Computers & Industrial Engineering 90 (2015) 241–258

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Enumerations and stability analysis of feasible and optimal line balances for simple assembly lines

Yuri N. Sotskov [a,1], Alexandre Dolgui [b,*], Tsung-Chyan Lai [c,d,2], Aksana Zatsiupa [a,1]

[a] United Institute of Informatics Problems, National Academy of Sciences of Belarus, 6 Surganova Street, Minsk 220012, Belarus
[b] Ecole des Mines de Nantes, IRCCYN, UMR CNRS 6597, La Chantrerie, 4, rue Alfred Kastler - B.P. 20722, Nantes F-44307 cedex 3, France
[c] Department of Business Administration, National Taiwan University, 85 Roosevelt Road, Section 4, Taipei 106, Taiwan
[d] School of Economics and Management, Harbin Engineering University, 145 Nantong Street, Harbin 150001, China

ARTICLE INFO

Article history:
Received 26 October 2014
Received in revised form 28 August 2015
Accepted 29 August 2015
Available online 18 September 2015

Keywords:
Combinatorial optimization
Assembly line balancing
Uncertain processing times
Stability radius
Complexity analysis

ABSTRACT

For a simple assembly line, it is necessary to minimize a number of the workstations for processing a partially ordered set of the tasks $V = \{1, 2, \ldots, n\}$ within a fixed cycle time (such a problem is denoted as SALBP-1). A dual assembly line balancing problem denoted as SALBP-2 is to minimize a cycle time provided that a number of the workstations is fixed. An initial vector $t = (t_1, t_2, \ldots, t_n)$ of the processing times of the tasks V is given for both problems SALBP-1 and SALBP-2. For a subset $\widetilde{V} \subseteq V$ of the manual tasks $j \in \widetilde{V}$, the processing times t_j may vary since operators may have different skills, levels of fatigue, experience, and motivation. For any automated task $i \in V \setminus \widetilde{V}$, the processing time t_i cannot vary. We investigate a stability of an optimal line balance for the assembly line with respect to variations of the processing times of the manual tasks (a line balance is stable, if it is optimal for any sufficiently small variation of the processing times). We developed the enumerative algorithms for constructing feasible and stable optimal line balances for the problem SALBP-1 and those for the problem SALBP-2. Computational results for the stability of the assembly line balances showed that there are a lot of unstable optimal line balances for the tested benchmark assembly lines. The simulation for the benchmark assembly line showed that the stable optimal line balance considerably outperforms the unstable ones. The complexity analysis of the assembly line balancing problems with different partial orders given on the task set V has been developed.

1. Introduction

We consider a single-model paced assembly line, which continuously manufactures a homogeneous product in a large quantity (Amen, 2000; Baybars, 1986; Erel & Sarin, 1998; Scholl, 1999; Suer, 1998; Wee & Magazine, 1982; Wei & Chao, 2011; Zacharia & Nearchou, 2013). The assembly line consists of the ordered set of the workstations $S = \{S_1, S_2, \ldots, S_m\}$ linked by a conveyor belt or another moving equipment. The total work content to be performed on the assembly line has been split up into work elements called tasks. The set of tasks $V = \{1, 2, \ldots, n\}$ being repeatedly processed on the workstations S is fixed. Each task $i \in V$ is indivisible and workstation $S_k \in S$ has to perform a set of the assigned tasks within a cycle time c. All the workstations S start simultaneously with processing the assigned tasks one by one in some order. The technological factors define a partial order on the task set V given by the precedence digraph $G = (V, A)$.

The assembly line is often labor-intensive (Song, Wong, Fan, & Chan, 2006; Suer, Arikan, & Babayigit, 2009; Zacharia & Nearchou, 2013). A variety of tasks have to be manually performed by the operators. Variations in the processing times of the manual tasks are inevitable since manual task times depend on operator fatigue, skill level, job dissatisfaction, poorly maintained equipment, etc. In this paper, we assume that set V of the tasks consists of two specific subsets: The non-empty subset $\widetilde{V}$ includes all the manual tasks with variable processing times and the remaining subset $V \setminus \widetilde{V}$ includes all the automated tasks with fixed processing times.

The assembly line balancing problem is to find an optimal line balance, i.e. an assignment of the partially ordered task set V into a minimal number m of the linearly ordered workstation set S for the given production rate $1/c > 0$. In the (Baybars, 1986) paper

* Corresponding author. Tel.: +33 2 51858218.
E-mail addresses: sotskov@newman.bas-net.by (Y.N. Sotskov), alexandre.dolgui@mines-nantes.fr, dolgui@emse.fr (A. Dolgui), tclai@ntu.edu.tw, tclai@hrbeu. edu.cn (T.-C. Lai), ztp.oksana100@yandex.ru (A. Zatsiupa).
[1] Tel.: +375 217 2842120.
[2] Tel.: +886 2 33661046, +86 451 82519783.

http://dx.doi.org/10.1016/j.cie.2015.08.018
0360-8352/

and (Scholl, 1999) monograph, the abbreviation SALBP-1 is used to denote such a Simple Assembly Line Balancing Problem. In the industrial practice, the problem SALBP-1 arises when a new single-product assembly line has to be designed for the production rate $1/c$. The problem SALBP-1 may also arise when a cycle time c of the assembly line must be changed because of the changing customer demands for the final product produced on the assembly line.

A dual Simple Assembly Line Balancing Problem, SALBP-2, is to find an optimal line balance for a given set of the m workstations, i. e. to find a feasible assignment of the partially ordered set of the tasks $V = \{1, 2, \dots, n\}$ into the linearly ordered set of the workstations $S = \{S_1, S_2, \dots, S_m\}$ in a way such that a cycle time c is minimal. In the assembly industry, the problem SALBP-2 arises when the assembly line needs a re-engineering due to significant changes of the task times.

A labor cost per the assembly product may be estimated as a sum of the wage rates of all workstations multiplied by cycle time c. Manual task times $\widetilde{t} = (t_1, t_2, \dots, t_{\tilde{n}}),\ \tilde{n} \leqslant n$, are variable depending on the operators that have to be paid for the whole cycle time c irrespective of factual times used for their tasks. So, for the labor-intensive assembly line, a line balance generated for the fixed processing times may result in a lower effectiveness and an increased production cost when the processing times of the manual tasks will be changed. To avoid these drawbacks, we propose to use line balances, which remain optimal in larger stability balls in space of variations of the task processing times $\widetilde{t}$. A formal definition of the stability ball is given in Section 2.4.

1.1. Contributions of this work

Since both problems SALBP-1 and SALBP-2 are binary NP-hard even for two workstations (Scholl, 1999; Wee & Magazine, 1982), an exact algorithm for solving them must estimate (either implicitly or explicitly) the objective values for all feasible line balances. We propose to make these estimations explicitly via constructing the set B^* of all feasible line balances. Using the constructed set B^*, one can solve both problems SALBP-1 and SALBP-2 exactly, investigate the stability of the optimal line balances respecting the variations of the task times $\widetilde{t}$, and select stable line balances that remain optimal for all sufficiently small variations of the task processing times.

In this work, we developed algorithms for constructing the line balances for the problems SALBP-1 and SALBP-2. The complexity of the enumeration of the line balances B^* depends on the precedence digraph $G = (V, A)$: For the same order $n = |V|$ of the digraph G, this enumeration is easer, if number $|A|$ of the arcs is larger. So first, we shown how to construct feasible line balances for the easiest case with the complete precedence digraph $G = (V, A)$, where $A = A^C$ with $|A^C| = \frac{n(n-1)}{2}$, i.e., when the order strength of the circuit-free digraph $G = (V, A^C)$ is equal to 1. Then we shown how to construct all feasible line balances for any circuit-free digraph $G = (V, A)$ using the set B^* constructed for the complete circuit-free digraph $G = (V, A^C)$ with $A \subset A^C$.

We proved several claims for simplifying the construction of the set B^*. These claims are used in the developed enumerative algorithms. It is proven that if the precedence digraph G is complete, $G = (V, A^C)$, then the problem SALBP-1 can be solved in $O(n)$ time. We proved also that both problems SALBP-1 and SALBP-2 with $m = 2$ are binary NP-hard if the precedence digraph G is complete (n/m)-partite digraph: $G = (V, A_{n/m})$. It is shown that the order strength of the complete (n/m)-partite digraph $G = (V, A_{n/m})$ is tending to 1 if number n of the tasks is tending to infinity while number m of the workstations is fixed.

Using the proven results, we developed algorithms for constructing the set $B^*(G, c, t)$ of all stable optimal line balances for the problem SALBP-1 and the set $B^*(G, m, t)$ of those for the problem SALBP-2. Hereafter, t denotes an initial vector of the task processing times. Both sets $B^*(G, c, t)$ and $B^*(G, m, t)$ may be used in the assembly industry when the assembly line is designed or re-engineering of the acting assembly line may increase the effectiveness of the assembly line.

1.2. Paper outline

Section 2 presents problem settings and a survey of stability results with criteria for zero stability radii. A background for efficient algorithms for constructing feasible and stable optimal line balances is developed in Section 3. The proofs of the claims including algorithms are given within the main text. Other proofs are given in appendixes (the proofs of Theorem 5, Lemmas 2 and 3). The complexity analysis of the problems SALBP-1 and SALBP-2 with respect to the order strength of the precedence digraph has been developed in Section 4. Section 5 contains algorithms for enumerations of the stable optimal line balances for the problems SALBP-1 and SALBP-2. Computational results for the benchmark instances are discussed in Section 6, where it is explained how to use the proven formulas and the developed algorithms in the assembly industry (Section 6.4). Concluding remarks and perspectives are discussed in Section 7.

2. Problem settings, notations, and stability results

We assume that $\widetilde{V} = \{1, 2, \dots, \tilde{n}\}$ and $V \setminus \widetilde{V} = \{\tilde{n}+1, \tilde{n}+2, \dots, n\}$, where $1 \leqslant \tilde{n} \leqslant n$. An initial vector of the processing times of the manual tasks $\widetilde{V}$ is denoted as $\widetilde{t} = (t_1, t_2, \dots, t_{\tilde{n}})$, that of the automated tasks $V \setminus \widetilde{V}$ as $\bar{t} = (t_{\tilde{n}+1}, t_{\tilde{n}+2}, \dots, t_n)$, and that of the tasks V as $t = (\widetilde{t}, \bar{t}) = (t_1, t_2, \dots, t_n)$.

2.1. Setting of the problem SALBP-1

Let a subset $V_k^{b_r} \neq \emptyset$ of the task set V be assigned to the workstation S_k, where $k \in \{1, 2, \dots, m\}$. The following assignment b_r:

$$V = V_1^{b_r} \bigcup V_2^{b_r} \bigcup \cdots \bigcup V_m^{b_r} \tag{1}$$

of the partially ordered set V of the tasks into the linearly ordered set $S = \{S_1, S_2, \dots, S_m\}$ of the workstations, where $V_k^{b_r} \bigcap V_l^{b_r} = \emptyset,\ 1 \leqslant k < l \leqslant m$, is feasible and called a line balance for the problem SALBP-1, if the following two conditions hold.

Condition 1: The assignment b_r does not violate a partial order given on the set V by a circuit-free digraph $G = (V, A)$, i.e., inclusion $(i, j) \in A$ implies that task i is assigned to workstation $S_k \in S$ while task j is assigned to workstation $S_l \in S$ in a way such that $1 \leqslant k \leqslant l \leqslant m$.

Let $B(G)$ denote a set of all assignments b_r defined in (1) such that Condition 1 holds.

Condition 2: The cycle time c is not exceeded for any workstation $S_k,\ k \in \{1, 2, \dots, m\}$, i.e. the sum $t(V_k^{b_r})$ of the processing times of all tasks $V_k^{b_r}$ assigned to workstation S_k (the set $V_k^{b_r}$ is called a workstation load) cannot be greater than cycle time: $c \geqslant t(V_k^{b_r}) = \sum_{i \in V_k^{b_r}} t_i$.

In the above notation $t(V_k^{b_r}),\ t$ indicates vector $t = (\widetilde{t}, \bar{t})$ of the processing times, for which the workstation time $t(V_k^{b_r}) = \sum_{i \in V_k^{b_r}} t_i$ has been calculated. Let $B(G, c) = \{b_1, b_2, \dots, b_h\}$ be a set of all assignments $b_r \in B(G)$ such that Conditions 1 and 2 hold. For the problem SALBP-1, line balance $b_r \in B(G, c)$ is optimal, if it uses a