Computers & Industrial Engineering 57 (2009) 1218–1225

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Assembly line balancing with station paralleling

Yunus Ege, Meral Azizoglu *, Nur E. Ozdemirel

Industrial Engineering Department, Middle East Technical University, 06531 Ankara, Turkey

ARTICLE INFO

Article history:
Received 5 September 2008
Received in revised form 25 March 2009
Accepted 28 May 2009
Available online 23 June 2009

Keywords:
Single model assembly line balancing
Station paralleling
Branch and bound algorithm

ABSTRACT

We consider the NP-hard problem of assembly line balancing with station paralleling. We assume an arbitrary number of parallel workstations can be assigned to each stage. Every task requires a specified tooling/equipment, and this tooling/equipment should be available in all parallel workstations of the stage to which the task is assigned. Our objective is to find an assignment of tasks to stages so as to minimize sum of station opening and tooling/equipment costs. We propose two branch and bound algorithms: one for optimal solutions and one for near optimal solutions. We find that optimal solutions can be found quickly for medium sized problem instances; for larger sized problems we find high quality solutions in reasonable solution times.

1. Introduction

An assembly line is a sequence of stages through which a set of tasks is processed. The stages are linked serially, usually by a transport mechanism. A stage may consist of a number of parallel workstations that are equipped identically. Each task has a prespecified processing time and a set of predecessors such that it cannot start before all its predecessors are complete.

An assembly line balancing (ALB) problem considers the assignment of tasks to stages so as to minimize some prespecified performance measures like cycle time and station opening costs. The assignment is feasible provided that the precedence constraints are satisfied and the sum of the task times assigned to a stage does not exceed time capacity, i.e., cycle time multiplied by the number of workstations. The majority of the research in the ALB literature assumes a design where each stage has a single workstation. In today's competitive manufacturing, increased diversity and volume of products necessitate parallel assembly lines where workstations of the same stage produce different units of the same product.

Two types of paralleling have been considered in the literature, namely task paralleling and workstation paralleling. Task paralleling allows a task to be performed in more than one stage, so the tool/equipment required by that task should be placed in all those stages. Paralleling a workstation, on the other hand, requires the duplication of all tools/equipment needed for the tasks assigned to that stage.

Paralleling might decrease labor requirements as the tasks can be fit to a stage more tightly due to the increased time capacity. The production rate of the line is limited with the maximum task time in the absence of paralleling. The increased time capacity brought by paralleling increases maximum task time thereby increasing the production rate. However, there might be extra capital cost due to the tool/equipment duplication.

Despite its practical importance, the research on the parallel assembly lines is quite limited. The studies to date present optimization algorithms for some special cases of the problem or approximation algorithms for the general case. In this study, we design an optimization algorithm for the ALB problem with station paralleling. The problem we are considering is NP-hard, which suggests that any optimization procedure will run into the computational difficulties as the problem size increases. For the problem sizes that cannot be handled by our algorithm, we propose a branch and bound based heuristic procedure.

The rest of the paper is organized as follows. In the next section, we define our problem and review previous research. Section 3 introduces our solution procedures and Section 4 reports the computational results of our experiment. We conclude with a summary in Section 5.

2. Problem definition and related literature

We consider deterministic parallel assembly line balancing problem in which an arbitrary number of parallel workstations are allowed in each stage. We use the following assumptions throughout the study:

- The cycle time to meet the demand, C, is known.
- There are n tasks. Task i has a processing requirement of t_i time units and a tool/equipment (equipment hereafter for simplicity) requirement of type e_i.
- The precedence relations that give information about the relative processing order of tasks are known and fixed.

* Corresponding author. Tel.: +90 312 210 2281; fax: +90 312 210 4786.
E-mail addresses: meral@ie.metu.edu.tr (M. Azizoglu), nurevin@ie.metu.edu.tr (N.E. Ozdemirel).

0360-8352/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2009.05.014

– Equipment k costs A_k money units. This cost represents amortized purchasing and operational cost.
– There is a fixed cost of L money units for opening a workstation. This cost includes the labor and overhead costs of operating a workstation.

Our constraints can be stated as follows:

– Each task should be assigned to exactly one stage.
– A task can be performed in any stage provided that the equipment required by the task is available in the workstation, and all its predecessors are processed in no later stages.
– The sum of the processing times of the tasks assigned to a stage cannot exceed the cycle time multiplied by the number of workstations in that stage.

Our objective is then to minimize

$$Z = \sum_{j=1}^{n} Ly_j + \sum_{j=1}^{n} \sum_{k \in T_j} A_k y_j$$

where y_j is the number of workstations opened in stage j and, T_j is the set of equipment types assigned to stage j.

Note that n is an upper bound on the number of stages, m, where m is a solution by the model such that $y_m > 0$ and $y_{m+1} = 0$.

A number of variants of the problem have been studied in the literature. The problem reduces to the traditional single workstation ALB problem when A_k is zero for all k and y_j is set to 1 for all j. Single workstation ALB can be reduced to a bin-packing problem whose NP-hardness complexity result is set by Garey and Johnson (1981). Hence any generalization of the single workstation ALB problem, therefore our problem, is NP-hard. Several authors have studied the single workstation ALB problem, and several review articles have appeared. Some noteworthy review articles are due to Baybars (1985, 1986), Ghosh and Gagnon (1989), Gagnon and Ghosh (1991), Scholl and Becker (2006), Becker and Scholl (2006) and Boysen, Fliedner, and Scholl (2007). The research on the parallel workstation ALB is limited and of relatively recent origin. Pinto, Dannenbring, and Khumawala (1981) propose a branch and bound algorithm for the special case with $A_k = 0$ for all k. Bard (1989) presents a dynamic programming approach assuming A_k is equal to L for all k. Moreover, Pinto et al. (1981) and Bard (1989) set y_j to one or two for all j, i.e., they allow at most two parallel workstations in each stage. McMullen and Frazier (1998) propose a simulated annealing technique for stochastic task times. They measure the line performance by the total cost and the extent to which cycle time is met. Vilarinho and Simaria (2002) introduce a two-stage heuristic method for mixed-model assembly lines. Their primary goal is to minimize the number of workstations for a given cycle time and the secondary goal is to balance the workload between workstations. Bukchin and Rubinovitz (2003) study the equipment selection problem on parallel workstations. They investigate the influence of assembly sequence flexibility and cycle time on the balancing improvement due to the station paralleling. Vilarinho and Simaria (2006) propose an ant colony optimization algorithm for balancing the mixed-model assembly lines with zoning restrictions and parallel workstations. Askin and Zhou (1997) consider mixed-model assembly lines with task dependent equipment costs and arbitrary number of parallel workstations. They propose a heuristic procedure that uses a threshold value for the equipment utilization. We, in this study, consider parallel workstations and propose a branch and bound algorithm to minimize total equipment and workstation opening costs. To the best of our knowledge our study is the first optimization effort that considers equipment and workstation opening costs simultaneously with arbitrary number of workstations.

3. Solution algorithms

In this section we propose two solution procedures: a branch and bound algorithm for finding optimal solutions and a heuristic branch and bound algorithm for finding near optimal solutions.

3.1. Branch and bound algorithm, BAB1

BAB1 forms a complete assignment starting from the first stage. A node at the rth level of the branch and bound tree corresponds to a partial assignment with r tasks assigned to the initial stages. For each node there are two decisions for each unassigned task. These decisions are:

Type I decision: An assignment to a current stage, by adding parallel workstation(s) when cycle time constraint does not permit.

Type II decision: An assignment to the next stage, i.e., closing the current stage.

We need the following additional notation:

σ: a partial assignment of tasks.
σ_i: a partial assignment formed by appending unassigned task i to σ.
$y_j(\sigma)$: number of workstations assigned to stage j in σ.
$ST(\sigma)$: last stage considered in σ, i.e., number of stages in σ.
$S(\sigma)$: set of tasks assigned to stage $ST(\sigma)$.
$T_j(\sigma)$: set of equipment types assigned to stage j in σ.

After the addition of task i to σ, the number of workstations is updated as follows:

$$\textit{For Type I decision}: \quad ST(\sigma_i) = ST(\sigma) \quad \text{and} \quad y_{ST(\sigma_i)}(\sigma_i) = \left\lceil \frac{\sum_{u \in S(\sigma)} t_u + t_i}{C} \right\rceil$$

$$\textit{For Type II decision}: \quad ST(\sigma_i) = ST(\sigma) + 1 \quad \text{and} \quad y_{ST(\sigma_i)}(\sigma_i) = \left\lceil \frac{t_i}{C} \right\rceil$$

Total Cost, $TC(\sigma_i)$, can be calculated as follows:

$$TC(\sigma_i) = L \sum_{j=1}^{ST(\sigma_i)} y_j(\sigma_i) + \sum_{\substack{j=1 \\ k \in T_j(\sigma_i)}}^{ST(\sigma_i)} A_k y_j(\sigma_i)$$

To evaluate the partial assignments, lower bounds are developed for total station opening cost, LB_S, and total equipment cost, LB_E, as:

$$LB_S = L\left(\sum_{j=1}^{ST(\sigma_i)-1} y_j(\sigma) + \left\lceil \frac{\sum_{u \in S(\sigma_i) \text{ or } u \notin \sigma_i} t_u}{C} \right\rceil \right) \quad \text{and}$$

$$LB_E = \sum_{j=1}^{ST(\sigma_i)} \sum_{k \in T_j(\sigma_i)} A_k y_j(\sigma_i) + \sum_{k \in \{e_u | u \notin \sigma_i\} \text{ and } k \notin T_{ST(\sigma_i)}} A_k.$$

The first component in LB_S is the station opening cost for the closed stages. The second component is the minimum number of workstations that should be formed for the tasks already assigned to the current stage and the tasks that are not assigned yet.

The first component in LB_E is the equipment cost of assigned tasks, whereas the second component is the minimum equipment cost for unassigned tasks. The minimum equipment cost is found by assuming all unassigned tasks are processed in the current stage, without any workstation duplication.

We fathom a node whenever $LB_S + LB_E$ is no smaller than the cost of the best known solution. The nodes that cannot be fathomed by the lower bounds are listed in nondecreasing order of their total costs, and the node at the top of the list is selected for further branching.