Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

A Second Order-Cone Programming Formulation for Simple Assembly Line Balancing Problem

Vasile Moraru, Sergiu Zaporojan

Abstract

Decision support in order to assure an optimal business in the framework of an industrial company is based on some mathematical models, including optimization. The paper discusses the numerical solving of a task known as the Simple Assembly Line Balancing Problem (SALBP-I). In this context, a model based on the Second-Order Cone Programming (SOCP) it is proposed.

1 Introduction

The success at the level of an enterprise requires optimal organization of the production processes and related activities. These activities include organizational processes, economic and financial, production, trade, and, not least, information processes. What it is important in the organization of the business process it is the optimization of the enterprise activity, taking into account the market demands and current technology power. The activities listed are made by human agents and/or machines to help achieve business objectives across the enterprise. So, business process optimization is related to activities (or tasks), participants (human agents and/or machines) and targets (performance indicators).

In general, the requirements for optimal organization of the enterprise activity are actual and fit into the idea of reengineering. This idea is centered on all processes in the modern enterprise. Reengineering is a radical redesign of a business process to achieve a considerable improvement in performance indicators (cost, quality, productivity, etc.). The idea of reengineering is actual one as information technology is constantly changing. Hence the ability to adapt quickly to market demands.

Decision support for an optimal business in the framework of an industrial enterprise is not always possible without software products that are based on mathematical models of combinatorial optimization. It is the case of the simple assembly (manufacturing) line balancing problem (SALBP).

The assembly line consists of a finite number of workstations that are running individual operations (tasks) to manufacture a single product. The problem now is to combine operations and workstations in order to obtain an optimal distribution of the workload to a minimum number of stations. At the same time, it is necessary to ensure conditions of precedence for the execution of operations.

Assume the following conditions ([1]):

- the assembly line is designed for a single product and supports only one mode of functioning;
- the stations are serial arranged;

- the execution time for operations is deterministic;
- the partition of operations is prohibited;
- all operations must be performed;
- there are precedence constraints;
- the execution time of an operation does not depend on the station on which is running;
- the cycle time is fixed.

In the literature, several models of the SALBP-I problem have been presented ([2]). Most of them are formulated in terms of linear programming with binary variables and fit into the class of NP-hard problems ([3]). In this paper, we propose modeling the problem as a second-order cone programming (SOCP).

2 Mathematical programming model for SALBP-1

It is assumed an assembly line of *n* stations. At each station runs one operation by a person or an automatic device (robot), which is needed to manufacture a product. For product assembly all operations must be done in a strict order.

We denote by *P* the immediate precedence matrix of dimension $n \times n$:

$$P(i,j) = \begin{cases} 1, \mathit{if\ task\ j\ is\ an\ immediate\ successor\ of\ task\ i}, \\ 0, \mathit{otherwise.} \end{cases}$$

It is consider to be known the production rate *R* (the number of items collected per unit time) and the processing time t_i of the operation i. The number of stations can not be lower than T/C, where $T = \sum_{i=1}^{n} t_i$ is the time of all needed operations, and $C = 1/R$ is the total cycle time of the assembly line.

It is required to perform the distribution of operations to workstations so that the number of jobs to be minimal, i.e., the maximizing of the number of "empty" jobs. An "empty" is a plant that does not perform any operation.

We introduce Boolean variables x_{ij} and y_i:

$$x_{ij} = \begin{cases} 1, \mathit{if\ task\ i\ is\ assigned\ to\ workstation\ j}, \\ 0, \mathit{otherwise.} \end{cases}$$

$$y_i = \begin{cases} 1, \mathit{if\ workstation\ i\ has\ a\ task\ assigned\ to\ it}, \\ 0, \mathit{otherwise.} \end{cases}$$

such that, $x_{ij} = 1$ if the operation i is carried out at the workstation j and $x_{ij} = 0$ when it is carried out at another station; $y_i = 1$ if one of the operations is performed at the station i, and $y_i = 0$ in the opposite case. For the *SALBP-I* problem different formulations have been proposed. We will consider the following mathematical model ([1]):

$$\sum_{i=1}^{n} y_i \to \min \tag{1}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1, \forall i = 1,2,\mathrm{K}\ ,n\ , \tag{2}$$

$$\sum_{i=1}^{n} t_i x_{ij} \le C, \forall j = 1,2,K\ ,n\ , \tag{3}$$

$$x_{ik} \le \sum_{j=1}^{k} x_{sj}, \forall k = 1,2,K\ ,n, \forall i,s : P(i,s) = 1\ , \tag{4}$$

$$\sum_{i=1}^{n} x_{ik} \le n(1 - y_k), \forall k = 1,2,K\ ,n\ , \tag{5}$$

$$x_{ij} \in \{0,1\}, y_i \in \{0,1\}, \forall i,j = 1,2,K\ ,n\ . \tag{6}$$

Restrictions (2) ensure that each operation will be performed only at a single workstation, and (3) - that the cycle must be greater than or equal to the length of time at all stations. Conditions (4) require the precedence relations between operations. If $x_{ik} = 0$ (operation *i* is not running at the station *k*), then $\sum_{j=1}^{k} x_{ik}$ may take any value of 0 or 1 and (4) becomes $\sum_{j=1}^{n} x_{sj} \ge 0$, it is always true and not represent a constraint. If $x_{ik} = 1$, then (4) is equivalent to (2). Restrictions (5) means the following: if $y_k = 0$, then $\sum_{i=1}^{n} x_{ik} \le n$, which relationship is always satisfied. Thus (1) gives the number of "empty" jobs, i.e. the number of stations with $\sum_{i=1}^{n} x_{ik} = 0$, stations that do not perform any operation. Conditions (6) force x_{ij} and y_i to be binary.

Various heuristic and exact methods ([1]), ([4]), ([5]), ([6]) have been proposed to solve the zero-one linear program (1)-(6).

3 A second order-cone programming reformulation

The conditions (6) are equivalent to the non-convex quadratic constraints:

$$\left.\begin{aligned} y_i^2 - y_i &= 0, \forall i \\ x_{ij}^2 - x_{ij} &= 0, \forall i,j \end{aligned}\right\}. \tag{7}$$

It can be established that the constraints (7) are equivalent with:

$$\left.\begin{aligned} &\sum_{i=1}^{n} y_i^2 \ge \sum_{i=1}^{n} y_i, \\ &0 \le y_i \le 1, \forall i \end{aligned}\right\}, \tag{8}$$

and

$$\left.\begin{aligned} &\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}^2 \ge \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}, \\ &0 \le x_{ij} \le 1, \forall i,j \end{aligned}\right\}. \tag{9}$$

Applying the inequality:

$$\sum_{i=1}^{n} a_i \geq \sqrt{\sum_{i=1}^{n} a_i^2} \ ,$$

true for $\forall a_i \geq 0, a_i \in R$, from (8) we have:

$$\sum_{i=1}^{n} y_i^2 \geq \sum_{i=1}^{n} y_i \geq \sqrt{\sum_{i=1}^{n} y_i^2} \ .$$

Taking into account that the variables y_i are binary, from the last inequalities we obtain the second-order cone of dimension $(n+1)$ (Lorenz cone):

$$\mathrm{K}_1 = \left\{ \begin{bmatrix} z \\ y \end{bmatrix} : z \in R, y \in R^n \ \mathit{for} \sqrt{\sum_{i=1}^{n} y_i^2} \leq z \right\},$$

where $z = \sum_{i=1}^{n} y_i$ şi $y = (y_1, y_2, \mathrm{K}\ , y_n)^T$.

In the same way, from (9) we obtain:

$$\sqrt{\sum_{i=1}^{n} \left(\sum_{j=1}^{n} x_{ij} \right)^2} \leq \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} \leq \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}^2 \ .$$

Thus (9) determines the second-order cone:

$$\mathrm{K}_2 = \left\{ \begin{bmatrix} u \\ X \end{bmatrix} : u \in R, X \in R^n \ \mathit{for} \sqrt{\sum_{i=1}^{n} X_i^2} \leq u \right\}.$$

Above we used the notation:

$$X_i = \sum_{j=1}^{n} x_{ij}, i = 1,2,\mathrm{K}\ , n, \ \ X = (X_1, X_2, \mathrm{K}\ , X_n)^T \text{, and } u = \sum_{i=1}^{n} X_i \ .$$

Thus, we can reformulate the problem (1) - (6) in terms of second-order cone programming:

$$z \rightarrow \mathit{minimize} \ ,$$

subject to restrictions (2) - (5), and

$$\sqrt{\sum_{i=1}^{n} y_i^2} - z \leq 0,$$

$$\sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}^2} - u \leq 0,$$

$$\sum_{i=1}^{n} y_i - z = 0,$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} - u = 0,$$

$$0 \le y_i \le 1, i = 1,2,K\ ,n,$$

$$0 \le x_{ij} \le 1, i, j = 1,2,K\ ,n.$$

4 Conclusions

The paper contains a reformulation of the simple assembly line balancing problem in terms of second-order cone programming. This problem can be effectively solved by the interior point algorithm ([7]). There is specialized software for solving conic optimization problems ([8]).

References

[1] I. Baybars. A survey of exact algorithms for the simple line balancing problem. *Management Science*, *32*, 909-932, 1986. http://www.jstor.org/stable/2631657.

[2] A. School, *Balancing and Sequencing of Assembly Lines (series: Contributions to Management Science),* Physica-Verlag Heidelberg, 2-nd edition, 1999.

[3] T. K. Bhattachaje, S. Sahn. Complexity of Single Model Assembly Line Balancing Problems. *Engineering Cost and Production Economics*, *18*, 203-214, 1990.

[4] A. School, C. Becker. State of the Art Exact and Heuristic Solution Procedures for Simple Assembly Line Balancing. *European Journal of Operational Research*, *Vol. 168*, No. 3, 666-693, 2006.

[5] A. L. Arcus. COMSOAL a Computer Method of Sequencing Operations for Assembly Lines. *The International Journal of Production Research, Vol. 4,* No. 4, 259-277, 1996.

[6] E. Erel, S. C. Sarin. A Survey of the Assembly Line Balancing Procedures. *Production Planning and Control, Vol. 9,* No. 5, 414-434, 1998.

[7] Yu. Nesterov, A. Nemirovsky. Interior-point polinomial methods in convex programming. *Volume 13 of Studies in Applied Mathematics.* SIAM, Phildelphia, PA, 1994.

[8] H. D. Mittelmann. The State-of-the-Art in Conic Optimization Software. *In Handbook on Semidefinite, Conic and Polynomial Optimization, Volume 166*, 671-686, 2012.

Vasile Moraru
Technical University of Moldova
Applied Informatics Department
168, Stefan cel Mare str., Chisinau, 2004
MOLDOVA Republic of
E-mail: moraru@mail.utm.md

Sergiu Zaporojan
Technical University of Moldova
Computer Science Department
168, Stefan cel Mare str., Chisinau, 2004
MOLDOVA Republic of
E-mail: zaporojan_s@yahoo.com