Computers & Operations Research 44 (2014) 105–114

Contents lists available at ScienceDirect

Computers & Operations Research

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

A branch-and-bound algorithm for assembly line worker assignment and balancing problems

Mariona Vilà [1], Jordi Pereira [1,2,*]

[1] Escola Universitària d'Enginyeria Tècnica Industrial de Barcelona (EUETIB), Universitat Politècnica de Catalunya, C. Comte Urgell, 187 1st Floor, 08036 Barcelona, Spain
[2] Departamento de Ingeniería Industrial, Universidad Católica del Norte, Av. Angamos 0610, Antofagasta, Chile

ARTICLE INFO

Available online 7 November 2013

Keywords:
Branch-and-bound
Manufacturing
Assembly line balancing

ABSTRACT

In this paper, we studied the assembly line worker assignment and balancing problem, which is an extension of the classical assembly line balancing problem in which an optimal partition of the assembly work among the stations is sought along with the assignment of the operators to the stations. The relationship between this problem and several other well-studied problems is explored, and new lower bounds are derived. Additionally, an exact enumeration algorithm, which makes use of the lower bounds, is developed to solve the problem. The algorithm is tested by using a standard benchmark set of instances. The results show that the algorithm improves upon the best-performing methods from the literature in terms of solution quality, and verifies more optimal solutions than the other available exact methods.

1. Introduction

An assembly line is a manufacturing process that is used in the mass production of standardized products, such as automobiles. An assembly line is usually composed of several stations arranged in a serial fashion and linked together by a material handling system, such as a conveyor belt. Units of product are consecutively launched down the line and are moved from station to station by the material handling system. Each station is allotted an identical time, known as the cycle time, to perform one or more tasks on each product.

The assembly line balancing problem (ALBP) is the optimization problem of optimally partitioning (balancing) the assembly work among the stations according to some objective. The ALBP is a widely studied problem whose basic formulation is known as the simple assembly line balancing problem (SALBP), see [1] for a recent review. The SALBP is concerned with the assignment of tasks to stations in such a way that precedence relations between tasks (e.g., car seats must be assembled before doors in a car) are fulfilled and the line efficiency is maximized. Maximizing the line efficiency is equivalent to minimizing the total idle time, which is measured as the difference between the cycle time and the workload of each station (a station workload is defined by the sum of the operation times of the tasks that it performs). This objective can be achieved by minimizing the number of stations for a given cycle time (type-1 objective), by minimizing the cycle time for a given number of stations (type-2 objective) or by minimizing the product of the cycle time and the number of stations (type-E objective). There also exists a feasibility version known as the type-F problem, in which both the cycle time and the number of stations are known, and the objective is to find a feasible solution. The optimization versions of the SALBP are known to be NP-Hard as they generalize the Bin Packing problem, which is NP-hard [2].

Any problem that considers additional constraints or different objectives is commonly known as a general assembly line balancing problem (GALBP), see [3] for a classification scheme or [4] for a recent review. Usually, GALBPs are built on the SALBP formulation by incorporating additional characteristics of real-life situations. The problem studied in this paper builds on the SALBP by considering that the operation times could depend on the performance of the workers or the characteristics of the robots performing these tasks [5]. In this case, a solution requires an assignment of operators, workers or robots to the stations in addition to the partitioning of tasks.

Several problems under the aforementioned condition have been previously studied. Rubinovitz et al. [6] put forward the first study (to the best of our knowledge), in which both the operator assignments and the line balancing decisions are simultaneously considered. The study considers that different robots are available to perform the tasks. Robots have different characteristics, which modify the operation times that are required to perform each task.

---

[*] Corresponding author. Tel.: +34 934010769; fax: +34 934016054.
E-mail addresses: mariona.vila.bonilla@upc.edu (M. Vilà), jorge.pereira@upc.edu, jorgepereira@ucn.cl (J. Pereira).

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.10.016

The objective is to find a solution in which the robot and the task assignments selected for each station maximize the efficiency of the line. This problem is known as the robotic assembly line balancing problem (RALBP), and it considers that no previous line design exists, and thus, the number of available robots of each type is not limited.

More recently, Miralles et al. [7] studied a line balancing application to a work center for the disabled. The authors consider an existing workforce that is formed by operators who have different skills and who perform the tasks assigned to each of the stations. The problem is known as the assembly line worker assignment and balancing problem (ALWABP) to highlight the differences from the previous problem because the existing workforce becomes an additional constraint. While there are some exceptions (see the review in Section 2) the literature refers to this problem as ALWABP, when the operators are unique, and as RALBP, when the operator selection is part of the problem.

In this paper, we study the ALWABP case (unique operators), in which the number of available operators and stations is identical and the objective is to maximize the efficiency by minimizing the cycle time; this case is known as ALWABP-2. It represents the common situation in which the number of operators has been derived from an existing line design but technology or product mix changes alter the operation times of the tasks, thus forcing the rebalancing. Note that the problem is NP-hard as it generalizes the SALBP.

To solve the problem, we propose an exact enumeration procedure, which is based on the branch, bound and remember (BB&R) algorithm presented in [8] for the SALBP-1, which is the best-performing procedure in the literature for the simple case. The implementation of the algorithm shows that it improves upon the previous procedures from the literature, outperforming other methods in terms of solution quality. The algorithm is able to verify the optimality of 269 instances and to find the optimal solution for 318 instances out of the 320 instances from the reference benchmark instance set used in several recent studies (see [9–12]).

The remainder of this paper is structured as follows: Section 2 is devoted to the problem description and a literature review for balancing problems, especially for the problem at hand. Section 3 introduces the different lower bounding methods that will be used to develop a BB&R algorithm, which will be the focus of Section 4. Section 5 presents the results of several computational experiments with the proposed algorithm, and Section 6 gives a brief summary and presents selected conclusions to the present work.

2. Problem description and literature review

Formally, an ALWABP instance can be defined by a set of elementary tasks, V, into which the assembly process of the product is divided; a set of operators, O, that must perform the tasks; and an ordered set of stations ($j = 1, \dots, m$), where operators perform their assigned tasks. Each task has an operation time that depends on the operator assigned to perform the task, d_{ik} ($i = 1, \dots, |V| k = 1, \dots, |O|$). Each station has a single operator assigned to it, and each operator must be assigned to only one station (note that $m = |O|$). Operators assigned to any station have an identical fixed allotted time to perform their tasks, which is known as the cycle time, c. Let S_j be the set of tasks that are assigned to station j; if operator k is assigned to station j, then $\sum_{i \in S_j} d_{ik} \le c$ must hold.

In addition to the cycle time limit on each station, some tasks present precedence relations. Precedence relations are represented using a directed acyclic graph, $G(V, A)$, in which vertices are associated to tasks, and an arc between task i and task i' indicates that task i must be processed before task i'. In other words, if $i \in S_j$ and $i' \in S_{j'}$, then $j \le j'$ must hold. The set of tasks i' for which there is a path from i to i' in $G(V, A)$ is known as the set of successors of task i. Using the tuple notation proposed in [3] the described problem corresponds to the [pa, cum |equip| c].

Note that this problem is reversible. In other words, the direction of the arcs in $G(V, A)$ can be reversed, and any solution found for the new instance can be easily converted to a solution for the original instance. This property is important because the precedence relations affect the performance of the solution procedures, and the reversed instance could be easier to solve in practice.

We will focus our review on the solution procedures for both the RALBP and ALWABP cases with type-1 and type-2 objective functions. We refer the reader to [1,4] for recent reviews of the state-of-the-art for other line balancing problems.

Rubinovitz et al. [6] formulate the RALBP as the problem of allocating equal amounts of work to stations on the line while selecting a robot for each station. The authors consider that there are different types of robots to choose from, and the objective is to minimize the number of required stations. The problem is solved using a task-oriented best-first branch-and-bound, and several heuristic rules are used to truncate the search tree if the method fails to obtain the optimal solution. In [13], the same problem is studied with a type-2 objective. The objective is to assign tasks to stations and to select the best fit robot type for each station to minimize the cycle time. The problem is then solved using two different versions of a genetic algorithm (GA).

A hybrid GA for the type-2 problem is proposed in [14], in which a basic GA is combined with a local search procedure using the principles of a variable neighborhood search [15]. The members of the population are represented using three different vectors: a task sequence vector that contains a permutation of all of the tasks, a breakpoint vector that contains the positions of the first task of each station, and a robot assignment vector that indicates the operator that is assigned to each station. The GA is tested using a benchmark set in which the robots to be used are already known and, as such, the GA is used to solve the ALWABP-2. Despite this, the authors use the RALBP nomenclature, which highlights the relationship between the ALWABP and the RALBP.

For the ALWABP, an integer programming model is presented in [7]. The paper also reports its application to a work center for the disabled. The same authors propose a branch-and-bound algorithm as well as a branch-and-bound-based heuristic for the problem in [16]. The results show that the proposed branch-and-bound can optimally solve small-sized instances (11 tasks). Both studies are focused on the type-2 problem, even if the procedure proposed in [16] solves it by iteratively solving type-1 instances.

In [9], an iterated beam search (BS) method for a type-2 objective is proposed. This method is based on the reformulation of the type-2 problem as one of finding the smallest cycle time for which there exists a feasible solution with the desired number of stations for the type-F problem. The BS part is iteratively applied to find feasible solutions for type-F problems with smaller cycle times until the total run time is consumed. The solutions provided by the method on a set of instances previously proposed in [17] are compared to the solution of the integer programming model using the CPLEX commercial mixed integer linear programming solver. CPLEX can optimally solve the formulation for instances with up to 28 tasks within reduced time limits, but it is ineffective for larger instances.

In [10], the authors study the usage of a station-oriented constructive procedure for the problem. The procedure is based on the priority rule heuristic method for the SALBP, in which the assignment of operators to stations is taken into account. A computational experience using the instances proposed in [17]