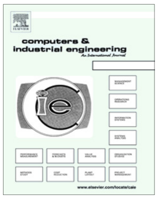
Computers & Industrial Engineering 104 (2017) 134–144

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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



Integer based formulation for the simple assembly line balancing problem with multiple identical tasks

Celso Gustavo Stall Sikora [a], Thiago Cantos Lopes [a], Daniel Schibelbain [b], Leandro Magatão [a,*]

[a] Graduate Program in Electrical and Computer Engineering (CPGEI), Federal University of Technology – Paraná (UTFPR), Curitiba 80230-901, Brazil
[b] Renault do Brasil S.A., São José dos Pinhais 83070-900, Brazil

ARTICLE INFO

Article history:
Received 5 April 2016
Received in revised form 24 November 2016
Accepted 18 December 2016
Available online 28 December 2016

Keywords:
Assembly line balancing
Mixed integer linear programming
Identical tasks

ABSTRACT

Assembly lines, especially those with welding procedures, can present several tasks with the same properties. These tasks can be treated as tasks with replicas, simplifying the problem. A Mixed Integer Linear Programming model is presented for the Simple Assembly Line Balancing Problem with Multiple Identical Tasks (or Repeated Tasks). Integer variables were used to define the number of identical tasks performed in each station. Along with variable reduction rules, the compact formulation presents only a fraction of the variables of equivalent binary models when several repeated tasks are present. Three instances inspired in real assembly lines and adapted benchmark problems with repeated tasks are used to compare the formulations. Using a universal solver, the integer formulation outperformed the binary formulation for the vast majority of instances and achieved competitive results in relation to the efficient procedure SALOME-2 (a dedicated algorithm based on branch-and-bound for Simple Assembly Line Balancing Problem). Grouping identical tasks proved to simplify the problem, allowing the procedure to solve larger instances.

1. Introduction

The Assembly Line Balancing Problem (ALBP) is the problem in which a set of product assembly operations is divided among workstations in flow-lines. Firstly defined by Salveson (1955), an ALBP solution is the task partition that maximizes an assembly line efficiency. The allocation of tasks, however, is subjected to technological restrictions in respect of the order tasks are assigned. When a task depends on the conclusion of another operation, a precedence relation is defined.

An ALBP can be described by a graph G in which weighted nodes V are tasks to be allocated in stations $S\ \{1,\ldots,M\}$ and edges represent the precedence relations A (Scholl, 1999). An instance can be defined as $G = \{V, A, t\}$ where V is the set of tasks $i\ \{1,\ldots,N\}$, and t_i is the duration time of task i. As a result of precedence relations, P_i (F_i) is defined as the set of direct predecessors (successors) of task i, while A is the task pairs $(i,j)\ |\{i \in V, j \in F_i\}$. The set P_i^* (F_i^*) contains all predecessors (successors) of task i, including direct and indirect relations. As an example, in Fig. 2, Task 1 is a direct predecessor of Task 4 while Task 10 is an indirect successor of Task 1.

Several ALBP variations have been modeled based on characteristics found in the industry. When several products are assembled in the same line, a mixed-model balancing is necessary (Gökcen & Erel, 1998). Further improvement on production levels are possible by integrating the balancing problem with the model sequencing problem (Hamzadayi & Yildiz, 2013; Kucukkoc & Zhang, 2014). Another related problem that can be solved simultaneously is the allocation of workers along the assembly line (Borba & Ritt, 2014; Moreira, Cordeau, Costa, & Laporte, 2015; Ramezanian & Ezzatpanah, 2015; Sikora, Lopes, & Magatão, 2016; Vilà & Pereira, 2014). Moreover, assembly line may present two operation sides (Lee, Kim, & Kim, 2001) or set-up times between operation pairs (Yolmeh & Kianfar, 2012). An effort of systematic classification of models was performed by Boysen, Fliedner, and Scholl (2007) while recent reviews on general topics on ALBP are from Becker and Scholl (2006), Boysen, Fliedner, and Scholl (2008) and Battaïa and Dolgui (2013).

The basic version of the problem, which is the base for most of the its extensions, was labeled by Baybars (1986). In his survey, Baybars defined the Simple Assembly Line Balancing Problem (SALBP) as subjected to several simplification hypothesis (SH):

- (SH-1) All stations are equally equipped and manned
- (SH-2) Task processing time is independent of the workstation
- (SH-3) Any station is able to perform any task

---

* Corresponding author.
E-mail address: magatao@utfpr.edu.br (L. Magatão).

http://dx.doi.org/10.1016/j.cie.2016.12.026
0360-8352/

- (SH-4) The assembly line is serial, no feeder or subassembly lines are considered
- (SH-5) The assembly line is designed for the production of a single product
- (SH-6) All problem's parameters are deterministic (processing time and precedence relations)

Baybars (1986) also defines two versions of the SALBP, namely type-1 and type-2. In SALBP-1, the line's cycle time is given and the objective is to minimize the number of workstations needed for the task assignments. On the other hand, SALBP-2 is subjected to a given number of workstations while the cycle time is the optimization focus.

Although extensions of ALBP are a growing trend on the more recent articles, the simple version (SALBP) has been the most intensively studied ALBP in the literature (Battaïa & Dolgui, 2013). The first two integer models for SALBP were proposed by Bowman (1960). Bowman's first model used unnecessary integer variables that were simplified into binary variables by White (1961). The computational capacity at the time was not enough advanced for practical purposes. A decade later, Thangavelu and Shetty (1971) and Patterson and Albracht (1975) developed binary models adapted to be solved with Balas' method (Geoffrion, 1967). Patterson and Albracht (1975) also developed a reduction approach to the number of variables using the problem structure, resulting in a compact model. Scholl (1999) presents a review on the SALBP models and introduces two SALBP models: one based on task sequencing and other whose assignment is determined by the difference of two binary variables. More recently, Pastor and Ferrer (2009) developed a model containing restrictions that dynamically reduced the search space as function of the incumbent solution. Their model does not require an initial upper bound, the restrictions use the intermediary solutions to cut the search field. The several formulations for the precedence relations were discussed by Ritt and Costa (2015), who proposed a model with a tighter precedence restriction.

SALBP formulations are extensively used among further modeling in ALBP extension using mixed-integer solvers (Battaïa & Dolgui, 2013). For SALBP, however, problem specific heuristics, metaheuristics, or exact methods are usually more efficient than solvers. A comparative study between 12 solution methods of different concepts was performed by Pape (2015). Pape compared the performance of genetic algorithm (Falkenauer & Delchambre, 1992; Sabuncuoglu, Erel, & Tanyer, 2000), differential evolutionary algorithm (Nearchou, 2005), ant colony optimization (Bautista & Pereira, 2002), tabu search (Lapierre, Ruiz, & Soriano, 2006; Scholl & Voss, 1997), MultiHoffman heuristic (Fleszar & Hindi, 2003), bounded dynamic programming (Bautista & Pereira, 2009), beam search (Blum, 2008), and branch-and-bound (Scholl & Klein, 1997, 1999) for the solution of SALBP-1. The most efficient algorithms are dedicated branch-and-bound or dynamic-programming procedures that use the problem structure to determinate the search procedure.

Scholl and Becker (2006) presented a survey on exact and heuristic procedures highlighting the performance of SALOME-1 (Scholl & Klein, 1997, 1999) and SALOME-2 (Klein & Scholl, 1996) for the SALBP-1 and SALBP-2, respectively. A further development on an exact procedure for SALBP-1 is due to Sewell and Jacobson (2012). Their Branch, Bound, and Remember Algorithm (BB&R) solved to optimality, for the first time, all the 269 SALBP-1 instances of Scholl (1999)'s dataset with an average of only 0.43 s. For SALBP-2, however, to the best knowledge of the authors, still no exact algorithm outperforms SALOME-2 developed by Klein and Scholl (1996).

All of the models and procedures presented treat each task individually, assigning a set of variables for each task. Assembly lines with welding procedures can, for example, present several tasks that have similar characteristics. Fig. 1, for instance, illustrates 16 tasks divided in three groups (G_1, G_2, and G_3). Within each group, the welding spots are performed in similar positions and have the same function. In an ALBP with identical tasks (repeated tasks with the same processing time and precedence relations), the problem formulation does not benefit from the fact that many tasks can be identical and therefore can be gathered in groups of similar tasks. Here, we propose a representation that benefits from repeated tasks to simplify the problem solving. This way, solution methods are able to compute bigger instances of assembly line balancing problems with these characteristics.

The paper is structured as follows. In Section 2, the definition of identical tasks is proposed, along with the requirement for grouping tasks and their effect on the problem structure. Section 3 presents one review of a binary formulation for SALBP and the new integer variable based formulation modeling groups of tasks. In Section 4 computational experiments are described with the results showing the advantages of an integer formulation. The conclusion remarks are found in Section 5.

2. The simple assembly line with multiple identical tasks problem

Among tasks performed in an assembly line, it is possible that some operations appear as copies of other tasks. Assembling two pieces using screws or welding operations exemplifies such cases: the operation might require multiple screws, welding spots or beads. When procedures have the same duration and the order they are performed is irrelevant, they can be said to be identical tasks.

Fig. 2 exemplifies a case in which 15 tasks are constituted of 5 tasks of 3 copies each. Between each group of identical tasks, once they can be performed in any order, there are not precedence relations. As a result, groups of identical tasks appear as vertical aligned tasks.

In this kind of problem, there might be several answers with identical quality. Suppose the optimal answer requires a workstation to perform Tasks 1 and 2. Another answer could assign Tasks 1 and 3 or 2 and 3 to that workstation, while still resulting in an opti-

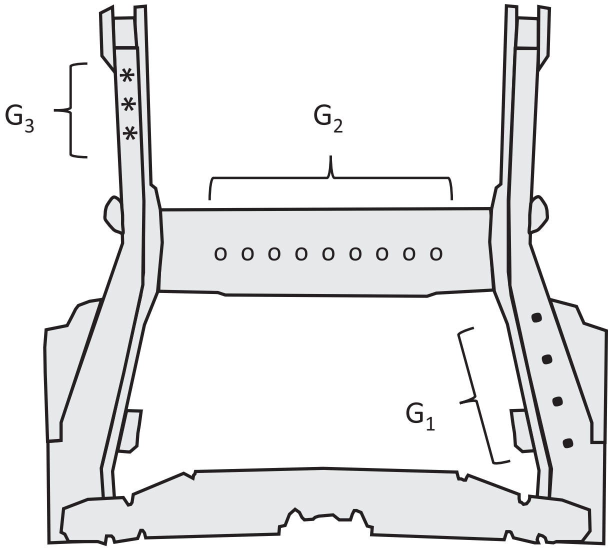


Fig. 1. Representation of an automotive part to illustrate a welding procedure. The symbols exemplify welding points. The asterisk and the filled and unfilled circles represent tasks than can be considered identical. Within each group (G_1, G_2, and G_3), the processing time and precedence relations are the same for each task.