An Adaptive Ant Colony Optimization for Solving Assembly Line Balancing Problem

Fuping Deng[1], Chaoyong Zhang[1], Kunlei Lian[1], and Shaotan Xu[1]

[1]State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan, 430074

Abstract

An adaptive ant colony optimization was proposed to solve the assembly line balancing problem (ALBP). According to the characteristics of the ALBP, a method of solution constructing strategy was developed, and a better differentiation of objective function was proposed to appraisal solution quality. However, general ant algorithm often falls into local optimal and consume excessive time, in order to overcome these shortcoming, an improved ACO was presented by adaptive adjustment of the parameters in the algorithm, which has a good ability of searching better solution at higher convergence speed. Finally, the proposed algorithm was tested and compared against best known algorithms reported in the literatures, and the experimental results indicate the feasibility and effectiveness of the proposed algorithm.

Key words: assembly line balancing; ant colony optimization (ACO); adaptive; artificial intelligence

1 Introduction

Assembly line is a widely used component of manufacturing factories and it faces many problems in the design and operation of assembly lines, assembly line balancing problem (ALBP) being one of most important ones. A small improvement towards ALBP often leads to significant efficiency enhancement and cost reduction (Salveso, 1955); On the other hand, ALBP is a classical NP-hard combinatorial optimization problem in which the complexity increases exponentially with more number of jobs and there yet exists polynomial-time algorithms to find optimization solutions for this problem (Kilincci, 2010). Currently, there exist three solution strategies to solve ALBP: exact algorithms, heuristic algorithms and artificial intelligence algorithms. Exact algorithms are able to find the optimal solutions, for only small-sized problems and with tremendous computation times, therefore, they can hardly be applied in real-world production systems (Peeters and Degraeve, 2006; Scholl and Becker, 2006). Heuristic algorithms have received many attentions from researchers due to its theoretical simplicity; on the other hand, they generally take long time to identify the optimal solution and it is often hard to verify the solutions they found are optimal (Ponnambalam et al., 1999). In recent years, artificial intelligence algorithms, including genetic algorithms, simulated annealing and tabu search, witness significant advances in the fields and have been used to solve ALBP successfully (O. et al., 2011; Ozcan and Toklu, 2008).

Ant colony optimization (ACO) is another intelligent optimization algorithm proposed proposed by Clolrni (Colorni et al., 1991) in 1991 and has been applied to various combinatorial optimization problems. Bautista and Pereira (2002) made the first attempt to solve a simple assembly line balancing problem using ant colony algorithms based on ant system and the optimization results were not optimal. McMullen and Tarasewich (2003) obtained superior performance of ACO in solving the assembly balancing problem with multiple job types, stochastic processing times and parallel workstations. Bautista and Pereira (2007) studied an assembly balancing problem with timing and spatial constraints using ant colony algorithms.

The aforementioned researches in assembly balancing problems using ACO suffer from inferior performance when compared with other algorithms and the reasons are twofold. First, the way the pheromone is accumulated in some algorithms is too simplified, which prevents ACO finding optimal solutions. Second, the objective function considers only limited factors, which makes it difficult to differentiate good solutions from bad solutions. To address this, we propose an adaptive ant colony algorithm to solve the assembly line balancing problems. It tries to avoid local optima by utilizing both external and historical information to dynamically adjust global pheromone evaporation factor in the process of path contruction of an ant. In addition, the algorithm incorporates balancing and smoothing as part of the objective function, which improves ACO's ability in differentiating solutions. The superirority of the proposed algorithm is validated on benchmark problem instances.

2 Assembly Line Balancing and Mathematical Model

ALBP refers to the assignment of finite job set to finite workstation set in order to maximize workstation utilization, minimize overall overload time and minimize balancing objective value, subject to processing constraints and workstation processing time satisfying cycling requirements. It involves the coordination of various processes within an assembly line and needs to address the inconsistency in process machining times. Assembly line productivity as well as product quality are both greatly affected by its balancing level.

The ALBP this paper aims to address can be mathematically described as follows:

$$\text{max.} \quad \lambda L - I \tag{1}$$

$$\text{s.t.} \quad L = \frac{\sum_{i=1}^{n} t_i}{mC} \tag{2}$$

$$I = \sqrt{\frac{\sum_{k=1}^{m}(max(T(S_k)) - T(S_k))^2}{m}} \tag{3}$$

$$S_i \cap S_j = \emptyset,\ i, j = \{1, 2, \cdots\}, i \neq j \tag{4}$$

$$\cup_k S_k = E,\ \forall i \in S_x, j \in S_y, 1 \leq x, y \leq n, x \leq y \text{ if } P_{ij} = 1 \tag{5}$$

$$T(S_k) \leq C \tag{6}$$

In this model, L is the balancing rate of an assembly line; I is the smoothing factor; λ is an user-defined parameter and $\lambda > 1$; E is the set of jobs within the assembly line; S_k is the set of jobs assigned to workstation k; C is the assembly line cycle; t_i is the processing time of job i; $T(S_k)$ is the total processing time of workstation k; P is the precedence matrix of ALBP and $P = [P_{ij}]_{n\times n}$, $P_{ij} = 1$ job i must be processed right before job j, 0 otherwise.

3 An Adaptive ACO for the ALBP

3.1 Solution construction strategy

In order to use ACO to solve the ALBP, the processing jobs can be seen as nodes on a graph which will be traversed by ants, also the connections between jobs and workstations can be seen as arcs on the graph. The assignmnet of processing jobs to workstations can then be seen as an ant colony travels through the graph with the guidance of pheromone and heuristic information.

The solution construction is a key step in emplying ACO to solve ALBP and this paper constructs a feasible solution by gradually assigning processing jobs to corresponding workstations. To this end, we define the following notations:

- no-assigned task: a task that hasn't been assigned to any workstation yet
- available task: a task that hasn't yet been assigned but satisfy precedence constraints between tasks, also all of its preceeding tasks have been assigned
- assignable task: an available task that satisfies cycling constraints.

The feasible solution construction algorithm can then be described as follows:

1. open a workstation
2. identify the set of available tasks from all no-assigned tasks and the precedence constraints among tasks, if the resulting set is empty, go to step 7
3. identify the set of assignable tasks from available tasks and the cycling constraint
4. if the set of assinable tasks is empty, go to step 6
5. select a task from the set of assinable tasks according to defined rules and assign it to the current workstation, go to step 2
6. open a new workstation, go to step 3
7. stop

Using the way defined in the above algorithm to identify the set of assignable tasks, there always exists an optimal assinable task in the set.

One key characteristic of ACO is its utilization of pheromone feedback and heuristic information during its search for global optimality; therefore, the way of pheromone updating and heuristic information selection have a huge impact on the performance of ACO. In this paper, we define τ_{ij} as the pheromone intensity on arc (i, j) traversed by ants and represents the expectation of assigning task i to workstation j. The corrresponding heuristic informaiton is computed by static precedence rules. In addition, the heuristic information of ACO consists of the maximal task completion time and maximal number of succedding tasks, and the visibility η_i of task i can be computed as

$$\eta_i = \frac{t_i}{C} + \frac{U_i}{max_{i=1,2,\cdots,N} U_i} \tag{7}$$

where U_i is the number of succedding tasks of task i.

3.2 Objective function

3.3 Pheromone update strategy

References

Joaquin Bautista and Jordi Pereira. Ant algorithms for assembly line balancing. In Marco Dorigo, Gianni Di Caro, and Michael Sampels, editors, *Ant Algorithms*, pages 65–75, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45724-4.

Joaqun Bautista and Jordi Pereira. Ant algorithms for a time and space constrained assembly line balancing problem. *European Journal of Operational Research*, 177(3):2016 – 2032, 2007. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2005.12.017. URL http://www.sciencedirect.com/science/article/pii/S0377221705008490.

Alberto Colorni, Marco Dorigo, and Vittorio Maniezzo. Distributed optimization by ant colonies. 01 1991.

Ozcan Kilincci. A petri net-based heuristic for simple assembly line balancing problem of type 2. *The International Journal of Advanced Manufacturing Technology*, 46(1):329–338, Jan 2010. ISSN 1433-3015. doi: 10.1007/s00170-009-2082-z. URL https://doi.org/10.1007/s00170-009-2082-z.

Patrick R. McMullen and Peter Tarasewich. Using ant techniques to solve the assembly line balancing problem. *IIE Transactions*, 35(7):605–617, 2003. doi: 10.1080/07408170304354. URL https://doi.org/10.1080/07408170304354.

Ugur O., Talip K., and Bilal T. A genetic algorithm for the stochastic mixed-model u-line balancing and sequencing problem. *International Journal of Production Research*, 49(6):1605–1626, 2011.

U. Ozcan and B. Toklu. A tabu search algorithm for two-sided assembly line balancing. *The International Journal of Advanced Manufacturing Technology*, 43(7):822, Sep 2008. ISSN 1433-3015. doi: 10.1007/s00170-008-1753-5. URL https://doi.org/10.1007/s00170-008-1753-5.

Marc Peeters and Zeger Degraeve. An linear programming based lower bound for the simple assembly line balancing problem. *European Journal of Operational Research*, 168(3): 716 – 731, 2006. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2004.07.024. URL http://www.sciencedirect.com/science/article/pii/S0377221704004813. Balancing Assembly and Transfer lines.

S. G. Ponnambalam, P. Aravindan, and G. Mogileeswar Naidu. A comparative evaluation of assembly line balancing heuristics. *The International Journal of Advanced Manufacturing Technology*, 15(8):577–586, Jul 1999. ISSN 1433-3015. doi: 10.1007/s001700050105. URL https://doi.org/10.1007/s001700050105.

M. E. Salveso. The assembly line balancing problem. *The Journal of Industrial Engineering*, 6(3):1–25, 1955. URL https://ci.nii.ac.jp/naid/10003095017/en/.

Armin Scholl and Christian Becker. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168(3):666 – 693, 2006. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2004.07.022. URL http://www.sciencedirect.com/science/article/pii/S0377221704004795. Balancing Assembly and Transfer lines.