Chapter 41
Analysis of a Few Heuristics Proposed Based on Slope Indices to Solve Simple Type—I Assembly-Line Balancing Problems

A. Baskar, M. Anthony Xavior, N. Nithyanandan and B. Dhanasakkaravarthi

Abstract In an assembly line, any product is subdivided into many tasks which may include subassemblies and processing. These tasks are carried out in several work stations which are responsible for a single or a set of operations. Assembly lines need to be balanced to have even distribution of work for both men and machines. Type—1 simple assembly-line balancing problems (SALBP-1) refer to minimization of number of work stations by keeping the cycle time constant. This paper proposes a new set of heuristics that can be used to solve simple type—1 assembly-line balancing problems and analyzes them using a few benchmark problems available in the literature. They use slope indices to order the jobs and allot them to different work stations.

Keywords Assembly-line balancing · Type—I problem · Heuristics · Line efficiency · Smoothing index

41.1 Introduction

Let, there are 'n' tasks that comprise a job that needs to be completed in an assembly line and the corresponding time of completion be tj ($j = 0$ to n). The order of processing is partially controlled by certain constraints called as the 'precedence' that are not to be violated. This defines the tasks that need to be completed before starting a particular task. The tasks are carried out in different work stations that are to be balanced to the possible extent. The maximum time to be spent in a work station defines the 'cycle time, c.' The cycle time being constant, the objective is to minimize the number of work stations, m*.

A. Baskar (✉) · N. Nithyanandan · B. Dhanasakkaravarthi
Panimalar Institute of Technology, Chennai 600123, India
e-mail: a.baaskar@gmail.com

M. Anthony Xavior
Vellore Institute of Technology, Vellore 632014, India

M. S. Shunmugam and M. Kanthababu (eds.), *Advances in Simulation, Product Design and Development*, Lecture Notes on Multidisciplinary Industrial Engineering,
https://doi.org/10.1007/978-981-32-9487-5_41

The solution procedures include exact methods and heuristic methods. The exact methods are not suitable for larger problems as the computation time grows exponentially with the problem size. SALBP-1 are NP hard [1] and hence, efficient heuristics are required to solve within a reasonable time period. As a result, researchers have developed many efficient heuristic methods over the years.

Positional weight method [2], procedure based on number of predecessors [3], heuristic using trade and transfer [4], heuristic of Hackman et al. [5], precedence matrix method proposed by Hoffmann [6] are a few efficient heuristic methods proposed during earlier periods of research. Many heuristics are available in the literature based on simple as well as combined priority rules. It is generally accepted that Hoffmann's algorithm is still one of the best simple algorithms in this domain, at the cost of execution time.

In the next level, branch and bound method, dynamic programming and evolutionary algorithms are used by many to refine the accuracy of heuristics. Most of the evolutionary algorithms take the results from the simple heuristics as their seed solutions and proceed.

Sivasankaran and Shahabudeen [7] classified assembly-line balancing problems into eight types and conducted a comprehensive review on the available methods.

41.2 Data Set and Heuristics Considered

A precedence diagram is a graphical representation of a project that shows the number of tasks, their respective task times, and the sequence of tasks that need to be completed before a particular task. Figure 41.1 shows the precedence diagram of a SALBP-1 analyzed by Rosenberg and Ziegler [8].

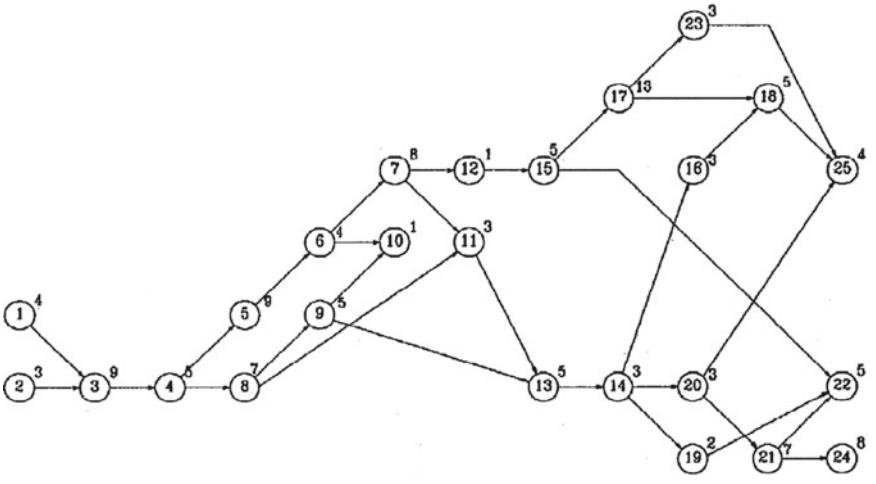

Fig. 41.1 A precedence diagram—Rosenberg and Ziegler

Table 41.1 Cycle time and optimum number of work stations

Rosenberg	c	m^*	c	m^*	c	m^*	c	m^*	c	m^*	c	m^*
25 Tasks	14	10	16	8	18	8	21	6	25	6	32	4

This particular project has 25 tasks with a total time of 125 units. For analyzing the heuristics considered, this particular problem is considered as it is a reasonably large sized tested data set. To have more number of problems, the cycle time is considered for different values, from 14 units to 32 units as listed in Table 41.1.

$$\begin{aligned}\text{Strength of the precedence, } D &= 2d/(N(N-1)) \\ &= (2 \times 32)/(25(25-1)) \\ &= 0.11\end{aligned}$$

Since the same data set is tested for six cycle times, we get a total of six SALBP-1.

'c' represents the cycle time and 'm^*' represents the optimum number of work stations for a particular cycle time. The optimum number of work stations for different cycle times is available in the data sets provided by Scholl [9] and is reproduced in Table 41.1. The parameters considered are listed in Table 41.2.

The precedence diagram is transformed into a precedence matrix as shown in the left half of Table 41.3 which can be directly used in a computer program. The matrix is appended with other required parameters (obtained from the precedence diagram) as described and presented in the right half of the same table.

Based on a presumption that simultaneously considering the same parameter before and after a task being considered can result in better algorithms; slope indices (ratios) are being computed for a particular task. They are the ratios of a particular parameter before and after a task.

Table 41.2 Parameters considered

S. No.	Parameter	Notation
1.	Total number of tasks having 'j' as its head (including self)	a
2.	Total number of tasks having 'j' as its tail (including self)	b
3.	Maximum number of tasks having 'j' as its head (including self)	c
4.	Maximum number of tasks having 'j' as its tail (including self)	d
5.	Number of immediate predecessors of 'j' (including self)	e
6.	Number of immediate successors of 'j' (including self)	f
7.	Number of levels prior to 'j' (including self)	g
8.	Number of levels after 'j' (including self)	h
9.	Position weight of 'j' from head (reverse position weight) … including self	i
10.	Position weight of 'j' from tail … including self	k

Table 41.3 Precedence matrix and other parameters

Task No. (j)	Task time (t_j)	Predecessor	Successor	Parameter considered									
				a	b	c	d	e	f	g	h	i	k
1	4	–	3	25	1	25	1	1	2	1	12	4	122
2	3	–	3	25	1	25	1	1	2	1	12	3	121
3	9	1, 2	4	23	3	23	3	3	2	2	11	16	118
4	5	3	5, 8	22	4	22	4	2	3	3	10	21	109
5	9	4	6	19	5	9	4	2	2	4	9	30	92
6	4	5	7, 10	18	6	8	5	2	3	5	8	34	83
7	8	6	11, 12	16	7	7	6	2	3	6	7	42	78
8	7	4	9, 11	14	5	7	4	2	3	4	7	28	61
9	5	8	10, 13	12	6	6	5	2	3	5	6	33	51
10	1	6, 9	–	1	9	1	6	3	1	6	1	47	1
11	3	7, 8	13	11	9	6	7	3	2	7	6	58	48
12	1	7	15	7	8	5	7	2	2	7	6	43	36
13	5	9, 11	14	10	10	5	8	3	2	8	5	63	45
14	3	13	16, 19, 20	9	12	4	9	2	4	9	4	66	40
15	5	12	17, 22	6	9	4	8	2	3	8	5	48	35
16	3	14	18	3	12	3	10	2	2	10	3	67	12
17	13	15	18, 23	4	10	3	9	2	3	9	4	61	25
18	5	16, 17	25	2	16	2	11	3	2	11	2	95	9
19	2	14	22	2	13	2	10	2	2	10	2	67	7
20	3	14	21, 25	4	13	3	10	2	3	10	3	69	27
21	7	20	22, 24	3	14	2	11	2	3	11	2	76	20
22	5	15, 19, 21	–	1	17	1	12	4	2	12	1	88	5
23	3	17	25	2	11	2	10	2	2	10	2	64	7
24	8	21	–	1	15	1	12	2	1	12	1	84	8
25	4	18, 20, 23	–	2	21	2	12	4	1	12	1	125	4
	$N = 125$	$d = 32$	$d = 32$										

The weights $w1$ to $w5$ are the slope indices as computed below. Weight 'w' is the rule proposed by Dar-El; weight, $w = a =$ total number of tasks having 'j' as its head (including self). Weight $w6$ is computed in a different way but, using tj, k and i that are used for $w5$ in a slope format. For any task 'j,' the slope is the ratio between the right and left parameter values.

$$\text{Weight, } w = a; \text{ Weight, } w1 = tj\frac{a}{b}; \text{ Weight, } w2 = tj\frac{c}{d}$$

$$\text{Weight, } w3 = tj\frac{f}{e};\ \text{Weight, } w4 = tj\frac{h}{g};\ \text{Weight, } w5 = tj\frac{k}{i}$$
$$\text{Weight, } w6 = tj(k-i).$$

Table 41.4 shows the parameters converted as weights for different heuristics. For solving the problems, the weights are arranged in descending order of their weights.

The two popular time-tested algorithms proposed by Dar-El [10] and Hoffmann [6] are taken as the benchmarks.

Table 41.4 Weights considered for the analysis

Task No. (j)	Task time (t_j)	Predecessor	Successor	Weights					
				$t(a/b)$	$t(c/d)$	$t(f/e)$	$t(h/g)$	$t(k/i)$	$t(k-i)$
1	4	–	3	100	100	8	48	122	472
2	3	–	3	75	75	6	36	121	354
3	9	1, 2	4	69	69	6	49.5	66.375	918
4	5	3	5, 8	27.5	27.5	7.5	16.667	25.952	440
5	9	4	6	34.2	20.25	9	20.25	27.6	558
6	4	5	7, 10	12	6.4	6	6.4	9.765	196
7	8	6	11, 12	18.286	9.333	12	9.333	14.857	288
8	7	4	9, 11	19.6	12.25	10.5	12.25	15.25	231
9	5	8	10, 13	10	6	7.5	6	7.727	90
10	1	6, 9	–	0.111	0.167	0.333	0.167	0.021	−46
11	3	7, 8	13	3.667	2.571	2	2.571	2.483	−30
12	1	7	15	0.875	0.714	1	0.857	0.837	−7
13	5	9, 11	14	5	3.125	3.333	3.125	3.571	−90
14	3	13	16, 19, 20	2.25	1.333	6	1.333	1.818	−78
15	5	12	17, 22	3.333	2.5	7.5	3.125	3.646	−65
16	3	14	18	0.75	0.9	3	0.9	0.537	−165
17	13	15	18, 23	5.2	4.333	19.5	5.778	5.328	−468
18	5	16, 17	25	0.625	0.909	3.333	0.909	0.474	−430
19	2	14	22	0.308	0.4	2	0.4	0.209	−120
20	3	14	21, 25	0.923	0.9	4.5	0.9	1.174	−126
21	7	20	22, 24	1.5	1.273	10.5	1.273	1.842	−392
22	5	15, 19, 21	–	0.294	0.417	2.5	0.417	0.284	−415
23	3	17	25	0.545	0.6	3	0.6	0.328	−171
24	8	21	–	0.533	0.667	4	0.667	0.762	−608
25	4	18, 20, 23	–	0.381	0.667	1	0.333	0.128	−484

41.3 Performance Measures

Perfect balancing of work stations is important to reduce the idle time of individual work stations. If they are not balanced properly, bottlenecks will be a problem in any assembly line. There are three basic measures for the effectiveness of the heuristics viz. (i) Line efficiency (ii) Smoothness index, and (iii) Computation time.

Only the former two measures are considered in this analysis as all the heuristics except the Hoffmann's have the same time complexity. The performance measures are defined as:

$$\text{Line efficiency (LE)} = \frac{\sum_{i=1}^{k} \text{ST}_i}{c \cdot k} \times 100$$

$$\text{Smoothness index (SI)} = \sqrt{\sum_{i=1}^{k} [(\text{ST}_{\max} - \text{ST}_i)]^2}$$

ST_i—Total station time of ith station
c—Cycle time
k—Number of work stations.

41.4 Computational Results

The line efficiency and smoothness indices are computed separately for each heuristic algorithm using the benchmark problems in addition to the reference algorithms. Higher values of efficiency and lower value of smoothing index are the indications of perfect line balancing. In all the cases, the ties are broken according to the task number, smaller first. The summary of results is presented in Table 41.5.

It is observed that the heuristic numbers three and seven perform better than others, including the benchmark algorithms in terms of the tested performance measures. When one-way AVOVA was carried out, it was observed that:

$$\text{Line efficiency} : \text{Experimental } F = 0.86 < \text{Critical } F = 2.372.$$

$$\text{Smoothness index} : \text{Experimental } F = 0.85 < \text{Critical } F = 2.372.$$

Hence, it is concluded that there is no significant difference among the heuristics. However, to confirm their relative performance, pairwise comparisons will be carried out in the extended work. The box plots for line efficiency and smoothing index obtained show that the variation in line efficiencies and smooth indices are relatively less for the weights a, $t(f/e)$ and $t(k - i)$.

Table 41.5 Summary of performance measures for different heuristics

S. No.	Heuristic	Mean efficiency	Mean smoothing index
1	Dar-El heuristic	83.5454	13.8729
2	Hoffmann's heuristic	84.8982	15.7373
3	(ta/b)	85.1931	11.2967
4	(tc/d)	80.8105	15.4748
5	(tf/e)	80.9713	16.2697
6	(th/g)	80.8105	15.6066
7	(tk/i)	85.1931	11.1094
8	($t(k-i)$)	82.0986	13.9643

After the computation, three more cases were analyzed as described below:

(i) Taking the weight as $t(a/b)+ t(k/i)$ (combination of better performing heuristics):

The results show that the mean line efficiency and smoothing index are exactly the same as that of $t(k/i)$ for each cycle time and slightly differ from that of $t(a/b)$.

$$\text{New, mean efficiency} = 85.1931\% \text{ and smoothing index} = 11.1094.$$

(ii) The order of one better performing heuristic is reversed to ascending and the measures are again computed.

In such a case, the performance decreases to 82.0986% and 15.8208 from 85.1931% and 11.1094 earlier.

(iii) The order of one average performing heuristic is reversed to ascending and the measures are again computed.

In this case, the performance increases to 81.9379% and 15.8160 from 80.9713 and 16.2697 earlier.

In the latter two cases, another observation is that the maximum number of jobs available for allotment is four as against five earlier.

41.5 Conclusion and Future Work

This paper discusses a newly proposed set of six simple heuristics based on the slope indices for the simple type—I assembly-line problems. They are tested against the benchmark data set for different cycle times. The results are compared with the popular time-tested algorithms proposed by Dar-El and Hoffmann. Two of the six

heuristics perform better than these two for the tested performance measures, line efficiency, and smoothing index.

To validate the results further, more number of data sets are to be used. Also, further improvements in simple heuristics including tie-breaking rules, different precedence strengths are to be applied for these heuristics also and the effects are to be analyzed. Only forward enumeration is considered here. Backward and bidirectional enumerations are to be implemented for further improvement in the performance.

References

1. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) Complexity of Computer Computations. Plenum Press, New York (1972)
2. Helgeson, W.P., Birnie, D.P.: Assembly line balancing using the ranked positional weight technique. J. Indus. Eng. **12**(6), 394–398 (1961)
3. Kilbridge, M.D., Wester, L.: A heuristic method of assembly line balancing. J. Indus. Eng. **12**(4), 292–298 (1961)
4. Moodie, C.L., Young, H.H.: A heuristic method of assembly line balancing for assumptions of constant or variable work element times. J. Indus. Eng. **16**(1), 23–29 (1965)
5. Hackman, S.T., Magazine, M.J., Wee, T.S.: Fast, effective algorithms for simple assembly line balancing problems. Oper. Res. **37**(6), 916–924 (1989)
6. Hoffmann, T.R.: Assembly line balancing with a precedence matrix. Manage. Sci. **9**(4), 551–562 (1963)
7. Sivasankaran, P., Shahabudeen, P.: Literature review of assembly line balancing problems. Int. J. Adv. Manuf. Technol. **73**(9–12), 1665–1694 (2014). https://doi.org/10.1007/s00170-014-5944-y
8. Rosenberg, O., Zeiger, H.: A comparison of heuristic algorithms for cost-oriented assembly line balancing. Zeitschrift für Oper. Res. **36**(6), 477–495 (1992)
9. Scholl, A.: Data of assembly line balancing problems. Techn. Hochsch., Inst. für Betriebswirtschaftslehre (1995)
10. Dar-El, E.M.: Solving large single-model assembly line balancing problem—a comparative study. AIIE Trans. **7**(3), 302–310 (1975)