production systems like the assembly line system we have to look for possibilities to cut down production cost. As final assembly is usually a *labour-intensive* kind of production we have to analyse the existent *wage compensation system*.

Almost all collective agreements between unions and employers work with a wage differential in most developed industrial nations, e.g. in German industry which has been analyzed in detail. The higher the difficulties to perform a task, the higher the point value of the task – and the wage rate. As the tasks in final assembly are similar but not of unique difficulty there exist certain different wage rates in assembly line production systems. Under this economic perspective the objective in organizing work in assembly line production systems is not to minimize the number of stations, but *to minimize the total production cost per unit*. Therefore we have to allocate the tasks to stations in a way that both, cost rates and number of stations are considered. This is done in *cost-oriented assembly line balancing*. A formal description of this objective and the restrictions of this problem is given in [8–10]. As this paper is directly related to a previous work [9] the formal descriptions needed are reduced to a minimum.

Compared to existent balances which were obtained by the use of time-oriented methods neglecting wage rate differences, it is possible to realize savings in production cost up to a two-digit percentage by a cost-oriented reallocation of tasks using cost-oriented methods. The actual amount depends on the existent wage compensation system and the range of possible wage rates in final assembly.

2. Heuristic methods for solving the cost-oriented assembly line balancing problem

In the recent related survey [9] the existent heuristic methods for solving the cost-oriented assembly line balancing problem were described, and two new heuristic methods were presented: the new cost-oriented dynamic priority rule "*best change of idle cost*" and the more sophisticated "*method with exact solution of sliding problem windows*". The latter can be taken as a *metaheuristic* because the solution principle is not limited to assembly line balancing.

As heuristic methods do not necessarily find the optimal solution the main criterion to compare them is the solution quality. Furthermore we have to consider the computing time needed. In order to make a fair comparison on the solution quality and the computing time requirements of the heuristic methods, one has not only to evaluate the overall performance, but also the standing of the method in its class of similar heuristic methods. According to the similarities of the solution procedures and the effort invested to find a solution, the (exact and heuristic) methods were 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.

The methods which are included in this study are abbreviated by a capital letter according to the class of solution procedures and a short identifying extension. For a detailed formal description of the heuristic methods we refer to Amen [9]. The methods investigated are:

Class Z: Methods with random choice task assignment

The methods of this class differ in the number of randomly generated solutions (cf. [[11], pp. 25–35 and 44–46]).

- Z-1 1 randomly generated solutions,
- Z-10 10 randomly generated solutions,
- Z-100 100 randomly generated solutions,
- Z-1000 1000 randomly generated solutions.

Class P: Methods with one problem-oriented priority rule

- P-MaxD maximal task duration (cf. [[12], p. 728]),
- P-MaxR maximal ranked positional weight (cf. [[13] p. 395; in the modified version of Hahn [[14], pp. 44–46]),
- P-MaxF maximal number of immediate followers (cf. [[12], p. 728]),
- P-MaxKt maximal cost rate (cf. [[15], p. 481]),
- P-MinKt minimal cost rate (cf. [[16], pp. 106–107]),
- P-MinKts minimal absolute difference to the current station cost rate (cf. [[16], pp. 106–107]),
- P-MinKI best change of idle cost (cf. [9]).

Class H: Methods with several problem-oriented priority rules

- H-Stef heuristic method of Steffen [[16], pp. 105–110],
- H-Heiz heuristic method of Heizmann [[17], pp. 110–124],
- H-WR heuristic method 'wage rate' of Rosenberg and Ziegler [[15], p. 481]
- H-WRS heuristic method 'wage rate smoothing' of Rosenberg and Ziegler [[15], pp. 481–484].

All of the methods of the classes P and H use the *non-problem-oriented* priority rule P-MinI (minimal task number) (cf. [[12], p. 729]) as final tie breaker if necessary.

Class F: Methods with exact solution of sliding problem windows

The versions of this method differ in their parameter settings: I^{PW} is the maximum number of tasks in a single problem window. PartM is the proportion of the number of stations generated in a solution of a problem window to be finally established [9].

- F-t20s50: $I^{PW} = 20$ tasks, PartM = 0.5 (50% finally established stations),
- F-t20s70: $I^{PW} = 20$ tasks, PartM = 0.7 (70% finally established stations),
- F-t20s90: $I^{PW} = 20$ tasks, PartM = 0.9 (90% finally established stations),
- F-t30s50: $I^{PW} = 30$ tasks, PartM = 0.5 (50% finally established stations),
- F-t30s70: $I^{PW} = 30$ tasks, PartM = 0.7 (70% finally established stations),
- F-t30s90: $I^{PW} = 30$ tasks, PartM = 0.9 (90% finally established stations).

Class E: Exact method

- E-1-3600 exact method of Amen [8,10] with search for one optimal solution and a computing time limit of 3600 s per problem instance.

The method E-1-3600 is included in the experimental investigation in order to *obtain benchmark solutions* of high quality (optimal solutions or nearly optimal solutions in the case of run time stops).

3. Comparison of the heuristic methods

In Section 3.1 we present results of a worst-case analysis of the solution quality and the computing time requirements. Results of an experimental investigation with respect to both criteria, solution quality and computing time are given in Section 3.2. Within both sections the following basic symbols are used:

- I number of tasks, dimensionless
- i index for the tasks, $i = 1(1)I$
- M number of stations, dimensionless
- 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]
- d^t_i duration of task i, $i = 1(1)I$, [TU/PU]
- d^s_m duration of the operation to be performed in station m (work content), $m = 1(1)M$, [TU/PU]
- k total cost per unit [MU/PU]
- k^s_m cost rate of station m, $m = 1(1)M$, [MU/TU]
- k^t_i cost rate of task i, $i = 1(1)I$, [MU/TU]
- W number of different cost rates (wage differential), dimensionless

(Note: Dimensions: TU time units, PU product units, MU monetary units. Labels: s station, t task.)

3.1. Worst-case analysis

3.1.1. Solution quality

For calculating the *maximal possible deviation* of the cost of a heuristic solution k from the minimal cost k^{opt} the following general bounds can be used (Labels: opt optimum, ub upper bound, lb lower bound):

$$k^{\text{ub}} = M^{\text{ub}} c k^{t,\max}, \quad k^{\text{lb}} = M^{\text{lb}} c k^{t,\min},$$

where $k^{t,\max}$ is the maximum task cost rate and $k^{t,\min}$ is the minimum task cost rate of the problem instance. With this an upper bound for the *worst-case performance* can be obtained from the following expression (cf. [[15], pp. 485–486]):

$$\frac{k}{k^{\text{opt}}} \leqslant \frac{k^{\text{ub}}}{k^{\text{lb}}} = \frac{M^{\text{ub}} k^{t,\max}}{M^{\text{lb}} k^{t,\min}}.$$

As $k^{t,\max}$ and $k^{t,\min}$ are given by the instances the bound depends on M^{ub} and M^{lb}. All of the methods presented in this paper fulfil the well-known *two-stations-rule*: The sum of the work content of two consecutive stations m and $m+1$ has to exceed the cycle time, i.e. $d^s_m + d^s_{m+1} > c$, $m = 1(1)M-1$, where $d^s_m = \sum_{i \in I^s_m} d^t_i$. Because of this $M^{\text{ub}}/M^{\text{lb}} \leqslant 2$ holds. Thus it is well known that 2 is the upper bound for the worst-case ratio for M/M^{opt} in time-oriented assembly line balancing for all methods that fulfil the two-stations-rule. Therefore,

$$2\,\frac{k^{t,\max}}{k^{t,\min}}$$

is an *upper bound for the worst-case ratio* for k/k^{opt} in cost-oriented assembly line balancing *for all methods* presented in this paper.

For the heuristic methods which fulfil the well-known maximally-loaded-station-rule it is possible to give a lower bound for the worst-case ratio for k/k^{opt}. A station m is called *maximally loaded* if $d^s_m + d^t_i > c$ holds for any available task i, $m = 1(1)M-1$ (cf. [[18], p. 266]). A task is called *available* if it has no predecessor task or only predecessor tasks already assigned to the current or to an earlier station. The maximally-loaded-station-rule is used in all heuristic methods presented except the 'exact solution of sliding problem windows': As the *cost-oriented optimum can be missed when the stations are loaded maximally*, any exact method must not use this rule (cf. [[8], p. 225; [10]]).

As it is shown in Queyranne [[19], pp. 1354–1355] for any polynomial heuristic which fulfils the maximally-loaded-station-rule the worst-case ratio for M/M^{opt} is within the interval [1.5, 2] unless P $\neq$ NP. In the next section it is stated that all heuristic methods presented in this paper have a polynomial computing time. As a result for *all these heuristic methods except the "exact solution of* *sliding problem windows*" we can state that in cost-oriented assembly line balancing the *worst-case ratio* for k/k^{opt} is within the interval

$$\left[1.5\,\frac{k^{t,\max}}{k^{t,\min}},\, 2\,\frac{k^{t,\max}}{k^{t,\min}}\right].$$

A further tightening is not known.

3.1.2. Computing time

The worst-case analysis of the computing time depends on the *data structure* which has been used in the implementation of the methods (cf. [[20], pp. 21–33]). In our implementations of all heuristic methods the data type "set" is used to store any set of tasks (e.g. station loads, sets of preceding tasks). With this data type computing time is saved because it is possible to use elementary bit operations. The time complexities (cf. [[21], p. 6]) which have been calculated for the heuristic methods are shown in Table 1.

For most of the heuristic methods the worst-case-computing time is restricted by a quadratic function of the number of tasks I (problem size). The *wage rate smoothing* method has a time complexity of $O(WI^3)$ which is much higher. Its worst-case-computing time depends on the number of different cost rates W and the number of tasks I and thus is restricted by a polynom of degree 4.

For the "*exact solution of sliding problem windows*" the O-notation is shown together with a factor $b(I^{\text{PW}})$. The *main procedure* of this metaheuristic itself has a linear worst-case-computing time $O(I)$. As the worst-case-computing time for solving a single problem window by the *exact method* depends linearly on the number of different cost rates W and exponentially on the maximum number of tasks in a problem window I^{PW}, the total worst-case-computing time can be written as $b(I^{\text{PW}})O(WI)$ if the tasks are already numbered topologically and $b(I^{\text{PW}})O(I^2)$ if they have to be renumbered first. In order to make a fair comparison the dependency of the total worst-case-computing time on I^{PW} must not be neglected. The total worst-case-computing time increases exponentially when the value of the parameter I^{PW} increases. On the other hand, the total worst-case-computing time $b(I^{\text{PW}})$ is constant for a given value of I^{PW}.

As a result all heuristic methods are "*efficient*" [[20], p. 39; [21], pp. 8–9] because of their polynomial worst-case-computing time. Note that the methods are "*provable efficient*" except for two: The "wage rate smoothing" method is not provable efficient because its computing time is bounded by a polynom of degree 4. The "exact solution of sliding problem windows" is not provable efficient as its computing time is bounded by a polynom with extreme large coefficients.

3.2. Experimental analysis

3.2.1. Design of the experiment

For evaluating the performance of the heuristic methods test problem instances have been generated

Table 1
Time complexities of the heuristic methods

	Heuristic methods	Time complexity
Z	Methods with random choice task assignment	$O(I^2)$
P	Methods with one problem-oriented priority rule	$O(I^2)$
H-Stef	Method of Steffen	$O(I^2)$
H-Heiz	Method of Heizmann	$O(I^2)$
H-WS	Wage rate	$O(I^2)$
H-WRS	Wage rate smoothing	$O(WI^3)$
F	Methods with exact solution of sliding problem windows	
	if the tasks are already numbered topologically	$b(I^{\text{PW}})O(WI)$ with $b(I^{\text{PW}})$ exponential
	if the tasks are not yet numbered topologically	$b(I^{\text{PW}})O(I^2)$ with $b(I^{\text{PW}})$ exponential

independently and randomly using the following *parameters*:

(1) Number of tasks I, [dimensionless]:	50, 75, 100
(2) Order strength OS, [dimensionless]:	0.7, 0.8, 0.9
(3) Cycle time to maximal task duration $c/d^{t,\max}$, [dimensionless]:	2, 3
(4) Maximal to minimal task duration $d^{t,\max}/d^{t,\min}$, [dimensionless]:	5, 10
(5) Maximal to minimal task cost rate $k^{t,\max}/k^{t,\min}$, [dimensionless]:	1.25, 1.50, 2.00

The *importance of these parameters* for characterizing cost-oriented assembly line balancing problem instances is evident. By definition assembly line balancing is the assignment of a given number of tasks (I) with different durations ($d^{t,\max}/d^{t,\min}$) to a not yet known number of stations under certain restrictions: The technological precedence relations (OS) must not be violated and the sum of the durations of the tasks must not exceed the cycle time ($c/d^{t,\max}$). In cost-oriented assembly line balancing we have to cope with different cost rates ($k^{t,\max}/k^{t,\min}$) – caused by wage rate differences (according to the different point values of the tasks) in order to minimize the total cost per product unit. Therefore the parameter set describes the size of the problem instances (I), the set of restrictions (OS, $c/d^{t,\max}$ and $d^{t,\max}/d^{t,\min}$) and the objective function characteristics ($k^{t,\max}/k^{t,\min}$). These or quite similar parameter sets were used in several experimental investigations done by other researchers (e.g. [15,22,23]).

(1) *Number of tasks I.* It is well known that when the number of task increases it becomes more difficult to find good solutions. The chosen values of I (50, 75 and 100 tasks) are representative of *small- and medium-sized problem instances* which occur in industrial practice. The actual number of tasks in large practical industry problem instances can be several hundreds or even thousands (cf. [[24], p. 157]). Experiments in time-oriented assembly line balancing also work with a maximum of 100 tasks (cf. [[22], p. 438; [15], p. 487]). By the way it is very important how to define a task. Usually a *task* consists of a variety of "atomic" work elements which are already combined together because dividing them among several workers will cause additional work. For example, the worker who puts a screw into a hole should be the same who fixes it.

(2) *Order strength OS.* The order strength OS is a global measure for the strength of the connection of the total set of tasks. It is calculated as the *number of existing direct and indirect precedence relations divided by the theoretical maximal number* (cf. [[25], pp. 735–736]). With F_i^a set of all tasks, which can be performed if and only if the task i has been performed (set of technological direct and indirect followers) the formula is: $OS = \sum_{i=1}^{I} |F_i^a| / \frac{1}{2} I(I-1)$. The more the tasks are connected, the higher is the order strength and the lower is the number of possible assignments of tasks to operations. Therefore, it is much more difficult to obtain a good solution for problem instances with low order strength than for problem instances with high order strength. In academic investigations the order strength is usually much lower as in this investigation (cf. [[23], p. 233]). For example, Talbot et al. [[22], p. 439] take order strengths of 0.2, 0.3, 0.5 and 0.8. But as in industry the tasks are strongly connected the actual order strengths are usually very high (cf. [[26], p. 41]). Therefore, the values 0.7, 0.8 and 0.9 were chosen for this problem parameter.

(3) *Cycle time to maximal task duration* $c/d^{t,max}$. This measure provides a simple lower bound for the minimum number of tasks in a station. In other experiments ratios between 1.0 and 2.0 [[22], p. 439] or ratios of 2.0, 2.4 and 3.0 [[15], p. 487] were used. Because in German industry there is a tendency to increase the number of different tasks in each station to prevent taylorism the higher ratios 2.0 and 3.0 were chosen. An illustration for this tendency is that several collective agreements between unions and employers (e.g. in the metal and electrical industry – which covers such important branches as the automotive industry and the consumer durable industry – in Niedersachsen (Lower Saxony), Sachsen-Anhalt (Saxony-Anhalt)

and Nordwürttemberg/Nordbaden (North Württemberg/North Baden)) limit the cycle time by a lower bound of 90 seconds.

(4) *Maximal to minimal task duration* $d^{t,max}/d^{t,min}$. The ratio $d^{t,\max}/d^{t,\min}$ is a simple measure for the homogeneity or heterogeneity of the task durations. If there are no differences the assembly line balancing problem is trivial. The higher the ratio, the more difficult is the assignment process. For the experiment values of 5 and 10 were taken (same values as in [[15], p. 487]). The task durations of the problem instances were equally distributed between $d^{t,\min}$ and $d^{t,\max}$. Talbot et al. [[22], pp. 438–439] use a ratio of 30 where the task durations follow the binomial distribution. Sometimes much higher values of $d^{t,\max}/d^{t,\min}$ are reported (cf. [[23], p. 233]). It is clear that the values of this parameter depend on the definition of the term "task" used. For final assembly which is usually a labour-intensive kind of production it can be assumed that there are no extreme differences in the task durations because of the similar nature of work.

(5) *Maximal to minimal task cost rate* $k^{t,max}/k^{t,min}$. This parameter is specific for cost-oriented assembly line balancing. By definition $k^{t,\max}/k^{t,\min}$ has to exceed 1, otherwise the instance is a time-oriented assembly line balancing problem. The higher $k^{t,\max}/k^{t,\min}$, the more important it is to allocate only task with no or only moderate differences in their cost rates to the same station. Note that in manual assembly, differences between the task cost rates are only due *to wage rate differences* according to the point values of the tasks. Rosenberg and Ziegler [[15], p. 487] take ratios between 1.25 and 5.00. In fact a ratio of 5.00 will never occur in practice to our knowledge. In our experiment the upper limit is given by 2.00. With this value it is assumed that the most skilled worker at the assembly line earns twice as much as his lowest skilled colleague. This may happen if there are certain difficult tasks which require special abilities of the worker, e.g. a high degree of knowledge of the product and the production process is needed for quality control at the end of the line. In several important collective agreements between unions and employers the wage differential between the lowest and the highest wage rate is somewhere within the range of $k^{t,\max}/k^{t,\min} = 1.50$ and $k^{t,\max}/k^{t,\min} = 2.00$. If the cost of capital is also considered this ratio decreases. Therefore it is justified to include $k^{t,\max}/k^{t,\min} = 1.25$ into the investigation. In fact the absolute amounts of the cost rates are of no interest, neither for studying the problem structure, nor for solving a problem instance. Wage rates/cost rates are specific prices, and prices are only exchange ratios for resources and goods.

In each of the resulting 108 test fields ($2^2 \times 3^3$ experimental design) 50 problems were generated randomly and independently. This totals to 5400 test problems. Each test problem has been solved by each of the heuristic methods.

All methods have been implemented in BORLAND PASCAL 7.0 (protected mode) under the operating system MS-DOS 6.22. The computing times have been standardized on a 133 MHz Intel-Pentium PC with 16 MB RAM (60 ns).

3.2.2. Solution quality

To compare the solution quality of the heuristic methods a *benchmark* is needed. Therefore, all of the test problem instances were solved by the exact method (E-1-3600) suggested by Amen [8,10]. The run time of this method was restricted to 3600 seconds. As it was not possible to solve all test problem instances optimally within this time limit, and as there is a systematical deviation of the lower bound for the cost per product unit from the optimum, we have to use an alternative benchmark: For each heuristic solution we calculate *the percentage increase of the total cost per product unit over the best solution found by all methods* included in this investigation. As we have used the exact method we know that the benchmark is the optimal value for at least 86.48% of the test problem instances.

In Table 2 the methods are ranked according to the mean value of their percentage increase over the benchmark. Table 2 shows the mean, the standard deviation, the maximum deviation, and the percentage of best heuristic solutions found. Furthermore, the result of a *multiple z-test* is presented. For a single comparison we used $\alpha^{\text{single}} = 0.0001$. The total error of the multiple z-test therefore amounts to $\alpha^{\text{total}} = 0.02284$. The critical value for each single comparison is $z_{\text{cr}} = 3.89059$. As α^{single} has a small value only those differences for which there is

Table 2
Solution quality of the heuristic methods

Rank	Method	Increase over benchmark			Best solutions found (%)	Level of significance α^{total}: 0.02284 α^{single}: 0.0001 z_{cr}: 3.89059 ∣ no significant difference
		Mean (%)	Standard deviation (%)	Maximum (%)		
1	E-1-3600	0.0465	0.3172	4.9	96.96	
2	F-t30s50	2.8394	2.3424	14.6	16.85	
3	F-t30s70	3.0029	2.4385	21.4	17.15	
4	F-t30s90	3.0251	2.3212	14.1	15.22	
5	F-t20s50	4.3742	2.7335	19.1	6.72	
6	F-t20s70	4.7128	2.7346	19.1	5.35	
7	F-t20s90	5.4423	3.0171	21.4	4.52	
8	Z-1000	6.8867	3.3094	17.7	2.04	
9	H-WRS	7.7618	3.4812	24.7	0.80	
10	Z-100	8.4090	3.6109	19.6	0.98	
11	H-Stef	8.7092	3.6671	25.9	0.56	
12	H-Heiz	9.3014	3.9297	27.6	0.43	
13	P-MinKI	9.6731	3.8708	27.6	0.33	
14	Z-10	10.5305	3.9705	24.1	0.35	
15	P-MinKts	10.6063	3.9670	27.4	0.26	
16	P-MaxD	10.8954	4.1164	31.0	0.24	
17	P-MaxR	12.1654	4.3029	27.9	0.17	
18	H-WR	12.3155	4.4601	28.2	0.11	
19	P-MaxKt	13.4885	4.5761	34.8	0.13	
20	P-MaxF	13.7132	4.5824	31.2	0.04	
21	P-MinKt	13.7937	4.7429	33.7	0.22	
22	Z-1	14.0914	4.6521	34.5	0.09	

Note: 5400 test problem instances per method. z follows the standardized normal distribution.

a high probability that they exist in reality are found to be significant.

In Table 2 the data for the computing-time-bounded exact method (E-1-3600) are included. For 96.96% of all test problem instances the E-1-3600 has found the best solution known. The data in Table 2 show that there exist significant differences between the classes of the deterministic heuristic methods. The *best results* were obtained by the "*exact solution of sliding problem windows*" (class F). The version F-t30s50 has a mean increase of only 2.8394% over the lowest cost per product unit found by all 22 methods.

Almost all methods which use several problem-oriented priority rules (class H) dominate those which use only one problem-oriented priority rule (class P). The only exception is the "wage rate" method (H-WR). The new priority rule "*best change of idle cost*" (P-MinKI) significantly *dominates all other single priority rules* (class P).

The *random choice task assignment* (class Z) achieves *considerable good results* when the number of balances is high. With 10 balances (Z-10) it achieves a better solution quality on an average than all single priority rules (class P) except the new "best change of idle cost" (P-MinKI). With 100 balances (Z-100) the results are significantly better than all of the heuristic methods which use several problem-oriented priority rules (class H) except the "wage rate smoothing" (H-WRS). Only the new "exact solution of sliding problem windows" (class F) has a better performance than the random choice method with 1000 balances (Z-1000). It can be expected that the solution quality of the implemented versions of this new method can only be achieved by random choice task assignment with a much higher number of randomly generated balances.

Furthermore, the *rank order* of the deterministic methods *remains almost stable* when variations of the problem parameters occur. It has been found that the seldom pairwise changes in the rank order of the deterministic methods occur only when the methods involved have no significant difference in their solution quality.

3.2.3. Computing time

For all methods which use only one problem-oriented priority rule (class P), for H-Stef, for H-Heiz, and for H-WR a maximum computing time of only 0.05 seconds has been measured. This is the shortest possible duration which can be measured by the PC. The maximum computing time for H-WRS was only 0.99 seconds. For the random choice method Z-1000 a maximum of 35.40 seconds has been measured. The computing times for H-WRS and Z-1000 have been analysed in detail. As there is no significant difference between the solution process of these methods in the average case and in the worst case the measured computing times comply with the corresponding O-notations which are listed in Table 1 (see [[20], p. 42]).

For the "exact solution of sliding problem windows" (class F) a maximum computing time of 4795.11 seconds has been measured for a problem solved by the version F-t30s50. The maximum mean computing time was 128.04 seconds and occurred for $I = 100$, $W = 5$ and the version F-t30s50. As there is a huge number of different possible runs of the exact backtracking method which is used to solve a single problem window, it is not possible to see that the measured data follow the O-notation.

Since balancing an assembly line is a *long-term decision* the computing times of all heuristic methods presented in this paper are neglectable.

4. Conclusion

In this paper a comparison of existent and new heuristic methods for solving the cost-oriented assembly line balancing problem is presented. To draw a conclusion the new methods "best change of idle cost" and "exact solution of sliding problem windows" can be strongly recommended to solve the cost-oriented assembly line balancing problem within acceptable computing time. The priority rule "*best change of idle cost*" works significantly better than any other priority rule. The "*exact solution of sliding problem windows*" is the best heuristic method known. Furthermore, an interesting result is that the rank order of the deterministic heuristic methods remains almost stable when changes occur in the characteristics of the problem instances. Because of this the new methods can be strongly recommended to solve problem instances that occur in practice, regardless of the characteristics of the actual real-world problem.

References

[1] A. Smith, An Inquiry into the Nature and Causes of the Wealth of Nations, 1st Edition, 1776 (cited from: Der Wohlstand der Nationen – Eine Untersuchung seiner Natur und seiner Ursachen, Vollständige Ausgabe nach der 5. Auflage (letzter Hand), London 1789, Deutscher Taschenbuch Verlag, 1978).

[2] M.E. Salveson, The assembly line balancing problem, Journal of Industrial Engineering 6 (1955) 18–25.

[3] R. Wild, Mass-production Management – The Design and Operation of Production Flow-line Systems, Wiley, London, 1972.

[4] H. Ford, Mein Leben und Werk, 27th Edition, Paul List Verlag, Leipzig, 1923.

[5] F.W. Taylor, The Principles of Scientific Management, Harper & Brothers Publishers, New York/London, 1911.

[6] K. Williams, C. Haslam, J. Williams, Ford versus 'Fordism': The beginning of mass production, Work, Employment and Society 6 (1992) 517–555.

[7] K. Williams et al., The myth of the line: Ford's production of the Model T at Highland Park, 1909-16, Business History 35 (1993) 66–87.

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

[9] M. Amen, Heuristic methods for cost-oriented assembly line balancing: A survey, International Journal of Production Economics 68 (2000) 1–14.

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

[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/New York, 1972.
[15] O. Rosenberg, H. Ziegler, A comparison of heuristic algorithms for cost-oriented assembly line balancing, Zeitschrift für Operations Research 36 (1992) 477–495.
[16] R. Steffen, Produktionsplanung bei Fließbandfertigung, Gabler, Wiesbaden, 1977.
[17] J. Heizmann, Soziotechnologische Ablaufplanung verketteter Fertigungsnester zur Erhöhung der Flexibilität von Montage-Fließlinien, Jochem Heizmann Verlag, Karlsruhe, 1981.
[18] J.R. Jackson, A computing procedure for a line balancing problem, Management Science 2 (1956) 261–271.
[19] M. Queyranne, Bounds for assembly line balancing heuristics, Operations Research 33 (1985) 1353–1359.
[20] E. Horowitz, S. Sahni, Fundamentals of Data Structures in Pascal, 4th Edition, Computer Science Press, New York, 1994.
[21] M.R. Garey, D.S. Johnson, Computers and Intractability. A Guide to the Theory of NP-completeness, Update March 1991, W.H. Freeman and Company, New York, 1979.
[22] F.B. Talbot, J.H. Patterson, W.V. Gehrlein, A comparative evaluation of heuristic line balancing techniques, Management Science 32 (1986) 430–454.
[23] A. Scholl, Balancing and Sequencing of Assembly Lines, 2nd ed., Physica-Verlag, Heidelberg, 1999.
[24] K.-P. Kistner, M. Steven, Produktionsplanung, 2nd Edition, Physica-Verlag, Heidelberg, 1993.
[25] A.A. Mastor, An experimental investigation and comparative evaluation of production line balancing techniques, Management Science 16 (1970) 728–746.
[26] St.C. Graves, C.A. Holmes Redfield, Equipment selection and task assignment for multiproduct assembly system design, International Journal of Flexible Manufacturing Systems 1 (1988) 31–50.