Multi-Objective Robotic Assembly Line Balancing Problem: A NSGA-II Approach Using Multi-Objective Shortest Path Decoders

Lahrichi Y., Deroussi L., Grangeon N. and Norre S.

LIMOS-CNRS, Université Clermont Auvergne, Aubière, France
{youssef.lahrichi,laurent.deroussi,nathalie.grangeon,sylvie.norre}@uca.fr

Keywords: Robotic Assembly Line Balancing Problem, Multi-objective optimization, NSGA-II, Multi-objective decoder, Shorthest path.

1 Problem statement

The Robotic Assembly Line Balancing Problem (RALBP) is a combinatorial optimization problem that is concerned with simultaneously assigning a set of operations to a set of workstations placed among a serial assembly line and assigning to each workstation a type of robot. The processing time of an operation i depends on the type of robot r used and is denoted d_i^r. Each type of robot r is also characterised by its cost c_r. Besides, operations are linked by precedence relations. The workload of a workstation represents the sum of the processing times of the operations assigned to it. The cycle time stands for the maximum workload among the stations and is a key performance indicator of the assembly line.

The RALBP is of significant importance due to the growing robotization of assembly lines. In this study, we consider an additional parameter which is sequence-dependent setup times: in addition to processing times, setup times $t_{i,i'}^r$ should be considered if operation i is performed just before operation i' in some workstation equipped by a robot of type r. The workload of a workstation is the sum of processing times and sequence-dependent setup times induced by the operations assigned to it. Sequence-dependent setup times raises an additional decision which is the sequencing of operations in each workstation. Sequence-dependent setup times have been rarely considered in literature for the RALBP despite their industrial importance. We study the problem while minimizing simultaneously three objectives:

(Z_1): Minimizing cycle time,
(Z_2): Minimizing the total cost of robots used,
(Z_3): Minimizing the number of workstations used.

The problem has been introduced in Rubinovitz et al. (1993) where the basic assumptions are presented. Most authors consider the single objective of minimizing cycle time as Nilakantan et al. (2015) and Borba et al. (2018) and few perform multi-objective study (Yoosefelahi et al. (2012), Çil et al. (2016)). Sequence-dependent setup times have not been considered until very recently in Janardhanan et al. (2019). Table 1 positions our study in literature.

2 Example

We illustrate the problem with a small instance. We consider 8 operations and 3 types of robots. Precedence relations are illustrated in the precedence graph (Fig. 1). Processing times and sequence-dependent setup times are supposed given. A feasible solution is depicted in Fig. 2.

Table 1. Position of our study in the literature.

Article	Objectives Z_1	Z_2	Z_3	Sequence-dependent setup times
Rubinovitz et al. (1993)			✓	
Levitin et al. (2006)	✓			
Gao et al. (2009)	✓			
Yoosefelahi et al. (2012)	✓	✓		
Nilakantan et al. (2015)	✓			
Çil et al. (2016)	✓	✓	✓	
Borba et al. (2018)	✓			
Janardhanan et al. (2019)	✓			✓
Our study	✓	✓	✓	✓

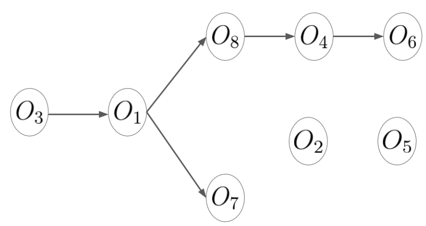

Fig. 1. Precedence graph

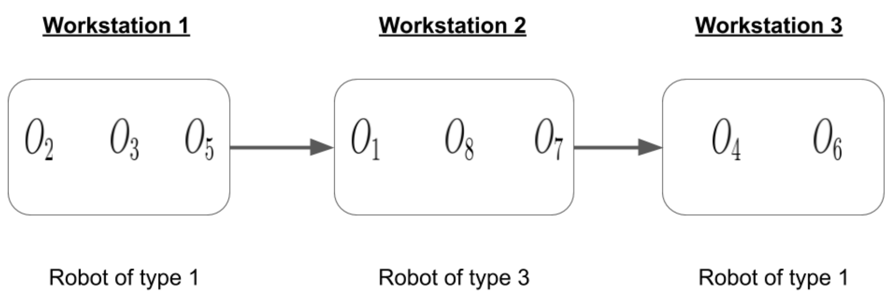

Fig. 2. Feasible solution

Let's compute the cost of this solution:

(Z_1) The total cost of robots used:

$$Z_1 = c_1 + c_3 + c_1$$

where c_1 is the cost of a robot of type 1 and c_3 is the cost of a robot of type 3.

(Z_2) Cycle time : Its is obtained by computing the maximum among the workloads of the workstations:

– On the first workstation, the workload is given by:

$$d_2^1 + t_{2,3}^1 + d_3^1 + t_{3,5}^1 + d_5^1 + t_{5,2}^1$$

– On the second workstation, the workload is given by:

$$d_1^3 + t_{1,8}^3 + d_8^3 + t_{8,7}^3 + d_7^3 + t_{7,1}^3$$

– On the third workstation, the workload is given by:

$$d_4^1 + t_{4,6}^1 + d_6^1 + t_{6,4}^1$$

(Z_3) The number of workstations used:

$$Z_3 = 3$$

3 Optimization method

We first derive a pseudo-polynomial time exact algorithm to compute all the Pareto optimal solutions provided the giant sequence of operations is given (in Figure 2, the giant sequence is $O_2, O_3, O_5, O_1, O_8, O_7, O_4, O_6$). We use then this algorithm as a decoder in a NSGA-II metaheuristic. For this purpose, we suggest a generalization of NSGA-II that supports multi-objective decoders.

3.1 Fixed giant sequence

We suppose that the giant sequence σ is fixed. Without lose of generality, we suppose $\sigma = (1, 2, \ldots, n)$. The problem is equivalent to finding a multi-criteria shortest path in an auxiliary graph $\mathcal{H}_{\mathcal{I}}(\sigma)$.

$\mathcal{H}_{\mathcal{I}}(\sigma) = (V, A)$ is a bi-valued oriented multi-graph. $V = \{0, 1, \ldots, n\}$ is the set of vertices. Vertex i $(i > 0)$ represents operation O_i while vertex 0 is fictitious. A is the multi-set (each element can have several duplicates) of arcs. It contains all arcs from i to j where $i < j$. An arc (i, j) represents a workstation to which the sequence of operations $O_{i+1}, O_{i+2}, \ldots, O_j$ is assigned. Each arc (i, j) is duplicated as many times as there are robots type r. A duplicate of (i, j) for robot type r is denoted $(i, j)_r$. The graph is bi-valued, each arc $(i, j)_r$ is weighted in $\mathbb{R}^2$ as follows:

$$w((i,j)_r) = [c_r, \sum_{k=i+1}^{j} d_k^r + \sum_{k=i+1}^{j-1} t_{k,k+1}^r + t_{j,i+1}^r]$$

The first weight represents the cost of the robot used in the workstation and the second weight represents the workload of the workstation, which means the duration to perform the sub-sequence $(i + 1, \ldots, j)$ with a robot of type r.

A path from vertex 0 to vertex n stands for a feasible solution of the balancing sub-problem. Solving optimally the multi-objective balancing problem given a giant sequence σ can be done by solving the multi-objective shortest path problem from vertex 0 to vertex n in $\mathcal{H}_{\mathcal{I}}(\sigma)$ minimizing simultaneously the sum of the first weights among the path, the sum of the second weights among the path and the number of arcs in the path.

All Pareto optimal solutions for the multi-objective shortest path problem can be computed thanks to a pseudo-polynomial algorithm (which we denote Split) within $O(c_{max}.N_r.n^3)$ where c_{max} is the maximum cost of a robot, N_r the number of robot types and n the number of operations.

3.2 General case

We derive an approximate method embedding the split to solve the problem in the general case where the giant sequence is not given. To encode a solution, we use a giant sequence. The split algorithm is used to decode a giant sequence. The general scheme of the metaheuritic is a NSGA-II (Non-dominated Sorting Genetic Algorithm). However, NSGA-II does not support multi-objective decoders, i.e decoders yielding several non-dominated solutions. For this reason we suggest a novel generalization of NSGA-II supporting multi-objective decoders.

4 Conclusion

In this study, a pseudo-polynomial algorithm is presented for solving the RALBP with sequence-dependent setup times given a fixed giant sequence in a multi-objective context. Then we derive an approximate method for solving the problem in the general case using a novel generalization of NSGA-II. Experiments are actually being held and the first results are encouraging.

Acknowledgements

The authors acknowledge the support received from the *Agence Nationale de la Recherche* of the French government through the program "Investissements d'Avenir"(16-IDEX-0001 CAP 20-25).

References

Borba, Leonardo, Marcus Ritt, and Cristóbal Miralles. 2018. "Exact and heuristic methods for solving the robotic assembly line balancing problem." *European Journal of Operational Research* 270 (1): 146–156.

Çil, Zeynel Abidin, Süleyman Mete, and Kürşad Ağpak. 2016. "A goal programming approach for robotic assembly line balancing problem." *IFAC-PapersOnLine* 49 (12): 938–942.

Gao, Jie, Linyan Sun, Lihua Wang, and Mitsuo Gen. 2009. "An efficient approach for type II robotic assembly line balancing problems." *Computers & Industrial Engineering* 56 (3): 1065–1080.

Janardhanan, Mukund Nilakantan, Zixiang Li, Grzegorz Bocewicz, Zbigniew Banaszak, and Peter Nielsen. 2019. "Metaheuristic algorithms for balancing robotic assembly lines with sequence-dependent robot setup times." *Applied Mathematical Modelling* 65:256–270.

Levitin, Gregory, Jacob Rubinovitz, and Boris Shnits. 2006. "A genetic algorithm for robotic assembly line balancing." *European Journal of Operational Research* 168 (3): 811–825.

Nilakantan, J Mukund, Sivalinga Govinda Ponnambalam, N Jawahar, and Ganesan Kanagaraj. 2015. "Bio-inspired search algorithms to solve robotic assembly line balancing problems." *Neural Computing and Applications* 26 (6): 1379–1393.

Rubinovitz, Jacob, Joseph Bukchin, and Ehud Lenz. 1993. "RALB–A heuristic algorithm for design and balancing of robotic assembly lines." *CIRP annals* 42 (1): 497–500.

Yoosefelahi, A, M Aminnayeri, H Mosadegh, and H Davari Ardakani. 2012. "Type II robotic assembly line balancing problem: An evolution strategies algorithm for a multi-objective model." *Journal of Manufacturing Systems* 31 (2): 139–151.