Applied Soft Computing 11 (2011) 3186–3198

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Multiple-colony ant algorithm for parallel assembly line balancing problem

Lale Ozbakir[a], Adil Baykasoglu[b,*], Beyza Gorkemli[a], Latife Gorkemli[a]

[a] Erciyes University, Faculty of Engineering, Department of Industrial Engineering, Kayseri, Turkey
[b] University of Gaziantep, Faculty of Engineering, Department of Industrial Engineering, Gaziantep, Turkey

ARTICLE INFO

Article history:
Received 16 June 2009
Received in revised form 26 July 2010
Accepted 12 December 2010
Available online 17 December 2010

Keywords:
Line balancing
Parallel assembly lines
Ant colony optimization
Meta-heuristics

ABSTRACT

Assembly lines are designed as flow oriented production systems which perform operations on standardized products in a serial manner. Balancing of assembly lines is one of the most important problems among the other problems of assembly lines like designing and managing. In today's highly competitive manufacturing environment increasing system flexibility, reducing failure sensitivity, improving system balance and productivity are crucial. Parallel assembly lines provide some opportunities in improving these objectives especially when the capacity of production system is insufficient. Unlike the traditional assembly lines there are a few studies on balancing parallel assembly lines in the present literature. Parallel assembly line balancing is a NP-hard problem similar to other assembly lines. In this paper, a novel multiple-colony ant algorithm is developed for balancing bi-objective parallel assembly lines. The proposed algorithm is also one of the first attempts in modeling and solving the present problem with swarm intelligence based meta-heuristics. The proposed approach is extensively tested on the benchmark problems and performance of the approach is compared with existing algorithms. It is shown that the proposed approach is very effective.

1. Introduction

Assembly lines consist of serial workstations connected with conveyor belts or other material handling systems to produce high quantity standardized products. Operations of the work pieces are performed by highly specialized operators on these workstations repeatedly. The main purpose of such systems is to reduce the costs for mass production of standardized products. However, the requirements of today's manufacturing systems are changing as customer needs diversifying. Increasing product variety causes to take over the concept of assembly to order. This means that the highly efficient and flexible assembly lines must be designed and balanced in order to produce different products. As discussed by some authors efficiency of assembly lines can be further improved by considering neighboring lines together while balancing them [1–3]. In order to reduce failure sensitivity, increase flexibility in terms of capacity and product variety balancing parallel lines together is recently attracting the attention of researchers in industry and academia. Traditionally straight assembly lines are balanced separately by assigning operations to workstations considering the precedence relations and cycle time to minimize the number of workstations. This is the most frequently studied assembly line balancing problem in the literature (SALBP). In addition to the well known simple assembly lines, researchers and practitioners developed several other assembly line designs and balancing techniques such as "mixed and multi model lines", "U shaped lines", "assembly line with parallel workstations or tasks", etc. For an excellent and detailed explanation and classification of assembly lines with discussions on existing solution procedures refer to Boysen et al. [4,5]. A survey is also presented by Lusa [6] in terms of parallel assembly lines. This paper is a good survey presenting different types of parallel assembly lines along with discussions on their main advantages and disadvantages. Parallel Assembly Line Balancing Problem (PALBP) has initially been considered by Gokcen et al. [1] in the literature. Scholl and Boysen [2] presented a detailed definition of PALBP. They entitled this problem as Multiproduct Parallel Assembly Line Balancing Problem (MPALBP) and divided it into two connected sub-problems; assignment of the products to parallel lines and balancing the lines simultaneously.

It has been proven in the literature that meta-heuristic approaches are quite effective in handling NP-hard problems [7]. Ant Colony Optimization (ACO), which is one of the swarm intelligence based meta-heuristic algorithms, is successfully applied to several other types of assembly line balancing problems in the literature. Bautista and Pereira [8] applied ACO to time and space constrained assembly lines. Baykasoglu and Dereli [9] proposed an algorithm based on ACO for two sided assembly line balancing problem. McMullen and Tarasewich [10] employed ACO for solving the multi-objective assembly line balancing problem. Sabuncuoglu

* Corresponding author. Tel.: +90 3423604383; fax: +90 3423604383.
E-mail addresses: baykasoglu@gantep.edu.tr, baykasoglu@gmail.com (A. Baykasoglu).

1568-4946/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2010.12.021

et al. [11] developed an ACO algorithm for U-type assembly line balancing problem. Vilarinho and Simaria [12] proposed an ACO based balancing procedure for mixed-model assembly lines with parallel workstations. Simaria and Vilarinho [13] developed an ACO algorithm for two sided assembly line balancing problems. Based on the motivation of the previous successful applications and encouraging results of our preliminary studies with ACO for PALBP [14] we decided to develop a new and improved ACO algorithm which also extends our previous work. The proposed ACO algorithm is a multiple colony type and multiple criteria are considered in evaluating the performance of the generated solutions. The remainder of this paper is organized as follows: PALBP is explained in Section 2. The proposed multiple colony ant algorithm and components of this algorithm is presented in Section 3. In Section 4, the performance of the proposed algorithm is tested and compared with the published results. Finally, the conclusions are given in Section 5.

2. Problem definition

PALBP consists of a number of serial assembly lines arranged in a parallel form. Different products and/or models can be assigned to different serial lines by considering a joint cycle time. Although the concept of parallel assembly lines is known in the industry and academia [15], there are just a few research papers on their balancing in the existing literature. The problem is first introduced by Gokcen et al. [1]. They proposed several procedures and a mathematical programming model for assembly line balancing problem which contains parallel lines. The objective of their study was to minimize the number of stations while balancing more than one assembly line together. Ozcan et al. [16] developed a multi-objective tabu search algorithm for solving PALBP with the aim of maximizing line efficiency and minimizing variation of workloads. Benzer et al. [17] proposed a network model for PALBP. Their model is based on the shortest route formulation which was developed for single model assembly line. Kara et al. [18] proposed a fuzzy goal programming model that can be used for balancing parallel assembly lines. Scholl and Boysen [2] modeled the problem as a binary linear programming problem and proposed an exact solution procedure based on the branch and bound algorithm. In a later work, Scholl and Boysen [3] mentioned that the original objective function (*minimizing number of workstations alone*) seems not totally sufficient for obtaining a competitive line design as it might lead to inefficiencies (i.e. longer lines). They advised to use the term "workplace" instead of "workstation" and defined the "split workplace" concept as a place where only directly opposite workstations can be linked to form a workplace where workers can also work in parallel (back to back). They tried to achieve this by defining the property of active multi-line balances and adding a second-order term to the original objective function which might results in a more compact multi-line layout. However, we should mention here that allowing workers to work in parallel (back to back) may reduce the total line length but this in turn will necessitate more space between the parallel lines in most of the applications. This also means more walking distance for an operator who is working on two lines.

There are two basic sub-problems in PALBP; matching models/products with lines and balancing adjacent lines together. Traditionally, each line is balanced separately considering the precedence relations between the tasks and cycle time in order to minimize the total number of workstations. But, parallel design of lines can offer a reduced number of workstations by taking into account the neighboring lines. If neighboring lines are settled in such a way that a single worker can operate in both directions then it is possible to reduce the number of workstations. In other words, the main purpose of PALBP is to balance multiple serial assembly lines simultaneously. This is achieved by assigning task(s) from several neighboring lines to a multi-skilled operator. Such a design with two parallel lines is schematically represented in Fig. 1, workers 1 and 3 can operate on both neighboring lines (*in* Fig. 1, *1st operation of product one and, 1st and 2nd operations of product two are assigned to worker one*).

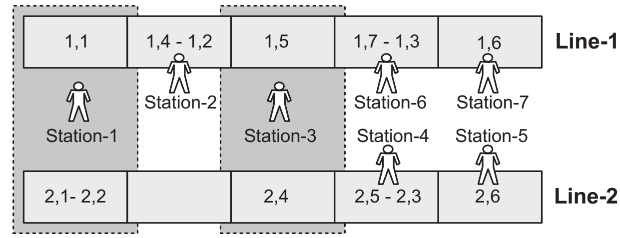

Fig. 1. Schematic representation of a typical parallel assembly line.

Similar to previous studies, the following assumptions are made in this study for modeling and solving the present PALBP [1–3]:

- Precedence diagrams of each model produced on each line are known.
- Cycles times of each line are known and identical.
- The task times of each product are known.
- The number of parallel lines is defined as equal to number of products.
- Operators are multi-skilled.
- Operator transfer times between two adjacent lines are ignored.

3. Development of multiple-colony ant algorithm for PALBP

One of the most efficient meta-heuristic algorithms on combinatorial optimization problems is known as ACO. This is especially true if the problem has an inherently network structure [19]. The ACO meta-heuristic mimics the foraging behavior of natural ant colonies which share the central information called pheromone trail. Observation of ants in nature shows that ants are very successful in finding shortest path between food sources and their nest by depositing pheromone and choosing their way by using the pheromone concentration. The probability of choosing paths is increased by having strong pheromone concentration [20]. There are two main parts in ant colony algorithms; solution construction and pheromone update. In the first stage ants select one operation from a candidate list until a complete solution is obtained. In the second stage ants deposit the pheromone on the way of its completed solution. "Ant density", "ant quantity" and "ant cycle" are the basic ant algorithms which have different pheromone trail update strategies. In this study "ant cycle" algorithm which implies that ants deposit pheromone after they have built a complete tour is employed. The basic ant cycle steps are as follows [21]:

```
Initialize parameters and pheromone trails
Until the termination criteria is satisfied
  For each ant in the colony
  Build a new solution according to the pheromone trail and heuristic information
Update the pheromone trail
```

The main idea behind the multiple-colony ant algorithm is due to ACO's suitable structure for parallelization. In the present study, each colony is located on different processors and during search process they exchange information within predetermined periods. Thus the main purpose of the multiple-colony ant algorithms is to search different regions of search space by the colonies cooperating to find good solutions by information exchange. In developing the present multiple-colony ant algorithm we motivated from the Udomsakdigool and Kachitvichyanukul's [22] study. They developed a multiple-colony ant algorithm which consists of multiple