A hybrid genetic algorithm for assembly line re-balancing problem with assignment restrictions

BELASSIRIA IMAD[1], MAZOUZI MOHAMED[1], ELFEZAZI SAID[2]

1. Laboratoire de Contrôle et de Caractérisation des Matériaux, Génie Mécanique – ENSEM / Casablanca, Maroc, (email : imad.belassiria@gmail.com, d.liauto@gmail.com)
2. Laboratoire Analyse des Processus, Signaux, Systèmes Industriels et Informatiques – EST / Safi, Maroc, (email : selfezazi@gmail.com)

Abstract

In this paper, we propose a hybrid genetic algorithm to solve assembly line re-balancing problem. There are two objectives to be achieved: minimizing the number of workstations (number of operators) for a given cycle time and balancing the workstation simultaneously. The model provide more realistic situation of assembly line re-balancing problem with assignment restriction. The genetic algorithm may lack the capability of exploring the solution space effectively, so we aim to provide its exploring capability by sequentially hybridizing the well-known assignment rules heuristics with genetic algorithm.

Keywords: *assembly line re-balancing problem; genetic algorithm; hybrid; assignment rules heuristics; assignment restriction;*

1. Introduction

Assembly lines have gained great importance in manufacturing of high quantity standardized products particularly automobile manufacturing [1]. An assembly line consists of a sequence of workstations in which components are consecutively added to create one semi-finished assembly, this one moves from one station to the next station until the final assembly is produced. In order to meet required performance, the assembly line needs to be balanced by assigning tasks to workstation in such a way that the assembly objective is fulfilled, the demand is met and the constraints are satisfied. Therefore, an assembly line balancing is an effective tool for improving productivity and increasing efficiency.

There are two-known types of assembly line balancing problems (ALBPs) [2], ALBP-I intends to assign tasks to workstations such that the number of stations is minimized for a pre-specified cycle time while ALBP-II aims to minimize the cycle time, or equivalently, maximizes the production rate for a specific number of stations. Both versions of the problems are NP-hard [3].

In real-world line balancing some disturbing events can occur and change the characteristics of the line. However, due to the changes on productions or processes, tools and resources of the existing assembly lines need to be reused or adapted. For this reason in industrial practices the majority of assembly line balancing is conducted for reconfiguration instead of first time installation. Furthermore, the balancing problem covers many planning horizons depends to the disturbances nature [4], see Fig. 1. Indeed, the re-engineering of the assembly line concerns major changes on the structure of the line or the introduction of new products. For the re-balancing problem, Grangeand et al. [5] takes into account the production demand changes and some structure like addition or removal of workstations. Dealing with the short-term planning, the dynamic rebalancing problem as discussed in Antoine et al. [6] could concern delays, breakdowns, temporary shortage and availability of workforces.

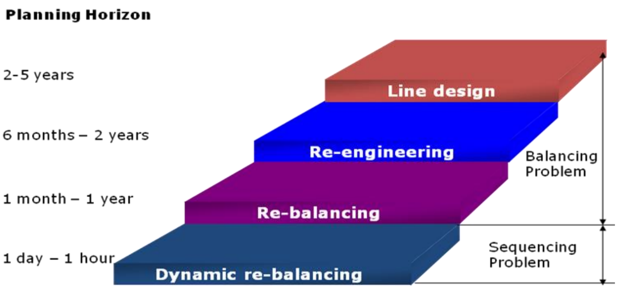

Fig. 1. Balancing problem through planning horizons

In the reconfiguration of existing assembly lines there are some cases that you cannot avoid assignment restrictions. For example, some workstations have its own heavy equipment, therefore all tasks that required the equipment must be assigned to a given workstation due to the huge costs to move the heavy equipment elsewhere in the shop. The most of researchers address

ALBPs by classical restrictions like precedence constraints which led to ignore many aspects of the real-world problem, in this case those solutions becomes unusable in the industry. In this paper, more realistic situations are provided to deal with manual ALBPs.

Solutions are often classified into two categories: exact or heuristic. As ALBP is NP-hard and the required computational time for obtaining an optimal solution with an exact method for most of line balancing problems increases exponentially with the size of instance considered, as a consequence, metaheuristic methods are clearly needed in order to cope with large scale cases [7].

Genetic algorithm, as one of metaheuristics approaches, has been widely applied to solve assembly line balancing problems. In the real world line balancing an additional difficulty has presented, although meta-heuristics may lack the capability of exploring the solution space effectively as problems get larger and more complex as in real life. To improve upon this issue we develop a hybrid meta-heuristics which provide individuals for initial population based on heuristic methods instead of individuals randomly generated. So that, in this paper we present a new hybridization of Genetic algorithm and task assignment heuristics which consists in such a way that an individual represents one rule (heuristic) from the list of task assignment rules (see TABLE I), to be used to construct a feasible solution. In result, we will have ten feasible solutions based priority rules which are sequentially hybridized with GA.

TABLE I. LIST OF TASK ASSIGNMENT RULES

Rule no.	Task assignment rules
1	Shortest Processing Time (SPT)
2	Longest Processing time (LPT)
3	Minimum Total Number of Successor Tasks (MiTNST)
4	Maximum Total Number of Successor Tasks (MaTNST)
5	Minimum Total Time of Successor Tasks (MiTTST)
6	Maximum Total Time of Successor Tasks (MaTNST)
7	Minimum Total Number of Predecessor Tasks (MiTNPT)
8	Maximum Total Number of Predecessor Tasks (MaTNPT)
9	Minimum Total Time of Predecessor Tasks (MiTTPT)
10	Maximum Total Time of Predecessor Tasks (MaTTPT)

2. Formulation of the ALBP-I

The following assumptions are also stated to clarify the setting in which the problem arises:

- ZP is the set of tasks must be assigned into the same workstation and ZN is the set of tasks must be assigned into different workstations.
- Station restriction : specific tasks need to be assigned into specific worstations.
- The factory contains k assembly lines; each line produces one kind of product.
- Each line includes m workstation and n tasks.
- One operator can be assigned to one workstation.
- The rebalancing frequency is depend on disturbance events occur on the assembly line.
- The precedence graph is given.

The objective function of the problem is to minimize the number of workstations (i.e., the number of operators) for a given cycle time and to balance the workstation simultaneously. Minimizing the number of workstations is equivalent to maximization of the assembly line efficiency that is reducing the idle time as much as possible.

The assembly line efficiency is formulated as

$$\text{WE} = \frac{\sum_{i=1}^{n} t_i}{m \times CT}$$

The average idle time for a single workstation is calculated as

$$\bar{s} = \frac{\sum_{j=1}^{m} s_j}{m}$$

The workload balance between workstation is given by

$$B = \frac{\sum_{j=1}^{m} |s_j - \bar{s}|}{\sum_{j=1}^{m} s_j}$$

Then, the objective function is formulated as

$$\text{Max } F = \alpha.\text{WE} - \beta.\text{B}$$

3. Proposed hybrid genetic algorithm (hGA)

Flow diagram of the proposed hGA is depicted in fig. 2.

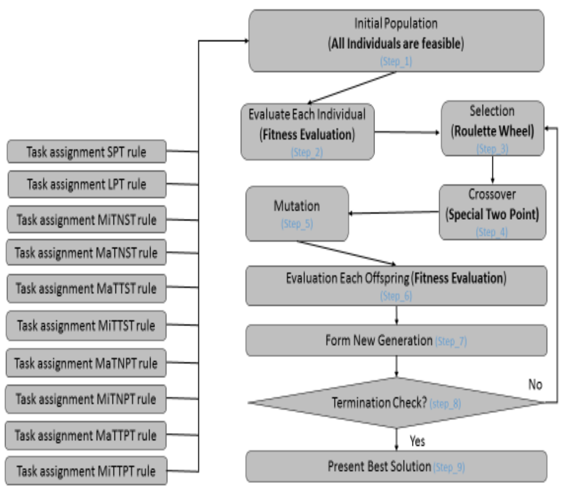

Fig. 2. Flow diagram of the proposed hybrid genetic algorithm

4. Industrial application

In order to validate the proposed hGA for rebalancing problem, an industrial case from wire harness manufacturer is used. The objective is to test the method and get it applicable on real problem. In this case we apply hGA to one line which has the following characteristics:

- The productivity requested is 250 items per day, and the cycle time is pre-defined CT = 120 s.
- The number of tasks performed in the line is 45 tasks and the precedence graph was done.
- Zoning constraint: Tasks type (INS) and (RET) cannot be executed on the same workstation.
- Station restriction: The task can only be assigned into specific workstation.
- The maximum number assigned to workstation is limited to 4 tasks.

Workstation No	Tasks	Workstation Time	Idle Time
WS 001	470-450-650-550-640-380-130	120	0
WS 002	100-420-310	110	10
WS 003	170-080-120-140-200	120	0
WS 004	330-260-390-180	105	15
WS 005	270-440-410-400	105	15
WS 006	520-480-540	105	15
WS 007	300-280-290	105	15
WS 008	210-220-430-360	120	0
WS 009	370-240-230	105	15
WS 010	560	109	11
WS 011	350-250-610	80	40
WS 012	570	109	11
WS 013	580	115	5
WS 014	600	100	20
WS 015	590	90	30
WS 016	620	120	0

Number of Workstation = 016
Efficiency = 89.5%
Workload Balance = 0.634

Fig. 3. Task assignment representation of the best solution

The initial population is determined by using heuristic method (task assignment heuristic) which creates ten candidates according the number of rules.

Fig.3 displays the chromosome for the best solution among the ten candidates included in the initial population.

The first column of the table denotes the workstation index, the second column the set of tasks assigned to the workstation, the third column is the total time of each workstation, and the fourth is the idle time for each workstation. The small table beside shows the total numbers of workstations required by the solution, and the assembly line efficiency and within-station the workload balance value. As it's shown in Fig.3, the number of workstation found by this solution is 16, and the WE and B are found as 89.5% and 0.634 respectively.

For the next step, and in order to find out the optimal reconfiguration of the assembly line, we are going to implement genetic algorithm mechanism which contains fitness evaluation and selection, genetic operators and creation of new generation following the steps mentioned in Fig. 2.

The proposed hGA is going to be coded in C# 6.0 (visual studio 2013), and the optimal solution will be found soon. The value of genetic parameters used for the hGA, crossover rate = 0.5 and mutation rate = 0.15. The hGA was terminated when the total number of iterations exceeds a predefined number (250 iterations).

5. Conclusions:

In the present work, a hybrid genetic algorithm was proposed in order to improve the actual line-balancing problem inspired by an industrial case of wire harness manufacturer, and help the line manager to rebalance the assembly line against changes with assignment restriction. Besides minimization of number of workstations, maximizing workload smoothness between workstations was also considered. The hybridization was realized by inserting the solutions obtained by the ten assignment rules heuristics as initial population of genetic algorithm. For the next step of this work that will be done soon, we are going to optimize the re-balancing problem by using genetic operators and evaluate the performance of the proposed hGA.

References

[1] Bagher, M., Zandieh, M. & Farsijani, H. Balancing of stochastic U-type assembly lines: an imperialist competitive algorithm. Int J Adv Manuf Technol (2011) 54: 271.

[2] Nima Hamta, S.M.T. Fatemi Ghomi, F. Jolai, Unes Bahalke. Bi-c criteria assembly line balancing by considering flexible operation times. Applied Mathematical Modeling,Volume 35, Issue 12, December 2011, Pages 5592-5608.

[3] R. W. Sierenberg. An Algorithm for the Line Balancing Problem. Volume 1971 of the series Proceedings in Operations Research pp 722-744.

[4] Scholl, A., & Scholl, A. (1999). Balancing and sequencing of assembly lines.

[5] Grangeon, N., Leclaire, P., & Norre, S. (2011). Heuristics for the re-balancing of a vehicle assembly line. International Journal of Production Research, 49(22), 6609-6628.

[6] Antoine, M., Hind, B. E. H., Wahiba, R. C. K., & Lounes, B. M. (2016). Iterated Local Search for dynamic assembly line rebalancing problem. IFAC-PapersOnLine, 49(12), 515-519.

[7] Battaïa, O., & Dolgui, A. (2013). A taxonomy of line balancing problems and their solutionapproaches. International Journal of Production Economics, 142(2), 259-277.