Some Complexity Results for the Simple Assembly Line Balancing Problem

Evgeny R. Gafarov[1], **Alexandre Dolgui**[2], **Alexander Lazarev**[3]

[1] *Ecole Nationale Superieure des Mines, FAYOL-EMSE, CNRS:UMR6158, LIMOS, F-42023 Saint-Etienne, France;* `axel73@mail.ru`
[2] *Ecole Nationale Superieure des Mines, FAYOL-EMSE, CNRS:UMR6158, LIMOS, F-42023 Saint-Etienne, France;* `dolgui@emse.fr`
[3] *Institute of Control Sciences of the Russian Academy of Sciences, Profsoyuznaya st. 65, 117997 Moscow, Russia;* `jobmath@mail.ru`

Introduction

We consider the simple assembly line balancing problem (SALBP-1) which is formulated as follows.

Given a set $N = \{1, 2, \dots, n\}$ of operations and K stations (machines) $1, 2, \dots, K$. For each operation $j \in N$ a processing time $t_j \geq 0$ is defined. The cycle time $c \geq \max\{t_j,\ j \in N\}$ is given. Furthermore, finish-start precedence relations $i \to j$ are defined between the operations according to an acyclic directed graph G. The objective is to assign each operation $j,\ j = 1, 2, \dots, n$, to a station in such a way that:
- number $m \leq M$ of stations used is minimized;
- for each station $k = 1, 2, \dots, m$ a total load time $\sum_{j \in N_k} t_j$ does not exceed c, where N_k – a set of operations assigned to a station k;
- given precedence relations are fulfilled, i.e. if $i \to j,\ i \in N_{k_1}$ and $j \in N_{k_2}$ then $k_1 \leq k_2$.

A survey on results for NP-hard in the strong sense SALBP-1 is presented, e.g., in [1,2].

Partition problem. Given is a set $N = \{b_1, b_2, \dots, b_n\}$ of numbers $b_1 \geq b_2 \geq \dots \geq b_n > 0$ with $b_i \in Z_+,\ i = 1, 2, \dots, n$, and a number $A \in Z_+$ with $A < \sum_{j \in N} b_j$. Is there a subset $N' \subset N$ such that $\sum_{j \in N'} b_j = A$?

The worst case running time of $B\&B$ algorithms for the well-known Knapsack Problem is analyzed, e.g., in [4]. In these papers authors choose to use only special cases, for which it is fairly easy to find an optimal solution with a $B\&B$ algorithm. However to prove its optimality, almost all feasible solutions should be considered. We use a similar idea to construct a special case of SALBP-1 for which each $B\&B$ algorithm with no matter what polynomial time computed Lower Bound has an exponential run time. This makes algorithms ineffective for instances with $n \geq 60$ operations.

Modified instance of the Partition problem. Given is a set $\overline{N} = \{\overline{b}_1, \overline{b}_2, \ldots, \overline{b}_{2n}\}$ of numbers $\overline{b}_1 \geq \overline{b}_2 \geq \cdots \geq \overline{b}_{2n} > 0$ with $\overline{b}_i \in Z_+,\ i = 1, 2, \ldots, 2n$, and a number $\overline{A} \in Z_+$ with $\overline{A} < \sum_{j \in \overline{N}} \overline{b}_j$. The numbers $\overline{b}_i,\ i = 1, 2, \ldots, 2n$ are denoted as follows:

$$\overline{b}_{2n} = 1,\ \overline{b}_{2i} = 2 \cdot \sum_{j=i+1}^{n} \overline{b}_{2j-1},\ i = n-1 \ldots, 1,\ \overline{b}_{2i-1} = \overline{b}_{2i} + b_i,\ i = n \ldots, 1,$$

where $b_1, b_2, \ldots, b_n$ – numbers from the initial instance. Let $\overline{A} = \sum_{i=1}^{n} \overline{b}_{2i} + A$. Without lost of generality let us assume $A = \frac{1}{2} \sum_{i=1}^{n} b_i$ and as consequence $\overline{A} = \frac{1}{2} \sum_{i=1}^{2n} \overline{b}_i$. The question is: "Is there a subset $\overline{N}' \subset \overline{N}$ such that $\sum_{j \in \overline{N}'} \overline{b}_j = \overline{A}$"? If for the initial instance of the Partition Problem the answer is "YES" (and the same answer has the modified instance) then $\overline{N'}$ contains one and only one number $\overline{b}^i$ from each pair $\{\overline{b}_{2i-1}, \overline{b}_{2i}\},\ i = 1, 2, \ldots, n$. If the number b_i is included in the set N' then $\overline{b}_{2i-1}$ is included in $\overline{N'}$, otherwise the number $\overline{b}_{2i} \in \overline{N'}$.

In the special case of SALBP-1 there are $2n$ operations. Let $w' = \min\{w | 10^w \geq 2\overline{A}\}$. Let us

$$t_i = 10^{w'} + \overline{b}_i,\ i = 1, 2, \ldots, 2n$$

and $c = \frac{1}{2} \sum_{i=1}^{2n} t_i$. There are no precedence relations between operations.

It is obvious that if and only if for the modified instance of the Partition Problem the answer is "YES" then the minimal umber of stations $m^* = 2$, otherwise $m^* = 3$. As a consequence, if $NP \neq P$, there is no polynomial time computed Lower Bound with a relative error equal or less than $\frac{3}{2}$. That means, for any set of polynomial time computed Lower Bounds $\{LB_1, LB_2, \ldots, LB_X\}$, there is a modified instance of the Partition Problem with an answer "NO", for which $LB_x = 2,\ i = 1, 2, \ldots, X$, although $m^* = 3$. For the special case of SALBP-1, any feasible solution is optimal. However, to prove its optimality almost all feasible solutions must be considered.

Let us estimate the possible number of feasible solutions. On the first station there could be processed at least $n-1$ operations. Thus, there are at least $\binom{2n}{n-1}$ possible loads of the first station, i.e. the number of feasible solutions which have to be considered is greater than $\binom{2n}{n-1} = \frac{n+1}{n}\binom{2n}{n} \approx \frac{n+1}{n} \cdot \frac{2^{2n}}{\sqrt{n\pi}}$. To solve such the instance of SALBP-1 with $2n = 60$ a computer must perform more than $\frac{2^{60}}{10}$ operations. Let us assume that the fastest known computer performs 2^{30} operations per second, or less than 2^{47} operations per day. Then a run time of an algorithm will be more than $\frac{2^{13}}{10} > 800$ days! That means there are instances of SALBP-1 for which any $B\&B$ algorithm with polynomial time computed Lower Bounds has an unappropriate running time.

We can conclude the following. Despite the best known algorithm $B\&B$ [3] solves all benchmark instances in less than 1 second per instance, known $B\&B$ algorithms for SALBP-1 remain exponential and can not solve some instances with the size $n > 60$ in an appropriate time. That is why we consider exact algorithms for the general case of the problem unpromising. Researchers can concentrate on special cases or on essentially new solution schemes.

Maximization of Number of Stations

To propose an essentially new solution scheme for SALBP-1, it is necessary to investigate properties of optimal solutions. We can investigate not only properties of good solutions to try imitate their character but properties of poor solutions as well to avoid solutions with their aspects. Here, in contrast to standart SALBP-1, where the number of stations used should be minimized we consider an optimization problem with the opposite objective criteria, in other words the maximization of the number of stations. The investigation of a particular problem with the *maximum* criterion is an important theoretical task [5]. To make the maximization problem not trivial we assume that all stations (instead the last one) should be maximal loaded, i.e. for two stations $m_1, m_2,\ m_1 < m_2$ there is no operation j assigned on the station m_2 which can be assigned on station m_1 without violation of precedence constraints or the feasibility's condition "total load time of the station does not exceed the cycle time". Denote the maximization problem by $max - SALBP - 1$.

Theorem 1. *max-SALBP-1 is NP-hard in the strong sense (by reduction from the 3-Partition Problem).*

Theorem 2. *max-SALBP-1 is not approximated with an approximation ratio* $\leq \frac{3}{2}$ *unlike* $P = NP$.

An experimental study of maximal number of stations for benchmark instances published on http://www.assembly-line-balancing.de was done. The results show that the maximal founded deviation $m^{max} - m^{min}$ does not exceed 20%.

Flat Graph of Precedence Relations

In [6] authors propose a transformation of graph G of precedence relations to planar one for the well-known Resource-Constrained Project Scheduling Problem. The same idea can be used for SALBP-1.

Theorem 3. *For any instance of SALBP-1 with* n *operations and* v *precedence relations, there exists an analogous instance with a flat graph* G' *with* n' *operations and* v' *relations, where* $n + v \geq n' + v'$.

We obtain an analogous instance from the original one by adding "dummy" operations (with $t_j = 0$) and deleting all the unnecessary relations. According to the well-known Euler's Theorem, $v' \leq 3n' - 6$ in such the planar graph.

The number of precedence relations influences running time and the theoretical complexity of solution algorithms. The number of precedence relations is estimated by different authors as $O(n^2)$ (i.e, the "Order strength" on http://www.assembly-line-balancing.de is estimated according to the number $n \cdot (n-1)$ of precedence relations). If we consider only instances with planar graphs then the number of relations is $\leq 3n - 6$, i.e. $O(n)$. So, the fact mentioned in Theorem 3 allows us to reduce the run time of algorithms (by reduction of unnecessary relations) and estimate the complexity exacter.

The authors were supported by the Russian Foundation for Basic Research (projects no. 11-08-01321, 11-08-13121).

REFERENCES

1. *E. Erel, S.C. Sarin,* A survey of the assembly line balancing procedures, Production Planning and Control, 9, 414434 (1998).
2. *A. Scholl,* Balancing and Sequencing of Assembly Lines, Physica Verlag, A Springer-Verlag Company, 1999.
3. *E.C. Sewell, S.H. Jacobson,* A Branch, Bound and Remember Algorithm for the Simple Assembly Line Balancing Problem, INFORMS Journal on Computing, doi 10.1287/ijoc.1110.0462, 1–10 (2011)

4. *M.A. Posypkin , I. Kh. Sigal ,* Speedup estimates for some variants of the parallel implementations of the branch-and-bound method, Computational Mathematics and Mathematical Physics, 46, N 12, 2189 –2 202 (2006).
5. *M.A. Aloulou , M.Y. Kovalyov, M.-C. Portmann,* Maximization Problems in Single Machine Scheduling, Annals of Operations Research, 129, 21 – 32 (2004).
6. *A.A. Lazarev, E.R. Gafarov,* Transformation of the Network Graph of Scheduling Problems with Precedence Constraints to a Planar Graph, Doklady Mathematics, 79 N 1, 1-3 (2009).