MAX-MIN ANT SYSTEM FOR ASSEMBLY LINE BALANCING PROBLEM

Nuchsara Kriengkorakot , Nalin Pianthong and Rapeepan Pitakaso
Industrial Engineering Department, Faculty of Engineering,
Ubon Rajathanee University,Thailand
Email: ennuchkr@ubu.ac.th

ABSTRACT
The assembly line balancing problem (ALBP) is known as one of difficult combinatorial optimization problems. It has received a great attention over the year. In general, it consists of assigning tasks to an ordered sequence of stations such that the precedence relations among the tasks are satisfied and some performance measure is optimized.

This paper presents the simple assembly line balancing problem type I (SALBP-1; the number of workstations is minimized for a given cycle time) using Max-Min Ant System (MMAS) method to find optimal or near optimal solutions that compared with the previous published research work. The performance of this method is measured by solving a large-scale of benchmark problems that available in the literature. From the results, the computational experiments shows that the MMAS is quite effective and competitive with the other metaheuristic method for this problem.

KEY WORDS
Assembly Line Balancing Problem (ALBP), Combinatorial Optimization Problem, Max-Min Ant System, Metaheuristic

1. Introduction

Assembly line balancing is the process of allocating a set of tasks to an ordered sequence of stations in such a way that some performance measures (e.g. cycle time, number of stations, line efficiency) are optimized subject to the precedence relations among the tasks. This problem is known as the simple assembly line balancing problem (SALBP) which can be stated as follows (e.g. Baybars [4] ; Scholl and Klein [29]) ;

- A single product is manufactured in large quantities. Performing the tasks $j = 1,\ldots, n$ takes deterministic operation times t_j. The sum of all operation times is denoted by $\sum t$.
- The tasks are partially ordered by precedence relations as directed arcs. An arc (i,j) means that task i must be finished before task j can be started.

Figure 1 shows an example of a precedence diagram with n = 11 tasks an operation time as node weights.

- Each task must be assigned to exactly one station. The set of task S_k assigned to station $k = 1,\ldots,m$ are called station loads; stations are numbered consecutively along the line.
- The total operation time of tasks assigned to a station k, called station time $t(S_k)$, must not exceed the cycle time

$$t(S_k) = \sum_{j \in Sk} t_j \le c \quad k= 1,\ldots,m \qquad (1)$$

- The precedence relations must be observed. When a task j is assigned to a station k, each task i which precedes j in the precedence network must be assigned to one of stations $1,\ldots,k$.
- The objective consists of maximizing the line efficiency E which is defined as :

$$E = \sum t / (\mathrm{m} \times \mathrm{c}) \times 100\% \qquad (2)$$

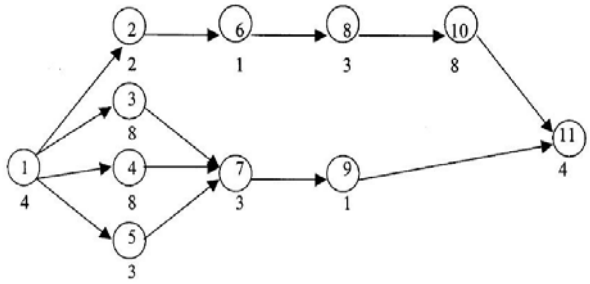

Figure 1: Precedence diagram and the task times of the Jackson's problem [16]

- Type 1 (SALBP-1) of this problem consists of assigning tasks to work stations such that the number of stations *(m)* is minimized for a given production rate (fixed cycle time, c), for a survey see Scholl and Becker [28].

2. Literature review

The first published paper of the assembly line balancing problem (ALBP) was made by Salveson [25] who suggested a linear programming solution. Since then, the topic of line balancing has been of great interest to researchers. However, since the ALB problem falls into the NP hard class of combinatorial optimization problems [12], it has consistently developed the efficient algorithms for obtaining optimal solutions. Thus numerous research efforts have been directed towards the development of computer-efficient approximation algorithms or heuristics (e.g. Kilbridge and Wester [18]; Helgeson and Birnie [14]; Hoffman,

[15]; Mansoor [21]; Arcus [1]; Baybar [3]), exact methods (e.g. Jackson [16]; Bowman [6]; Van Assche and Herroelen [35]; Mamoud [20]; Hackman et al. [13]; Sarin et al. [21]) and metaheuristics to solve the ALB problems. (e.g. Suresh and Sahu [32]; Watanabe [36]; McMullen and Frazier [23]; Chiang [7]; Bautista and Pereira [2])

In addition, with the growth of knowledge on the ALB problem, review articles are necessary to organize and summarize the finding for the researchers and practitioners. In fact, several articles (e.g. Kilbridge and Wester [19]; Mastor [22]; Johnson [17]; Talbot et al. [33]; Baybars [4]; Ghosh and Gagnon [11]; Erel and Sarin [10]) have reviewed the work published on this problem. Recently two articles by Scholl and Becker [28]; Becker and Scholl [5] provide the state-of-the-art exact and heuristic solution procedures for simple assembly line balancing (SALB) and a survey on problems and methods in generalized assembly line balancing (GALB) respectively.

In this paper, we presents the simple assembly line balancing problem type I (SALBP-1; the number of workstations is minimized for a given cycle time) using Max-Min Ant System (MMAS) method to find optimal or near optimal solutions that compared with the previous published research work. The performance of the MMAS method is measured by solving a large-scale of benchmark problems that available in the literature.

3. Max-Min Ant System

Ant Colony Optimization (ACO) is a metaheuristic using artificial ants to find desirable solutions in difficult combinatorial optimization problem. The behavior of artificial ants is based on the trails of real ants, plus additional capacities that make them more effective, such as a memory of past actions. Each ant of the "colony" builds a solution to the problem under consideration, and uses information collected on the problem characteristics and its own performance to change how other ants see the problem. The first ACO algorithm, called Ant System (AS), as well as many of the ACO algorithms proposed subsequently, was first tested on the Traveling Salesman Problem (TSP). A more detailed explanation of ACO can be found in [8].

The basic ant system is conceptually simple. It can be summarized by the following pseudocode : [9]

```
While it < max_it do ,
        for each ant do ,
                build_solution ( ) ;
                update_pheromone ( ) ;
        endfor
endwhile
```

An improvement to the Ant System, called Max-Min Ant system (MMAS), was introduced in Stützle and Hoos [30]. On this implementation, the pheromone trail is updated only on the global best and/or local best solutions, instead of on solutions created by every ants, thus promoting a better exploitation of the search space. Another peculiarity is the inclusion of upper and lower bounds to the pheromone level ($\tau_{\max}$ and $\tau_{\min}$) , thus helping to avoid stagnation. Initially all trails are set to the upper bound in order to favor exploration.

3.1 solution generation

The majority of building procedures for solving of assembly line balancing problems are based on the application of priority rule. An in-depth study of the available procedures can be found in Scholl [27]. The priority rule is based on the building of feasible solutions for SALBP-1 instances. Each task is assigned a priority value that depends on its process time and its precedence relations.

In this study, we use station oriented procedure for solution building. This procedure is initiated by the opening of a first station (k = 1). Tasks are then successively assigned to this station until more tasks cannot be assigned, in which case, said station is closed and a new station is opened. In each iteration, the candidate task with the greatest priority is assigned to the current station; a task is a candidate when its preceding tasks have been assigned and it requires less time that available in the station under construction. When no more tasks may be assigned to the open station, this is closed and the following station (k + 1) is opened. The procedure finalizes when there are no more tasks left to assign.

For the building procedures, a priority rule by using the duration of a task is employed. This enables the MMAS to be used with visibility values similar for all the instances solved.

In general, the heuristic weight of a task, j, η_j , is equal to

$$\eta_j = t_j \qquad (3)$$

In the case of using a probabilistic building schema, as is characteristic of MMAS, it is necessary to modify the station-oriented procedure described above. The probabilistic selection criterion must depend on the heuristic information of the task, η_j , and on the information provided by the pheromone deposited by previous solutions, τ_{kj} . The pheromone may be used in directly, relating a task, j, with the station under construction, k. In this case of direct evaluation, the likelihood of selecting candidate task j belonging to the set S_k of candidate tasks to be assigned to station k, is determined in the following way:

$$p_{kj} = \frac{\left[\tau_{kj}\right]^{\alpha}\left[\eta_{j}\right]^{\beta}}{\sum_{i \in S_k}\left[\tau_{ki}\right]^{\alpha}\left[\eta_{i}\right]^{\beta}} \qquad (4)$$

where τ_{kj} and τ_{ki} correspond to the pheromone deposited by previous solutions and η_j , η_i to the heuristic weight of each task. The parameters α and β are user-defined and control the relative weight of trail intensity, τ_{kj} and heuristic information, η_j .

3.2 Pheromone update

In MMAS only one single ant is used to update the pheromone trails after each iteration[31]. Consequently, the modified pheromone trail update rule is given by

$$\tau_{kj}(t+1) = (1-\rho)\tau_{kj}(t) + \Delta\tau_{kj}^{best} \qquad (5)$$

where $0 \le \rho < 1$ is the pheromone evaporation rate and $\Delta\tau_{kj}^{best} = 1/f(s^{best})$ and $f(s^{best})$ denotes the solution of either the iteration-best (s^{ib}) or the global-best solution (s^{gb}) , MMAS focuses on the use of the iteration-best solutions.

3.3 Pheromone trail limits

In MMAS , lower and upper limit $\tau_{\min}$ and $\tau_{\max}$ on the possible pheromone values on any arc are imposed in order to avoid search stagnation such that for all pheromone trails $\tau_{kj}(t)$, $\tau_{\min} \le \tau_{kj}(t) \le \tau_{\max}$. After each iteration one has to ensure that the pheromone trail respects the limits. If we have $\tau_{kj}(t) > \tau_{\max}$, we set $\tau_{kj}(t) = \tau_{\max}$; analogously, if $\tau_{kj}(t) < \tau_{\min}$, we set $\tau_{kj}(t) = \tau_{\min}$. Also note that by enforcing $\tau_{\min} > 0$. Initially all trail are set to the upper bound in order to favor exploration. As in [31], the upper bound is usually chosen to be

$$\tau_{\max} = \frac{1}{1-\rho} \cdot \frac{1}{f(s^{opt})} \qquad (6)$$

Where $f(s^{opt})$ is the optimal solution value for a specific problem and the lower bound is set to

$$\tau_{\min} = \frac{\tau_{\max}(1-\sqrt[n]{0.05})}{((n/2)-1)\sqrt[n]{0.05}} \qquad (7)$$

The pheromone bounds are taken from [31,24].

4. Computational results

In order to test the performance of our MMAS method with respect to Chiang's best implementation [7], (i.e. best improvement without task aggregation) we have solved the 24 instances of problem sets (up to 111 tasks) and the optimal solutions for these problems can be found in [34]. The problems were solved on a personal computer using FoxPro 6.0 with a Pentium 4, 3.0 GHz, 512 MB RAM and the operating system on windows XP.
The best set of control parameters was previously determined, as well as the best way of reading the pheromone, obtaining as the best combination among those tested values of α =1, β = 2 to 5, ρ = 0.05. $\tau_{\min}$ and $\tau_{\max}$ were set to 0.01 and 1, respectively and the pheromone trial was initialized to 1, imposing two stopping conditions on the MMAS : a maximum computation time of 60 seconds or the obtainment of the optimum solution.

Table 1 presents the performance comparison between Chiang' TS [7], and the proposed MMAS method for assembly line balancing. The first four columns show the characteristics of each instances, i.e problems, size, cycle time and optimal solutions (m*). Then the experimental results obtained respectively by the TS in [7] and by our method (MMAS) are reported. The minimum number of stations is listed under heading m and the calculation time under cal. time. All times are reported in seconds but, since they correspond to different machines and programming languages (the code in [7] is written in C and executed on a VAX 6420 mainframe) care should be taken when comparing them.

As seen by the results, the MMAS found all 24 optimal solutions whereas Chiang's best algorithm reached only 21 out of 24. The computational results indicated that the MMAS was able to identify better solutions than the Chiang's TS on 3 out of 24 instances and take less than thirty seconds in each instance problems (this take about 8.85 sec. on average). Considering the average relative deviations from the optimal solution, the ranking of MMAS was better than Chiang's TS.

5. Conclusion

This paper addresses the simple assembly line balancing problem type I (SALBP-1; the number of workstations is minimized for a given cycle time) using Max-Min Ant System (MMAS) method to find optimal or near optimal solutions that compared with Chiang' TS [7]. The performance of this method is measured by solving a large-scale of benchmark problems (up to 111 tasks) that available in the literature. From the results, the computational experiments indicates that the MMAS is quite effective and able to identify better solutions than the Chiang's TS for the test problems.

Future research will focus on the application of the MMAS to the multiple assembly line balancing problem and the U-shaped line balancing problem in the just-in – time environment.

Acknowledgements

We would like to thank the faculty of Engineering, Ubon Rajathanee University for the financial support.

Table 1: Performance Comparison between Chiang' TS and the Proposed MMAS Method for Assembly Line Balancing

Problems	Size	Cycle time	Optimal solution	Chiang [1]			MMAS		
			m*	m	%	Cal. Time (sec.)	m	%	Cal. Time (sec.)
Kilbridge & Wester	45	57	10	10	0	7.15	10	0	2.63
		79	7	7	0	6.35	7	0	21.37
		92	6	6	0	9.06	6	0	24.72
		110	6	6	0	3.41	6	0	2.93
		138	4	4	0	8.18	4	0	26.18
		184	3	3	0	9.11	3	0	3.33
Tonge	70	176	21	21	0	8.06	21	0	3.82
		364	10	11	10	5.82	10	0	4.39
		410	9	9	0	7.17	9	0	4.78
		468	8	8	0	7.23	8	0	6.58
		527	7	7	0	6.04	7	0	7.66
Arcus 1	83	5048	16	17	6.25	9.91	16	0	4.96
		5853	14	14	0	46.62	14	0	5.53
		6842	12	12	0	15.09	12	0	5.48
		7571	11	11	0	6.77	11	0	7.11
		8412	10	10	0	6.86	10	0	7.91
		8898	9	9	0	61.09	9	0	7.52
		10816	8	8	0	27.90	8	0	10.05
Arcus 2	111	5755	27	27	0	70.50	27	0	8.41
		8847	18	19	5.55	63.21	18	0	7.60
		10027	16	16	0	65.82	16	0	10.21
		10743	15	15	0	54.46	15	0	8.97
		11378	14	14	0	38.11	14	0	9.72
		17067	9	9	0	73.95	9	0	10.63
Total optimal solutions found from 24 instances problem				**21**	0.91 /inst.	25.74 /inst.	**24**	0 /inst.	8.85 /inst.

[1] Best implementation of Tabu Search, researched by Chiang [7].
m is the number of stations (**m*** is the optimal solution).
% is the average relative deviation from the optimal solution.

References

[1] Arcus A.L., COMSOAL: A Computer Method of Sequencing Operations for Assembly Lines. *International Journal of Production Research, 4*(4), 1966, 259-277.

[2] Bautista, J. and Pereira, J., Ant algorithms for a time and space constrained assembly line balancing problem. *European Journal of Operational Research*, *177* (3), 2007, 2016-2032.

[3] Baybar, I., An Efficient Heuristic Method for the Simple Assembly Line Balancing Problem. *International_Journal of Production Research, 24*(1), 1986a, 149-166.

[4] Baybars, I., A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem. *Management Science, 32*(8), 1986b, 909-932.

[5] Becker, C. and A. Scholl., A Survey on Problems and Methods in Generalized Assembly Line Balancing. *European Journal of Operational Research, 168*(3), 2006, 694-715.

[6] Bowman, E.H., Assembly Line Balancing by Linear Programming. *Operations Research, 8*(3), 1960, 385-389.
[7] Chiang, W.C., The application of a tabu search metaheuristic to the assembly line balancing problem. *Annals of Operations Research, 77*, 1998, 209-227.
[8] Dorigo, M. and Stützle, T., *Ant Colony Optimization* (The MIT Press, Massachusetts: Bradford, 2004).
[9] Dorigo, M., Optimization, Learning and Natural Algorithms. *Ph.D. Thesis*, Politecnico di Milano, Italy, 1992.
[10] Erel E., I. Sabuncuoglu and B.A. Aksu, Balancing of U-type Assembly Systems Using Simulated Annealing. *International Journal of Production Research, 39*(13), 2001, 3003-3015.
[11] Ghosh S. and R. J. Gagnon., A Comprehensive Literature Review and Analysis of the Design, Balancing and Scheduling of Assembly Systems. *International Journal of Production Research, 27*(4), 1989, 637-670.
[12] Gutjahr, A. L., Nemhauser, G.L., An algorithm for the line balancing problem. *Management Science, 11*(2), 1964, 308-315.
[13] Hackman, S.T., M. J. Magazine and T. S. Wee., Fast, Effective Algorithms for Simple Assembly Line Balancing Problems, *Operations Research, 37*(6), 1989, 916-924.
[14] Helgeson, W.P. and Birnie, D.P., Assembly Line Balancing Using the Ranked Positional Weight-Technique, *The Journal of Industrial Engineering, 12*(6), 1961, 394-398.
[15] Hoffmann,T.R., Assembly Line Balancing with a Precedence Matrix, *Management Science*, *9*(4), 1963, 551-562.
[16] Jackson, J.R., A Computing Procedure for a Line Balancing Problem, *Management Science*, *2*(3) , 1956, 261-271.
[17] Johnson, R.V., Assembly Line Balancing Algorithms: Computation Comparisons, *International Journal of Production Research, 19*(3), 1981, 277-287.
[18] Kilbridge, M.D. and Wester, L., A Heuristic Method of Assembly Line Balancing, *The Journal of Industrial Engineering, 12*(4), 1961, 292-298.
[19] Kilbridge, M.D. and Wester, L., A review of analytical systems of line balancing, *Operations Research, 10*(5), 1962, 626-638.
[20] Mamoud, K.I., A Generalised Assembly Line Balancing Algorithm. *PhD. Dissertation*, University of Bradford, UK., 1989, 489 pages.
[21] Mansoor, E.M., Assembly Line Balancing -An Improvement on the Ranked Positional Weight Technique, *The Journal of Industrial Engineering, 15*(2), 1964, 73-77.
[22] Mastor, A.A., An Experimental Investigation and Comparative Evaluation of Production Line Balancing Techniques, *Management Science, 16*(11), 1970, 728-746.
[23] McMullen, P.R. and G.V. Frazier, Using Simulated Annealing to Solve a Multiobjective Assembly Line Balancing Problem with parallel workstations, *International Journal of Production Research, 36*(10), 1998, 2717-2741.
[24] Rapeepan Pitakaso, Christian Almeder, Karl F. Doerner and Richard F. Hartl**.,** A MAX-MIN ant system for unconstrained multi-level lot-sizing problems. *Computers & Operations Research*, *34*(9), 2006, 2533-2552.
[25] Salveson, M.E., The Assembly Line Balancing Problem, *The Journal of Industrial Engineering, 6*(3), 1955, 18-25.
[26] Sarin, S.C., E. Erel and E.M. Dar-El., A Methodology for Solving Single-Model, Stochastic Assembly Line Balancing Problem, *Omega*, *27*, 1999, 525-535.
[27] Scholl, A., *Balancing and sequencing of assembly lines* (Physica-Verlag, Heidelberg.,1999).
[28] Scholl, A. and Becker, C., State-of-the-art Exact and Heuristic Solution Procedures for Simple Assembly Line Balancing, *European Journal of Operational Research, 168*(3), 2006, 666-693.
[29] Scholl, A., and Klein , R., ULINO-Optimally balancing U-shaped JIT assembly line. *International Journal of Production Research, 37*(4), 1999, 721-736.
[30] Stützle, T. and Hoos, H.H., The MAX-MIN ant system and local search for the traveling salesman problem, *Proceedings of the IEEE International Conference of Evolutionary Computation (ICEC'97),* IEEE Press, Piscataway, NJ, USA,1997, 309-314.
[31] Stützle, T. and Hoos, H.H., MAX-MIN ant system, *Future Generation Computer Systems*, *16*, 2000, 889-914.
[32] Suresh G. and S. Sahu., Stochastic Assembly Line Balancing Using Simulated Annealing, *International Journal of Production Research*, *32*(8), 1994, 1801-1810.
[33] Talbot, F. B., J.H. Patterson and W.V. Gehrlein., A Comparative Evaluation of Heuristic Line Balancing Techniques, *Management Science, 32*(4), 1986, 430-454.
[34] Talbot, F. B. and J.H. Patterson, An integer programming algorithm with network cuts for solving the assembly line balancing problem. *Management Science, 30*, 1984, 85-99.
[35] Van Assche, F. and Herroelen, W.S., An Optimal Procedure for the Single-Model Deterministic Assembly Line Balancing Problem, *European Journal of Operations Research, 3*, 1978, 142-149.
[36] Watanabe T., Y. Hashimoto, I. Nishikawa and H. Tokumaru, Line Balancing Using A Genetic Evolution Model, *Control Engineering Practice, 3*(1), 1995, 69-76.