Open access • Journal Article • DOI:10.1080/00207540802176046

A multiple single-pass heuristic algorithm solving the stochastic assembly line rebalancing problem — Source link

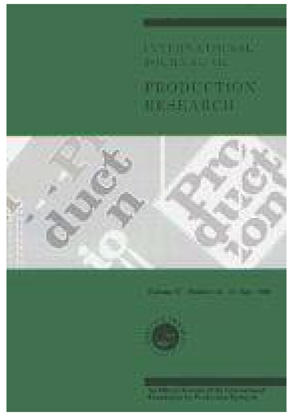
Rita Gamberini, Elisa Gebennini, Andrea Grassi, Alberto Regattieri

Institutions: University of Modena and Reggio Emilia, University of Bologna

Published on: 27 Feb 2009 - International Journal of Production Research (Taylor & Francis)

(19th ICPR) A multiple single-pass heuristic algorithm solving the stochastic assembly line re-balancing problem

Rita Gamberini, Elisa Gebennini, Andrea Grassi, Selcuk Sen



(19th ICPR) A multiple single-pass heuristic algorithm solving the stochastic assembly line re-balancing problem

Journal:	*International Journal of Production Research*
Manuscript ID:	TPRS-2008-IJPR-0128.R1
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	23-Apr-2008
Complete List of Authors:	Gamberini, Rita; University of Modena and Reggio Emilia, Engineering Faculty, Department of Engineering Sciences and Methods Gebennini, Elisa; University of Modena and Reggio Emilia, Engineering Faculty, Department of Engineering Sciences and Methods Grassi, Andrea; University of Modena and Reggio Emilia, Engineering Faculty, Department of Engineering Sciences and Methods Regattieri, Alberto; University of Bologna, Engineering Faculty, Department of Industrial Mechanical Plants
Keywords:	ASSEMBLY LINE BALANCING, HEURISTICS
Keywords (user):	ASSEMBLY LINE BALANCING, HEURISTICS

A multiple single-pass heuristic algorithm solving the stochastic assembly line re-balancing problem

R. Gamberini[1], E. Gebennini, A. Grassi, A. Regattieri[§]

Department of Engineering Sciences and Methods, University of Modena and Reggio Emilia, Via Amendola 2 – Padiglione Morselli, Reggio Emilia, Italy

[§]Department of Industrial Mechanical Plants, University of Bologna, Viale Risorgimento 2, Bologna, Italy

[1] Corresponding author:
R. Gamberini
Tel.: +39 0522 522 633
Fax: +39 0522 522 609
E-mail: rita.gamberini@unimore.it

A multiple single-pass heuristic algorithm solving the stochastic assembly line re-balancing problem

Abstract

Assembly line re-balancing is a problem companies are frequently confronted with as continuous changes in product features and volume demand caused by the volatility of modern markets result in re-definition of assembly tasks and line cycle time fluctuations. Consequently, managers are forced to adjust the balancing of their lines in order to adapt to the new conditions while trying to minimize both increases in completion costs and costs related to changes in task assignment. In particular, when modifications are made to line balancing, costs are incurred for operator training, equipment switching and moving, and quality assurance. The stochastic assembly line re-balancing problem is essentially composed of a multi-objective problem in which two joint objectives, total expected completion cost of the new line and similarity between the new and the existing line, must be optimized. Consequently, this paper presents a multiple single-pass heuristic algorithm developed for the purpose of finding the most complete set of dominant solutions representing the Pareto front of the problem. The operative parameters of the heuristic are set as a result of a great deal of experimentation. Moreover, a multi objective genetic algorithm is developed and then compared with the proposed heuristic in order to demonstrate its effectiveness. Finally, an illustrative case study is presented.

Keywords: assembly line balancing problem, re-balancing, multi-objective

1. Introduction

Assembly line balancing optimization is one of the most pressing problems companies must solve as most current production systems present a final assembly phase where components are joined together to form the final product. Balancing procedures optimize assembly line behavior by defining the workload of stations in order to satisfy both cycle time and technological precedence relationship constraints in addition to optimizing several measurements of performance such as line efficiency, the number of operators engaged on the line, and the productivity rate.

Since Salveson (1955) formulated the Assembly Line Balancing Problem (ALBP), a wide variety of solving procedures have been introduced addressing either deterministic or stochastic task execution times, on either single-model or multi/mixed-model lines, and in either paced or un-paced systems.

Nevertheless, most papers disregard the problem of implementing frequent re-definitions of station workload in order to meet market requests for variations in product characteristics and/or productivity rate. Otherwise, dynamic competitive environments address the definition of agile assembly systems able to respond to unpredictable events efficiently, since variations in customer requirements involve frequent line re-balancing, that is, changes in station workloads.

When manual assembly lines are considered, re-definitions of station workloads are implemented by involving the re-assignment of tasks to operators engaged at each station. Consequently, operators need to be trained in how to perform the new tasks required. Nevertheless, the learning process requires time and involves errors during initial phases, hence increasing costs both in training activities and in terms of equipment switching, quality assurance, and line performances. In fact, these costs are directly related to the number of task re-assignments.

This paper presents a study concerning innovative heuristic algorithms for solving the assembly line re-balancing problem for the purpose of minimizing both assembly cost and task re-assignments produced by changes in product characteristics.

2. Literature review

The wide variety of ALBP solving procedures has been recently reviewed by Ghosh and Gagnon (1989), Erel and Sarin (1998), Amen (2000, 2006), Pierreval *et al.* (2003), Becker and Scholl (2006), Scholl and Becker (2006) and Boysen et al. (2007, 2008).

The NP-*hard* nature of ALBP lead most studies to define heuristic algorithms. Erel and Sarin (1998) classified these in three categories. The first category contains the single-pass procedures for which a prioritizing list for task assignment was created according to a single attribute for each task. The second category is composed of multiple single-pass procedures for which different single-pass decision rules were adopted in order to generate a set of solutions from which the best-performing one was finally selected. The third category grouped approaches improving an initial balancing, including ameliorative heuristic procedures and evolutionary techniques (*i.e.* genetic algorithms (GAs)). Both multiple-single pass and evolutionary approaches aimed at exploring a wide variety of search directions in the solution space. In studying the Single-Model Deterministic (SMD) version of the problem, Talbot *et al.* (1986) compared single-pass, multiple single-pass, and ameliorative algorithms along with optimum-seeking approaches with a set time limit to restrict the time allowed for each solution. The authors concluded that multiple single-pass and ameliorative algorithms outperformed single-pass procedures.

Multiple single-pass heuristics for tackling ALBP were mainly developed by considering the SMD version of the problem (*i.e.* Buxey 1978, Akagi *et al.* 1983). Only Arcus (1966) and Kottas and Lau (1981) addressed the Single-Model Stochastic (SMS) ALBP. Analogously, concerning GAs, recent interesting contributions are mainly focused on the deterministic ALBP (Leu *et al.* 1994, Rubinovitz and Levitin 1995, Kim *et al.* 1998, Sabuncuoglu *et al.* 2000, Rekiek *et al.* 2001 and 2002, Tseng and Tang 2006 and Tseng et al. 2007). Only Suresh *et al.* (1996) solved the stochastic version of the problem, by optimizing workload smoothness and line stopping probability.

Despite the great attention scientists have given ALBP, only a few published papers address the problem of re-balancing an existing line (Boysen et al. 2007, 2008). As a result of the volatility and unpredictability of current markets, this is now a frequent problem in industrial contexts as there are frequent changes in product features and volume demand causing tasks, task performance times, and line cycle time to be changed.

To tackle the re-balancing problem, Sculli (1979, 1984) and, recently, Van Oyen *et al.* (2001), Gel *et al.* (2002) and Montano et al. (2007) have proposed work sharing as a methodology to manage uncertainty in assembly lines. Dar-El and Rubinovitz (1991) adopted MUST algorithm for managing production line re-balancing, to accomplish with changes in performance tasks time. In particular, zoning constraints were introduced with the aim of forcing work elements to the previously assigned stations. Consequently, the re-balancing problem was tackled by imposing constraints in the re-allocation of tasks to stations. Corominas et al. (2006) analyze the re-balancing process in a company manufacturing products with seasonal demand. The authors introduce a mixed linear programming model minimizing the number of temporary workers required, for a given line cycle time and team of workers on staff. Sotskov et al. (2006) addressed line re-balancing only for those situations re-engineering ensured a larger income than expenditure. Therefore, the authors investigated the stability radius of an optimal balance, when modifications in task performance times occur.

More recently, Gamberini *et al.* (2006) pointed out that task re-assignment is an objective function to be minimized. In particular, the authors focus their attention on single-model manual assembly lines, thus considering the stochastic version of the problem, and emphasized how the re-balancing of an existing line is better modeled as a multi-objective optimization problem where both total expected completion costs and the amount of re-assignment must be minimized. Hence, the authors proposed a single-pass heuristic fulfilling these aims. In fact, when changes in product characteristics (*i.e.* precedence constraints, tasks performance time, out of line completions costs) and/or in volume demand occur, tasks must be re-assigned so as to guarantee the feasibility of the production, while trying to keep total expected completion costs down. When operators perform tasks, activity re-assignment is implemented by involving costs directly related to the amount of re-assignments, such as training, equipment switching, and quality assurance costs. Since these cost factors are difficult to estimate, Gamberini *et al.* (2006) use an index, called Similarity Factor, introduced to measure the amount of re-assignments in the new re-balanced line. Therefore, by jointly minimizing the total expected completion costs and maximizing the similarity between the new and the initial line, good trade-off solutions can be found.

A multi-objective optimization problem is not generally characterized by a unique optimal solution, that is, a solution that is better than the others for all the performance criteria considered. A set of non-dominated solutions, each representing a local optima, is frequently identified. This set is known as the Pareto front and each dominant solution belonging to it is characterized by the fact that no other better solution exists, taking all the performance criteria into consideration. Therefore, the assembly line re-balancing problem formulated as per Gamberini *et al.* (2006) should be faced by finding the largest number of non-dominated solutions (*i.e.* line balancing), of good quality and widely distributed in the solution space, best able to represent the Pareto front.

The proposed approach, by providing the analyst with the most complete set of solutions representing the Pareto front, addresses the selection of the final (unique) balancing that best interprets the opinion of the decision maker. In particular, in the final selection process practical constraints that, for reasons of complexity, the solution methodology does not directly take into consideration, can be inserted.

This paper presents a multiple single-pass heuristic algorithm developed for the purpose of solving the single-model stochastic assembly line re-balancing problem. Since such an approach combines different heuristic rules during its run, a study is carried out to find the best rule combination in order to improve the ability to find large sets of solutions, well-representing the Pareto front. Moreover, a Multi Objective Genetic Algorithm (MOGA) is also developed to solve this problem, and then compared with the multiple single-pass heuristic so that the actual capabilities of the proposed heuristic can be examined very thoroughly.

The following section of this paper reports the notation used. Section 4 summarizes the model for evaluating the task re-assignment. Sections 5 and 6 respectively introduce the multiple single-pass and the GA algorithms for the assembly line re-balancing problem. Section 7 describes the experimentation carried out and the results obtained. A case study is described in section 8. Finally, conclusions are presented in section 9.

3. Notation

μ_i mean performance time of task i, for $i = 1,\ldots,N$;

σ_i standard deviation of the performance time of task i, for $i = 1,\ldots,N$;

c_j constant positive weight, for $j = 1, 2$;

cp crossover probability;

C cycle time;

D density of the precedence matrix;

EC expected completion cost, measured in [time units per product];

$f(x)$ fitness value of the current solution;

F_i^* set of direct and indirect successors to task i, with respect to M;

h parameter for controlling the local searches;

I_i' incompletion cost of task i, for $i = 1,\ldots,N$, measured in [time units per product];

I_i total incompletion cost of task i, for $i=1,\dots,N$ given by the sum of the incompletion cost of the task i and those of its followers in the precedence matrix, measured in [time units per product];

m number of stations;

mp mutation probability;

MSF mean similarity factor of the line balancing considered;

N number of tasks;

N_E number of elite individuals to select from the elite set;

N_P number of individuals in each population;

p_i probability of not completing task i, for $i=1,\dots,N$;

s_j variable assuming values $\{-1,1\}$ according to the search direction (minimization or maximization) in the genetic algorithm;

SF_i similarity factor of task i, for $i=1,\dots,N$;

TIB_i set of tasks, other than i, assigned to the same station as task i in the initial balancing, for $i=1,\dots,N$;

TNB_i set of tasks, other than i, assigned to the same station as task i in the new balancing, for $i=1,\dots,N$;

x, y solutions coded in chromosomes;

w_j random positive weight, for $j = 1, 2$.

4. The model for evaluating tasks re-assignment

As largely introduced by Gamberini *et al.* (2006), the inputs in the single-model stochastic assembly line re-balancing problem are:

- an initial balanced line assembling a product that needs re-designing or substituting with a new item due to new customer requirements;

- data describing the tasks required to produce the new item (number of tasks, precedence diagram describing relationships between tasks, mean tasks performance time, standard deviation of tasks time, tasks incompletion cost);
- cycle time required for the new line.

The problem can be stated as follows. Given an initial line balance identifying the sets of tasks assigned to each station, the dedicated manual line can be re-balanced under the new constraints by minimizing two objectives: (i) the unit labor and the expected unit incompletion costs, and (ii) the task re-assignment. In particular, since the main difficulty in practical cases is the precise estimation of costs relating to all possible task movements, a weighted multiple objective function involving both completion and task movement related costs is not introduced. Instead, two objective functions are introduced separately: expected completion costs and the degree of similarity between initial and new task assignments. Two independent evaluations are made, leaving the interpretations of the results to analysts, in accordance with given guidelines.

The first objective function i.e. the expected assembly cost (*EC*), is evaluated by using equation (1), which is based on considerations reported in Kottas and Lau (1973, 1976, 1981):

$$EC = m \cdot C + \sum_{i=1}^{N} p_i \cdot I_i \,, \tag{1}$$

where m is the number of stations in the new line configuration, C is the cycle time, p_i is the probability of not completing task i in the assigned station, and $I_i = I'_i + \sum_{j \in F_i^*} I'_j$ is the total incompletion cost of task i (Kottas and Lau 1973), which considers the incompletion costs of both task i and those that follow it in the dependency diagram. As similar as Kottas and Lau (1973, 1976, 1981), EC, I_i and $I_i^{'}$ are measured in [time units per product]. The formulation adopted involves a small degree of approximation compared to the models presented by Kottas and Lau (1976), Sarin and Erel (1990) and Sarin *et al.* (1999). Nevertheless, for most assembly line operative conditions this approximation is negligible i.e. when the probability of an in-line completion is high.

Moreover, the term $\sum_{i=1}^{N} p_i \cdot I_i$ is easy to evaluate and so obtained quickly making it very useful in algorithms that either solve large problems or, as in the methodology presented in this paper, propose a set of potential solutions.

To evaluate the second objective function (*MSF*) i.e. the degree of similarity between the initial and the new tasks assignment, the set of tasks, other than i, assigned to the same station as task i in the initial balancing (TIB_i) and the set of tasks, other than i, assigned to the same station as task i in the new balancing (TNB_i) are introduced. Hence, the Similarity Factor (SF_i) of the generic task i is evaluated using equation (2):

$$SF_i = \frac{Cardinality\ \{\{TIB_i\} \cap \{TNB_i\}\}}{Cardinality\ \{TIB_i\}}. \tag{2}$$

SF_i is the ratio between the number of tasks assigned to the same station as task i in the initial and in the new balancing, and the number of tasks assigned to the same station in the initial balancing. When task i in the initial balancing is alone in a station, SF_i assumes an indefinite form 0/0. In this case, the SF_i value is set to 0 in assignment procedures so as to make it less likely that this task will be assigned to a station where tasks belonging to other sets are present. On the other hand, in the final *MSF* evaluation step, where the similarity of the line as a whole is computed, SF_i is set to 1 in the case where task i is alone both in the initial and in the new balancing, or otherwise 0. Finally, the Mean Similarity Factor (*MSF*) between the new re-balanced line and the initial one is evaluated using equation (3).

$$MSF = \frac{\sum_{i=1}^{N} SF_i}{N}. \tag{3}$$

An example concerning the computation of *MSF* is widely described in Gamberini *et al.* (2006). In particular, higher *MSF* values address the choice of balances using fewer task movements.

5. A multiple single-pass procedure for the single-model stochastic assembly line re-balancing problem

As described in Erel and Sarin (1998), in order to generate a set of solutions, a multiple single-pass procedure iterates the execution of a single-pass algorithm. Then the best-performing solutions are selected. In particular, the multi-objective nature of the stochastic assembly line re-balancing problem addresses the determination of the most complete set of non-dominated solutions well-representing the Pareto front.

The basic single-pass algorithm, after defining tasks belonging to the set of assignable tasks, that is, those without unassigned predecessors and whose mean performance time is not greater than the remaining station idle time, assigns tasks to station by means of a heuristic procedure probabilistically selected from those now reported (HP_1, HP_2, HP_3, and HP_4).

Heuristic procedure HP_1

The heuristic procedure introduced by Gamberini *et al.* (2006) is implemented by ranking assignable tasks in accordance with the calculation of two attributes addressing the optimization of the two above mentioned objective criteria. The first attribute evaluates how desirable an assignment is in terms of minimizing resulting assembly cost. Consequently, three task groups are defined:

- *desirable tasks* are those tasks for which anticipated labor savings in the specific position considered are greater than the expected incompletion costs. In particular, at each iteration, assigning an additional task i to the current station on the one hand results in an expected cost saving of approximately μ_i, and on the other an increase in the expected incompletion cost of approximately $p_i \cdot I_i$. Hence, desirable tasks are those which $\mu_i \geq p_i \cdot I_i$ (Kottas and Lau 1981, p. 185);

- *sure tasks*, are those desirable tasks characterized by the likelihood of a completion rate not less than 99.5%;
- *critical tasks*, are those tasks that are not desirable.

Hence $(Attribute\,I)_i$, i.e. attribute I for task i, assumes the following values:

Formatted: Indent: Hanging: 62.45 pt, Bulleted + Level: 1 + Aligned at: 53.45 pt + Tab after: 71.45 pt + Indent at: 71.45 pt, Tabs: Not at 71.45 pt

- 3, if the task is critical;
- 2, if the task is sure;
- 1, if the task is desirable;

in accordance with the need to firstly assign critical tasks to empty stations, then sure tasks, and finally those that are only desirable.

The second attribute $(Attribute\,II)_i$ assumes the values of the similarity factor SF_i (see equation (2)) computed on the basis of the set of tasks already assigned in the current station. An example concerning the use of the SF_i factor in the assignment phase is reported in Gamberini *et al.* (2006).

These two attributes are the evaluation criteria in the TOPSIS technique used to identify the best assignment. If more than one operation assumes the highest rating, a selection procedure is implemented. If they are all critical, the task with the maximum number of immediate successors is chosen. If they are all sure, the one with maximum incompletion cost is selected. If they are all desirable, the one with the minimum incompletion cost is chosen. Otherwise, the first on the list is selected.

Heuristic procedure HP_2

As with HP_1, tasks belonging to the assignable tasks set are ranked in accordance with two attributes addressing the optimization of the two aforementioned objective criteria. The first attribute evaluates the difference between expected cost saving and increment in expected incompletion cost if task i is assigned to the current station. From the cost minimization point of view, the greater the difference, the more advantageous is the assignment of tasks to a station.

$$(Attribute\, I)_i = \mu_i - p_i \cdot I_i \,. \tag{4}$$

The second attribute again $(Attribute\, II)_i$ assumes the values of the similarity factor SF_i, with operations with the highest value being selected.

Finally, TOPSIS evaluates the best assignment. In particular, in agreement with considerations used in the Kottas and Lau approaches, no operation is selected if the current station is not empty and assignable tasks are all characterized by negative values of $(Attribute\, I)_i$. Rather, it is better to create a new station in order to minimize *EC*.

Heuristic procedure HP_3

As with HP_1 and HP_2, two attributes are evaluated for each assignable task. Whilst the second attribute assumes the value of SF_i, $(Attribute\, I)_i$ is defined in agreement with considerations reported in Kottas and Lau (1981):

- tasks with consistent incompletion costs should be executed first so that the incompletion probability p_i is reduced;
- tasks with low performance time should be executed first since they involve a small increase in the incompletion probability p_i.

Hence $(Attribute\, I)_i$ is evaluated using equation (5):

$$(Attribute\, I)_i = \frac{I_i}{\mu_i} \tag{5}$$

The greater $(Attribute\, I)_i$ is, the more appropriate assigning tasks in the early part of the station is. Otherwise, the assignment is preferable in the latter part. In particular, the early and the latter part of the station are defined by the idle time (more or less than a given percentage of the cycle time).

Heuristic procedure HP_4

The task for assignment is randomly selected from the assignable tasks.

6. A genetic algorithm for the single-model stochastic assembly line re-balancing problem

Analogous to the proposed multiple single-pass approach presented in section 5, the genetic algorithm was developed so that it finds the Pareto front of the solutions representing good compromises between the two objective functions considered i.e. minimizing the unit total expected completion cost and task re-assignment. Nevertheless, since a unique objective function is necessary for GA operators to be implemented, the optimization problem is described as follows:

$$\textit{Maximize} \quad f(x), \qquad \text{with}$$

$$f(x) = s_1 \cdot c_1 \cdot w_1 \cdot EC(x) + s_2 \cdot c_2 \cdot w_2 \cdot MSF(x), \tag{6}$$

where x is the solution, s_j may assume values $\{-1, 1\}$ according to the search direction (minimization or maximization) and c_j and w_j are respectively the constant and random (positive) weights. Constant weight c_j acts as a scale factor to make it possible to compare the variations in the objectives. Random weight w_j is used to alter the relative importance of the different objectives during the iterations, thereby influencing the search direction (Murata *et al.* 1996, Ishibuchi *et al.* 1998).

Each individual in the GA population represents the sequence of the operations in the same order in which they are allocated to the stations in the line. An assignment procedure is then introduced to build the line corresponding to each individual. The concept of marginal desirability developed by Kottas and Lau (1973) is merged with a partial random assignment approach, in-depth described in Gamberini et al. (2007).

The well-known Roulette Wheel Selection, which states that the probability of a particular chromosome being chosen is related to its fitness function value, is adopted as selection procedure.

Populations are generated by genetic operators crossover and mutation. Specifically, the Two Point Crossover (TPC) and the Scramble Mutation (SM) are used as their suitability has been proven by Leu *et al.* (1994), Rubinovitz and Levitin (1995), and Sabuncuoglu *et al.* (2000).

Furthermore, SM is also adopted as local search procedure. A parameter h controls the number of local searches. The higher h is, the more accurate the neighborhood search, but the more time the algorithm takes.

The multi-objective genetic algorithm

In addition to the classical form of genetic algorithms, a set of non-dominated (elite) solutions is introduced and continuously updated to keep trace of the best individuals. Moreover, at each iteration, a predefined number of elite individuals is chosen and directly transported to the population so as to make propagation of good solutions among the generations possible.

If N_P represents the number of individuals in each population, N_E the number of elite individuals to select from the elite set, cp and mp the crossover and mutation probabilities respectively, and h the parameter to control the local searches, the MOGA procedure can be described as follows:

Step 0: Specify the constant weights c_1 and c_2, and the search direction identifiers s_1 and s_2. Randomly generate an initial population of N_P individuals.

Step 1: For each individual in the initial population, calculate the values of the *EC* and the *MSF*, and update the elite set, that is, find the non-dominated solutions. Define the initial population as the old population.

Step 2: Generate the new population. First, select N_E individuals from the elite set and add them to the new population. Second, generate the remaining $(N_P - N_E)$ individuals by repeating the following sub-procedure as much as required:

i) according to *cp*, if a crossover is required, select two individuals from the old population and generate the new one by applying the TPC, otherwise select one individual from the old population;

ii) according to *mp*, if a mutation is needed apply the SM to the new individual;

iii) randomly specify the values for the weights w_1 and w_2, and assign them to the previously generated individual.

Step 3: For each individual in the new population, apply the local search procedure as a function of parameter h and weights w_1 and w_2.

Step 4: For each individual in the new population, calculate the values of the *EC* and the *MSF*, and update the elite set.

Step 5: If a prespecified stopping condition is satisfied (i.e. number of iterations), consider the elite set as the solution set for the problem and exit. Otherwise, define the new population as the old population and return to *Step 2*.

The elite set will represent all the best individuals found in the search process, that is, those individuals that have never been dominated by others in the process as a whole.

MOGA's performance is influenced by several parameters such as N_P, N_E, the number of iterations, and the values of h, c_1, and c_2. In this study, these parameters were set to agree with the settings adopted in the literature concerning the application of genetic algorithms to ALBPs and those suggested by Ishibuchi *et al.* (1998) for the multi-objective search.

7. Experimental results

A number of computational experiments were carried out to assess the performances of the proposed heuristics. A two-fold approach was adopted. Firstly, an experimentation was used to identify the best combination of the proposed heuristic rules in the multiple single-pass procedure for different assembly problem configurations. Secondly, the optimally-tuned multiple-single pass procedure was compared with a MOGA to estimate its effectiveness in solving re-balancing problems.

[TAKE IN TABLE 1]

Regarding the nomenclature reported in Table 1, which represents all the possible changes in initial data to obtain a new product, the experimentation was structured as follows (for a more detailed description of the parameters the reader may refer to Gamberini *et al.* (2006)):

1. *Size*: 100-task, 200-task, and 400-task ALBPs are randomly generated.
2. *Performance task times* μ_i^0, for $i = 1,\ldots,N^0$, are integer values randomly generated in the range $[1,30]$.
3. *Performance task times standard deviation* σ_i^0, for $i = 1,\ldots,N^0$ are generated as:

$$\sigma_i^0 = 0.1 \cdot \mu_i^0, \qquad \text{for } i = 1,\ldots,N^0. \tag{7}$$

4. *Incompletion costs* $I_i'^0$, for $i = 1,\ldots,N^0$, are generated as:

$$I_i'^0 = a \cdot \mu_i^0, \qquad \text{for } i = 1,\ldots,N^0, \tag{8}$$

   where a is randomly generated in the interval $[1.1,2]$.
5. *Cycle time*: the cycle time is generated as:

$$C^0 = b \cdot \max\left(\mu_i^0\right), \qquad \text{for } i = 1,\ldots,N^0, \tag{9}$$

   where values of b are equal to 2 and 4.
6. *Density of the precedence matrix* (Talbot et al., 1986), D^0 is set to 0.3 and 0.8.
7. *Mutations in precedence relationships*: among the changes introduced by a re-designed or a new item, the one involving mutations in precedence relationships of the tasks is the most critical

and widespread. Hence, the precedence matrix M^0 is randomly modified (Gamberini *et al.*, 2006) to obtain a new matrix M for a percentage of mutations PM in the set $\{2,8\}$.

8. *Problems solved*: 10 line balancing problems are generated for each size N^0 in $\{100,200,400\}$, each density D^0 in $\{0.3,0.8\}$, each value of b in $\{2,4\}$, each mutations percentage in $\{2,8\}$, solving 240 problems.

For each of them, different configurations of the multiple single-pass procedure were adopted to obtain the solutions, as follows:

A: 100% HP_1 rule. This configuration degenerates in the single-pass procedure proposed by Gamberini *et al.*, (2006).

Formatted: Numbered + Level: 2 + Numbering Style: A, B, C, … + Start at: 1 + Alignment: Left + Aligned at: 36 pt + Tab after: 54 pt + Indent at: 54 pt, Tabs: Not at 54 pt

B: 100% HP_2 rule.

C: 100% HP_3 rule.

D: 50% HP_1 and 50% HP_2 rules.

E: 50% HP_2 and 50% HP_3 rules.

F: 50% HP_2 and 50% HP_4 rules.

G: 70% HP_1 and 30% HP_2 rules.

H: 30% HP_1 and 70% HP_2 rules.

I: 80% HP_1 and 20% HP_2 rules.

L: 20% HP_1 and 80% HP_2 rules.

Formatted: Numbered + Level: 1 + Numbering Style: A, B, C, … + Start at: 12 + Alignment: Left + Aligned at: 36 pt + Tab after: 54 pt + Indent at: 54 pt, Tabs: Not at 54 pt

M: 90% HP_1 and 10% HP_2 rules.

N: 10% HP_1 and 90% HP_2 rules.

In all the different configurations and for each re-balancing problem, the multiple single-pass algorithm was set to carry out 30 runs, each characterized by 31 searches. Weights for the two objectives in each run were varied in the range $[0,100]$.

To identify the best configuration for the multiple single-pass procedure, different comparisons were carried out, as reported in Tables 2, 3, 4, and 5. Since in each different configuration the output of the algorithms is a set of non-dominated solutions, the comparison is carried out by measuring, for each different pair of configurations, the number of solutions of the one which are not dominated by no one solution of the other. The parameter which measures that relative dominance is called Out-Performance Rate (Zhao *et al.,* 2001) and is defined as:

$$OPR_1 = \frac{n_1}{n_1 + n_2}, \tag{10}$$

where OPR_1 is the relative dominance of algorithm 1 over algorithm 2, n_1 is the number of solutions of algorithm 1 which are not dominated by none solutions of algorithm 2, and n_2 is the opposite. Obviously, values of OPR_1 range in the interval $[0,1]$, assuming value 0 when solutions provided by algorithm 1 are completely dominated by those obtained by algorithm 2, and value 1 when the opposite situation occurs.

The values shown in Tables 2, 3, 4, and 5 represent the relative dominance of the configuration of the column compared to that of the row. In particular, values greater than 0.5 are represented in bold italic font as these are the better performing configurations reported in the column.

[TAKE IN TABLES 2, 3, 4, 5]

Table 2 presents the comparisons of configurations A, B, and C. This comparison identifies which heuristic performs better when considered alone, that is, when the multiple single-pass procedure degenerates into a single-pass procedure. Note how configuration B consistently outperforms the other two, while configuration C is the worst performing. Table 3 shows the comparisons of configurations D, E, and F. The configurations are characterized by the use of two heuristic with the same probability of selection. In particular, the best performing heuristic identified in the previous test is coupled with all the other three. As shown, compared to the others, configuration F consistently performs the worst, stating that heuristic HP_4, which is characterized by random behavior, involves a deterioration in the ability of the algorithm to find good solutions. As a consequence, such a heuristic will not be considered in the next experiments. Moreover, the configuration D is consistently superior to configuration E, showing that best results can be obtained by combining heuristics HP_1 and HP_2.

Tables 4 and 5 show the comparison of configurations A, G, H, I, L, M, and N in the cases of PM equal to 2 and 8 respectively. The use of a multiple single-pass heuristic consistently improves the ability to find good quality solutions compared to the single-pass case, as clearly shown by the values reported in the column representing configuration A. The other comparisons illustrate the effects of different combination percentages for the two best performing heuristics, that is, HP_1 and HP_2. If low modifications are imposed (PM equal to 2), configuration M is the well-performing one. This demonstrates that there is a consistent difference between the single-pass and multiple single-pass procedure. Indeed, when the heuristic procedures were considered alone (single-pass equivalent case), HP_2 was the better one, while when the heuristic procedures were considered in pairs (multiple single-pass procedure with two heuristics) the best combination resulted in a huge use of HP_1 and low use of HP_2. If consistent modifications are imposed (PM equal to 8), configuration M maintains the best performance when the number of operations is high (400), while in small problem configurations G and I tend to become the best ones.

Figure 1 presents comparisons between configuration M of the multiple single-pass heuristic and the proposed MOGA. The MOGA was set to carry out 500 generations, with N_P set at 20, N_E set at 3, crossover (cp) and mutation (mp) probability set at 0.9 and 0.1 respectively, the local search control parameter (h) was 2, parameter L was 40, constant weight c_1 was always 1, and constant weight c_2 was 15000 for 200-task problems and 150000 for 400-task problems. The multiple single-pass algorithm, configuration M, setting was the same as in the previous experiments. In figure 1, solutions proposed by the MOGA are reported in white, while solutions obtained by the multiple single-pass algorithm are depicted in grey. It shows that the MOGA does not optimize the similarity objective well, and its solutions are always dominated by those proposed by the multiple single-pass heuristic. Similar to what is stated in Kim *et al.* (1998), these comparisons underline that hard-constrained problems are well-solved either by customized heuristics or genetic algorithms involving problem specific operators.

[TAKE IN FIGURE 1]

8. Case study

The procedure proposed in section 5 is now applied in a case study composed of re-balancing a manual line assembling a product, whose technological precedence diagram is illustrated in figure 2, due to modifications in the product design induced by market requirements. Data concerning tasks are reported in table 6. By considering a line cycle time of 605 [s/unit], the initial station workloads were evaluated using the Kottas and Lau (1973) approach and are described in table 7.

[TAKE IN FIGURE 2]

[TAKE IN TABLES 6, 7]

New customer requirements lead to re-design of the product, whose new precedence diagram and characteristic data is respectively reported in figure 3 and table 8.

[TAKE IN FIGURE 3]

[TAKE IN TABLE 8]

Alternative station workloads for the re-balanced line are reported in Figure 4. They are obtained by adopting the Kottas and Lau (1973) algorithm (this solution is named by the acronym KL73 and described in detail in table 9), the Gamberini et al. (2006) algorithm (this solution is named using the acronym GGR06) and the multiple single-pass procedure described in section 5 (these solutions are named using the acronym MSP).

[TAKE IN FIGURE 4]

[TAKE IN TABLE 9]

MSP solutions outperform KL73 and GGR06, both concerning expected costs and MSF. As described in detail in table 10, which reports the MSP solution with the highest MSF, the proposed heuristic re-balances the line by assuring that the work content of stations 4, 5 and 6 remains the same, stations 1 and 2 carry out a new operation and only station 3 is characterized by the need to carry out 2 new tasks. Furthermore, an assembly cost comparable with that featuring KL73 is proposed. On the other hand, KL73, which is only oriented to optimizing the economical performance of the line, adds up to 4 new operations in station 3 and modifies the work assigned to 5 stations. Moreover, the Pareto front solutions obtained by MSP let the decision maker choose the final line configuration from a set of interesting balancing, that can be evaluated by considering factors not directly inserted into the objective functions, such as space constraints, stations equipment, and tasks learning rate.

[TAKE IN TABLE 10]

9. Conclusions

This paper deals with assembly line re-balancing, a problem that now afflicts most companies working in competitive markets as they are forced to change product features frequently and are subject to variations in sales volumes. This implies continuous change in the balancing of their assembly lines, involving re-definition of station workloads and consequent re-assignment of tasks to operators. These task re-assignments increase some costs factors, such as quality assurance, equipment switching, and operator training, which are very difficult to estimate. Therefore, containing these costs is addressed by lowering the number of re-assignments themselves.

This paper presents a multiple single-pass heuristic algorithm in order to solve the stochastic assembly line re-balancing problem from a multi-objective standpoint, that is, it minimizes the total expected completion cost while maximizing the similarity of the task assignments between the new and the original line. The aim of the algorithm is to find the most complete set of non-dominated solutions well-representing the Pareto front of the multi-objective re-balancing problem.

Different types of heuristic procedures are used in the multiple single-pass algorithm, while the best configuration of the algorithm is obtained by means of large scale experimentation. In particular, the heuristic procedure HP_1, derived from the model developed by Gamberini *et al.* (2006), coupled with the proposed new heuristic HP_2 is best at finding good non-dominated solutions.

Moreover, a Multi-Objective Genetic Algorithm (MOGA) was developed to solve the stochastic assembly line re-balancing problem, and was compared with the optimally tuned multiple single-pass heuristic algorithm. Several experiments show that the MOGA is not able to obtain consistent results in terms of similarity optimization whatever the problem complexity, resulting in a definitive domination by the multiple single-pass heuristic.

Finally, a case study is reported which highlights benefit of using the proposed multiple single-pass algorithm to ensure good trade off between the expected completion cost and similarity with the initial line configuration. Specifically, re-balancing both minimizes modifications in station workloads and guarantees costs comparable with those proposed by cost-oriented solution procedures. Moreover, alternative good line configurations are proposed by the solution Pareto front. Consequently, the decision makers can select the final balancing by considering additive factors not directly inserted into the objective function, such as: space constraints, station equipment, and task learning rate.

Finally, in agreement with Kim *et al.* (1998), further research should address the definition of problem specific operators, improving genetic algorithms behavior in solving the re-balancing problem.

References

Akagi, F., Osaki, H., and Kikichi, S., A method for assembly line balancing with more than one worker in each station, *International Journal of Production Research*, 1983, **21**, 755-770.

Amen, M., Heuristic methods for cost-oriented assembly line balancing: A survey, *International Journal of Production Economics*, 2000, **68**, 1-14.

Amen, M., Cost-oriented assembly line balancing: Model formulations, solution difficulty, upper and lower bounds, *European Journal of Operational Research*, 2006, **168**, 747-770.

Arcus, A.L., COMSOAL: A computer method of sequencing operations for assembly lines, *International Journal of Production Research*, 1966, **4**, 259-277.

Becker, C., and Scholl, A., A survey on problems and methods in generalized assembly line balancing, *European Journal of Operational Research*, 2006, **168**, 694-715.

Boysen, N., Fliedner, M., and Scholl, A., A classification of assembly line balancing problems, *European Journal of Operational Research*, 2007, **183**, 674-693.

Boysen, N., Fliedner, M., and Scholl, A., Assembly line balancing: Which model to use when?, *International Journal of Production Economics*, 2008, **111**, 509-528.

Buxey, G., Incompletion costs versus labour efficiency on the fixed-item moving belt flowline, *International Journal of Production Research*, 1978, **4**, 233-247.

Corominas, A., Pastor, R., and Plans, J., Balancing assembly line with skilled and unskilled workers, *Omega The International Journal of Management Science*, 2006, doi: 10.1016/j.omega.2006.03.003.

Dar-El, E.M., and Rubinovitz, J., Using learning theory in assembly lines for new products, *International Journal of Production Economics*, 1991, **25**, 103-109.

Erel, E., and Sarin, S.C., A survey of the assembly line balancing procedures, *Production Planning & Control*, 1998, **9**, 414 –434.

Gamberini, R., Grassi, A., and Rimini, B., A new multi-objective heuristic algorithm for solving the stochastic assembly line re-balancing problem, *International Journal of Production Economics*, 2006, **102**, 226-243.

Gamberini, R., Grassi, A., Regattieri, A., 2007, A multiple single-pass heuristic algorithm for the stochastic assembly line re-balancing problem, *19th International Conference on Production Research*, 29th July - 4 th August, Valparaiso, Cile.

Gel, E.S., Hopp, W.J., and Van Oyen, M.P., Factors affecting opportunity of worksharing as a dynamic line balancing mechanism, *IIE Transactions*, 2002, **34**, 847-863.

Ghosh, S., and Gagnon, R.J., A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems, *International Journal of Production Research*, 1989, **27**, 637 -670.

Ishibuchi, H., and Murata, T., A multi-objective genetic local search algorithm and its application to flowshop scheduling, *IEEE Transactions on Systems, Man, and Cybernetics – Part C: applications and reviews*, 1998, **28**, 392-403.

Kim, Y.J., Kim, Y.K., and Cho, Y., A heuristic-based genetic algorithm for workload smoothing in assembly lines, *Computers and Operations Research*, 1998, **25**, 99-111.

Kottas, J.F., and Lau, H.S., A cost oriented approach to stochastic line balancing, *AIIE Transactions*, 1973, **5**, 164 -171.

Kottas, J.F., and Lau, H.S., A total operating cost model for paced lines with stochastic task times, *AIIE Transactions*, 1976, **8**, 234-240.

Kottas, J.F., and Lau, H.S., A Stochastic line balancing procedure, *International Journal of Production Research*, 1981, **19**, 177 -193.

Leu, Y.Y., Matheson, L.A., and Rees, L.P., Assembly line balancing using genetic algorithms with heuristic-generated initial populations and multiple evaluation criteria, *Decision Sciences*, 1994, **25**, 581-606.

Montano, A., Villalobos, J.R., Gutierrez, M.A., and Mar, L., Performance of serial assembly line design under unequal operator speeds and learning, *International Journal of Production Research*, **45**, 5355-5381.

Murata, T., Ishibuchi, H., and Tanaka, H., Multi-objective genetic algorithm and its applications to flowshop scheduling, *Computers and Industrial Engineering*, 1996, **30**, 957-968.

Pierreval, H., Caux C., Paris, J.L., and Viguier, F., Evolutionary approaches to the design and organization of manufacturing systems, *Computers & Industrial Engineering*, 2003, **44**, 339-364.

Rekiek, B., De Lit, P., Pellichero, F., L'Eglise, T., Fouda, P., Falkenauer, E., and Delchambre, A., A multiple objective grouping genetic algorithm for assembly line design, *Journal of Intelligent Manufacturing*, 2001, **12**, 467-485.

Rekiek, B., De Lit, P., and Delchambre, A., Hybrid assembly line design and user's preferences, *International Journal of Production Research*, 2002, **40**, 1095-1111.

Rubinovitz, J., and Levitin, G., Genetic algorithm for assembly line balancing, *International Journal of Production Economics*, 1995, **41**, 343-354.

Sabuncuoglu, I., Erel, E., and Tanyer, M., Assembly line balancing using genetic algorithms, *Journal of Intelligent Manufacturing*, 2000, **11**, 295-310.

Sarin, S.C., and Erel, E., Development of cost model for the single-model stochastic assembly line balancing problem, *International Journal of Production Research*, 1990, **7**, 1305-1316.

Sarin, S.C., Erel, E., and Dar-El, E.M., A methodology for solving single-model, stochastic assembly line balancing problem, *OMEGA - The International Journal of Management Science*, 1999, **27**, 525-535.

Scholl, A., and Becker, C., State-of-the-art exact and heuristic solution procedures for simple assembly line balancing, *European Journal of Operational Research*, 2006, **168**, 666-693.

Sculli, D., Dynamic aspects of line balancing, *OMEGA - The International Journal of Management Science*, 1979, **7**, 557 -561.

Sculli, D., Short term adjustments to production lines, *Computers & Industrial Engineering*, 1984, **8**, 53 -63.

Sotskov, Y.N., Dolgui, A., and Portmann, M.-C., Stability analysis of an optimal balance for an assembly line with fixed cycle time, *European Journal of Operational Research*, **168**, 783-797.

Suresh, G., Vinod, V.V., and Sahu, S., A genetic algorithm for assembly line balancing, *Production Planning & Control*, 1996, **7**, 38-46.

Talbot, F.B., Patterson, J.H., and Gehrlein, W.V., A comparative evaluation of heuristic line balancing techniques, *Management Science*, 1986, **32**, 430 –454.

Tseng, H.-E., and Tang, C.-E., A sequential consideration for assembly sequence planning and assembly line balancing using the connector concept, *International Journal of Production Research*, 2006, **44**, 97-116.

Tseng, H.-E., Chen, M.-H., Chang, C.-C., and Wang, W.-P., Hybrid evolutionary multi-objective algorithms for integrating assembly sequence planning and assembly line balancing, *International Journal of Production Research*, 2007, doi: 10.1080/00207540701362564.

Van Oyen, M.P., Gel, E.S., and Hopp, W.J., Performance opportunity for workforce agility in collaborative and noncollaborative work systems, *IIE Transactions*, 2001, **33**, 761-777.

Zhao, Y., Brizuela, C.A., and Sannomiya, N., Application of the partial enumeration selection method in genetic algorithms to solving a multi-objective flowshop problem, *Proceedings of the 2001 IEEE International Conference on Systems, Man, and Cybernetics*, 2001, **4**, 2365-2370.

Description	Initial data	Modified data
Number of tasks	N^0	N
Precedence matrix	M^0	M
Mean performance tasks times	μ_i^0 for $i=1,\ldots,N^0$	μ_i for $i=1,\ldots,N$
Performance tasks times standard deviations	σ_i^0 for $i=1,\ldots,N^0$	σ_i for $i=1,\ldots,N$
Tasks incompletion costs	$I_i'^0$ for $i=1,\ldots,N^0$	I_i' for $i=1,\ldots,N$
Line cycle time	C^0	C

Table 1 – Initial and modified data.

	A												B											
N^0	100				200				400				100				200				400			
D^0	0.3		0.8		0.3		0.8		0.3		0.8		0.3		0.8		0.3		0.8		0.3		0.8	
b	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4
PM	2																							
B	0.096	0.154	0.046	0.134	0.243	0.032	0.060	0.080	0.369	0.405	0.343	0.080												
C	***0.967***	***0.917***	***0.967***	***0.800***	***0.917***	***0.812***	***0.967***	***0.800***	***0.908***	***0.942***	***1.000***	***0.838***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***
PM	8																							
B	0.024	0.145	0.058	0.062	0.127	0.176	0.043	0.103	0.135	0.210	0.144	0.051												
C	***0.838***	***0.617***	***0.900***	***0.675***	***0.917***	***0.867***	***0.950***	***0.817***	***0.792***	***0.925***	***0.950***	***0.842***	***1.000***	***0.992***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***

Table 2. Results of comparison of configurations A, B, and C in terms of out-performance rate.

	D												E											
N^0	100				200				400				100				200				400			
D^0	0.3		0.8		0.3		0.8		0.3		0.8		0.3		0.8		0.3		0.8		0.3		0.8	
b	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4
PM	2																							
E	***1.000***	***0.849***	***0.969***	***0.925***	***1.000***	***0.964***	***1.000***	***0.992***	***0.990***	***0.970***	***1.000***	***1.000***												
F	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***0.992***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***
PM	8																							
E	***0.878***	***0.860***	***0.920***	***0.889***	***0.988***	***0.933***	***0.978***	***0.908***	***1.000***	***1.000***	***1.000***	***0.983***												
F	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***	***0.983***	***0.977***	***0.986***	***0.980***	***0.980***	***0.988***	***0.974***	***0.884***	***0.931***	***1.000***	***0.958***

Table 3. Results of comparison of configurations D, E, and F in terms of out-performance rate.

	A				**G**				**H**				**I**				**L**				**M**			
PM	2																							
D^0	0.3		0.8		0.3		0.8		0.3		0.8		0.3		0.8		0.3		0.8		0.3		0.8	
b	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4
N^0	100																							
G	0.000	0.018	0.022	0.000																				
H	0.031	0.051	0.062	0.050	***0.767***	***0.663***	***0.676***	***0.772***																
I	0.000	0.022	0.000	0.021	0.413	0.489	***0.641***	***0.538***	0.259	0.307	0.487	0.259												
L	0.037	0.071	0.089	0.052	***0.842***	***0.774***	***0.711***	***0.809***	***0.674***	***0.597***	***0.641***	***0.629***	***0.813***	***0.732***	***0.631***	***0.723***								
M	0.000	0.010	0.053	0.000	***0.624***	0.419	***0.740***	***0.647***	0.341	0.282	***0.580***	0.415	***0.598***	0.394	***0.705***	***0.582***	0.231	0.270	0.478	0.404				
N	0.055	0.088	0.097	0.128	***0.900***	***0.792***	***0.816***	***0.848***	***0.801***	***0.626***	***0.819***	***0.733***	***0.835***	***0.788***	***0.695***	***0.771***	***0.605***	***0.595***	***0.619***	***0.654***	***0.781***	***0.823***	***0.602***	***0.649***
N^0	200																							
G	0.114	0.029	0.031	0.000																				
H	0.278	0.115	0.031	0.018	***0.948***	***0.825***	***0.664***	***0.840***																
I	0.063	0.000	0.035	0.000	0.343	***0.506***	***0.534***	0.422	0.062	0.235	0.314	0.193												
L	0.297	0.143	0.058	0.064	***0.927***	***0.926***	***0.824***	***0.832***	***0.648***	***0.667***	***0.664***	***0.584***	***0.954***	***0.847***	***0.783***	***0.856***								
M	0.024	0.000	0.040	0.000	0.297	0.408	***0.641***	0.489	0.087	0.234	0.490	0.217	0.439	0.411	***0.683***	***0.604***	0.089	0.174	0.395	0.232				
N	0.355	0.164	0.043	0.052	***0.988***	***0.920***	***0.898***	***0.814***	***0.799***	***0.755***	***0.738***	***0.660***	***0.988***	***0.871***	***0.815***	***0.822***	***0.773***	***0.600***	***0.649***	***0.584***	***0.927***	***0.862***	***0.697***	***0.742***
N^0	400																							
G	0.054	0.078	0.010	0.026																				
H	0.185	0.287	0.062	0.103	***0.842***	***0.923***	***0.852***	***0.840***																
I	0.025	0.086	0.031	0.014	0.343	0.294	0.444	0.393	0.157	0.071	0.113	0.111												
L	0.187	0.260	0.077	0.110	***0.929***	***0.971***	***0.924***	***0.962***	***0.571***	***0.558***	***0.734***	***0.591***	***0.858***	***0.983***	***0.896***	***0.943***								
M	0.000	0.023	0.017	0.008	0.167	0.208	0.447	0.393	0.143	0.013	0.182	0.212	0.333	0.312	0.434	0.394	0.142	0.000	0.135	0.131				
N	0.208	0.366	0.117	0.216	***0.887***	***0.989***	***0.901***	***0.923***	***0.678***	***0.731***	***0.738***	***0.768***	***0.858***	***0.972***	***0.918***	***0.950***	***0.630***	***0.718***	***0.625***	***0.674***	***0.820***	***1.000***	***0.913***	***0.928***

Table 4. Results of comparison of configurations A, G, H, I, L, M, and N in terms of out-performance rate, with PM set to 2.

	A				G				H				I				L				M			
PM	8																							
D^0	0.3		0.8		0.3		0.8		0.3		0.8		0.3		0.8		0.3		0.8		0.3		0.8	
b	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4
N^0	100																							
G	0.017	0.020	0.033	0.000																				
H	0.000	0.030	0.048	0.020	0.450	***0.724***	***0.535***	0.460																
I	0.000	0.025	0.030	0.017	0.404	0.498	0.467	***0.645***	***0.599***	0.313	0.456	***0.600***												
L	0.000	0.066	0.043	0.043	0.467	***0.727***	***0.551***	0.499	***0.507***	***0.687***	***0.598***	***0.508***	0.443	***0.789***	***0.609***	0.394								
M	0.000	0.054	0.047	0.050	***0.711***	***0.658***	***0.714***	***0.779***	***0.707***	***0.501***	***0.591***	***0.700***	***0.701***	***0.679***	***0.747***	***0.680***	***0.673***	0.304	***0.570***	***0.719***				
N	0.000	0.071	0.067	0.073	***0.588***	***0.825***	***0.599***	***0.572***	***0.640***	***0.830***	***0.649***	***0.646***	0.447	***0.777***	***0.651***	***0.502***	***0.600***	***0.598***	***0.577***	***0.666***	0.437	***0.692***	0.439	0.373
N^0	200																							
G	0.011	0.039	0.000	0.000																				
H	0.017	0.099	0.000	0.028	***0.764***	***0.897***	***0.636***	***0.667***																
I	0.000	0.000	0.000	0.000	0.425	0.388	0.492	0.499	0.288	0.077	0.419	0.284												
L	0.025	0.180	0.000	0.031	***0.857***	***0.934***	***0.677***	***0.662***	***0.525***	***0.585***	0.482	0.471	***0.806***	***0.883***	***0.642***	***0.679***								
M	0.090	0.204	0.017	0.013	0.451	0.309	***0.713***	***0.594***	0.332	0.115	***0.566***	0.413	0.495	0.428	***0.725***	***0.643***	0.226	0.088	***0.504***	0.407				
N	0.052	0.182	0.010	0.015	***0.866***	***0.902***	***0.714***	***0.725***	***0.695***	***0.534***	***0.659***	***0.647***	***0.832***	***0.911***	***0.657***	***0.789***	***0.598***	0.438	***0.677***	***0.645***	***0.796***	***0.880***	0.484	***0.623***
N^0	400																							
G	0.000	0.000	0.000	0.009																				
H	0.056	0.010	0.024	0.037	***0.684***	***0.831***	***0.774***	***0.814***																
I	0.010	0.000	0.000	0.000	0.406	0.299	0.492	0.345	0.263	0.085	0.273	0.158												
L	0.138	0.042	0.031	0.073	***0.774***	***0.934***	***0.818***	***0.771***	***0.709***	***0.707***	***0.554***	***0.517***	***0.773***	***0.908***	***0.786***	***0.828***								
M	0.131	0.107	0.176	0.132	0.464	0.275	***0.601***	0.350	0.345	0.125	0.419	0.281	***0.508***	0.471	***0.640***	0.476	0.273	0.065	0.347	0.257				
N	0.145	0.077	0.080	0.102	***0.824***	***0.903***	***0.871***	***0.852***	***0.682***	***0.721***	***0.678***	***0.687***	***0.802***	***0.955***	***0.798***	***0.897***	***0.542***	***0.531***	***0.674***	***0.665***	***0.826***	***0.882***	***0.746***	***0.799***

Table 5. Results of comparison of configurations A, G, H, I, L, M, and N in terms of out-performance rate, with PM set to 8.

	Mean tasks performance time [s/unit]	Standard deviation of the tasks performance time [s/unit]	Out of line completion costs [s/unit]
1	25	0.34	31
2	42	1.11	54
3	37	0.57	47
4	80	2.15	102
5	30	0.42	40
6	300	7.42	399
7	34	0.71	44
8	180	3.86	230
9	55	1.13	69
10	220	5.74	283
11	40	1.11	48
12	64	1.06	81
13	20	0.41	26
14	15	0.40	19
15	320	9.59	386
16	44	1.14	58
17	28	0.73	34
18	70	2.03	94
19	110	1.92	143
20	25	0.25	31
21	48	1.15	60
22	36	0.51	46
23	10	0.16	12
24	5	0.12	6
25	40	0.53	50
26	16	0.25	20
27	30	0.46	37
28	284	3.95	365
29	310	7.21	382
30	105	3.02	139
31	20	0.55	27
32	32	0.43	43
33	214	4.73	263
34	82	1.71	108
35	320	4.25	411
36	10	0.12	13
37	64	1.09	80
38	18	0.26	22
39	38	0.88	49
40	90	2.12	115

Table 6: Tasks initially belonging to the precedence diagram, in the case study.

Station	Tasks	Station	Tasks
1	1	4	28
	9		19
	4		32
	2		31
	10		22
	3		34
	5		26
	7	5	35
	11		33
	14		21
	36	6	29
2	6		37
	8		25
	12		39
	13		27
	20		38
3	30		40
	15		
	18		
	16		
	17		
	23		
	24		

Table 7: Initial stations workload, in the case study.

	Mean tasks performance time [s/unit]	Standard deviation of the tasks performance time [s/unit]	Out of line completion costs [s/unit]
7	50	1.00	62
9	72	1.90	97
11	38	0.52	47
15	280	3.77	355
23	26	0.44	33
24	10	0.29	13
25	26	0.30	31
34	74	1.40	95

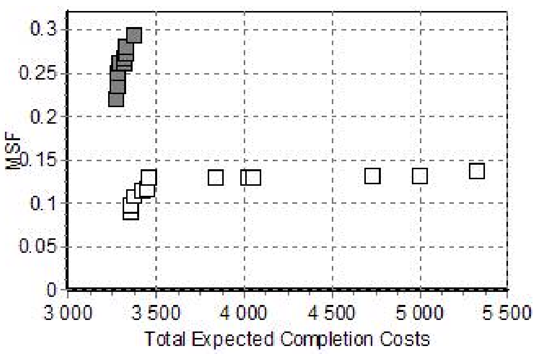
Table 8: Modified tasks data of the new item manufactured in the line, in the case study.

Station	Tasks	Station in the initial balance	Station	Tasks	Station in the initial balance
1	1	1	4	28	4
	7	1		19	4
	4	1		32	4
	2	1		22	4
	9	1		34	4
	3	1		21	5
	5	1	5	35	5
	10	1		33	5
	20	2		23	3
	24	3		26	4
2	6	2	6	29	6
	8	2		37	6
	11	1		25	6
	12	2		39	6
3	13	2		27	6
	14	1		38	6
	15	3		40	6
	30	3			
	18	3			
	16	3			
	17	3			
	31	4			
	36	1			
In line completion cost			3630 [s/unit]		
Expected out of line completion cost			9.03 [s/unit]		
Expected total completion cost			3639.03 [s/unit]		
MSF			0.55		

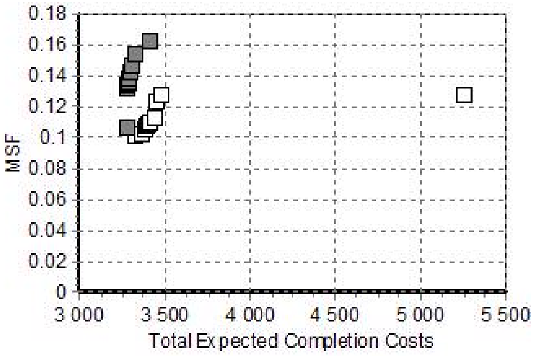
Table 9: Details of the KL73 solution, proposed for the new re-balanced line, in the case study.

Station	Tasks	Station in the initial balance	Station	Tasks	Station in the initial balance
1	1	1	4	28	4
	7	1		19	4
	4	1		32	4
	2	1		22	4
	9	1		31	4
	3	1		34	4
	5	1		26	4
	10	1	5	35	5
	36	1		33	5
	20	2		21	5
2	6	2	6	29	6
	8	2		37	6
	12	2		25	6
	11	1		39	6
3	13	2		27	6
	14	1		38	6
	15	3		40	6
	30	3			
	18	3			
	16	3			
	17	3			
	24	3			
	23	3			
In line completion cost			3630 [s/unit]		
Expected out of line completion cost			31.60 [s/unit]		
Expected total completion cost			3661.60 [s/unit]		
MSF			0.82		

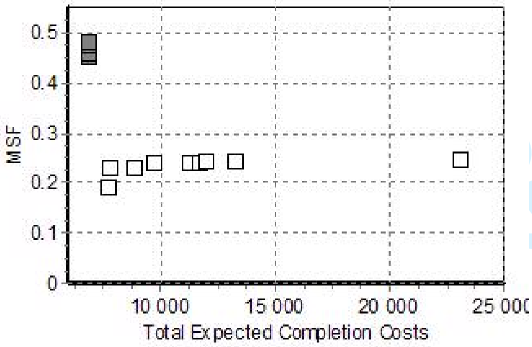
Table 10: Details of the MSP solution assuring the best value of the similarity factor, in the case study.



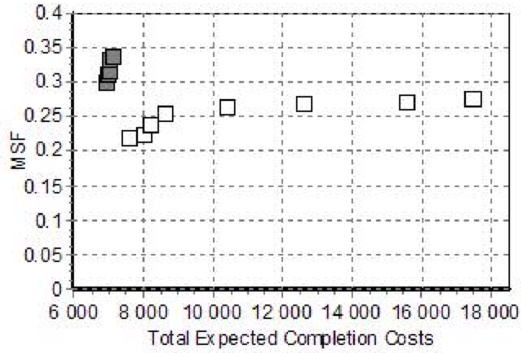
a) N^0 200, D^0 0.3, b 4, PM 2



b) N^0 200, D^0 0.3, b 4, PM 8



c) N^0 400, D^0 0.8, b 4, PM 2



d) N^0 400, D^0 0.8, b 4, PM 8

Figure 1: Comparisons between the MOGA and the multiple single-pass heuristic.

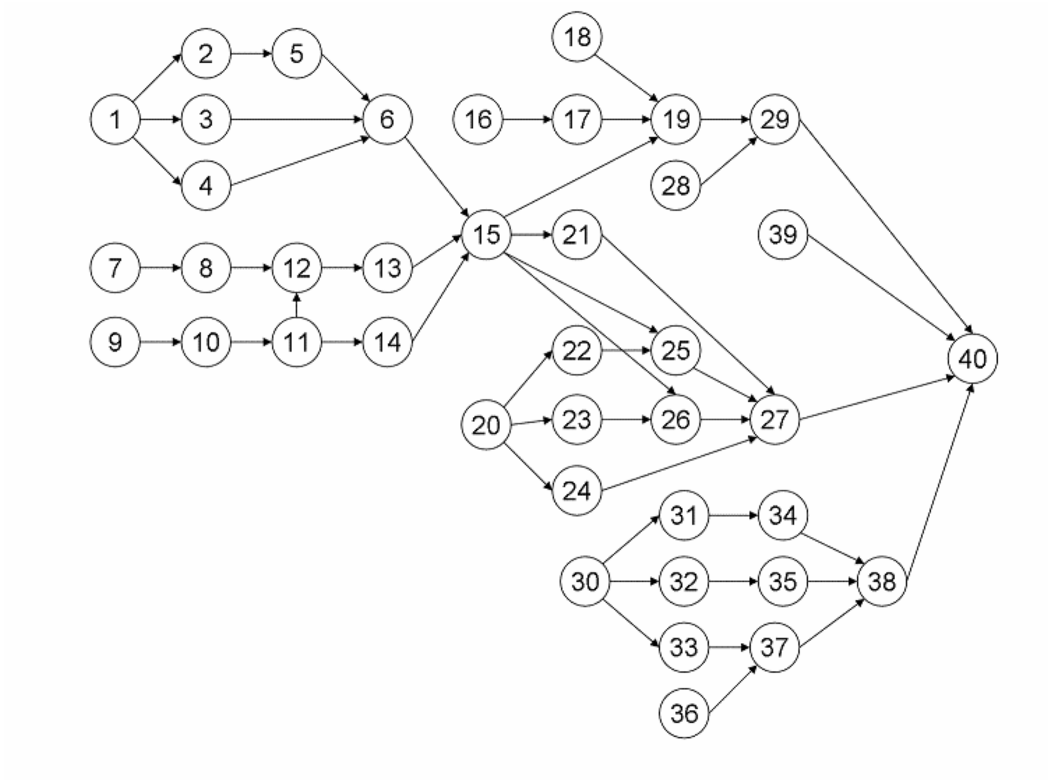


Figure 2: The precedence diagram of the item initially manufactured in the line, in the case study.

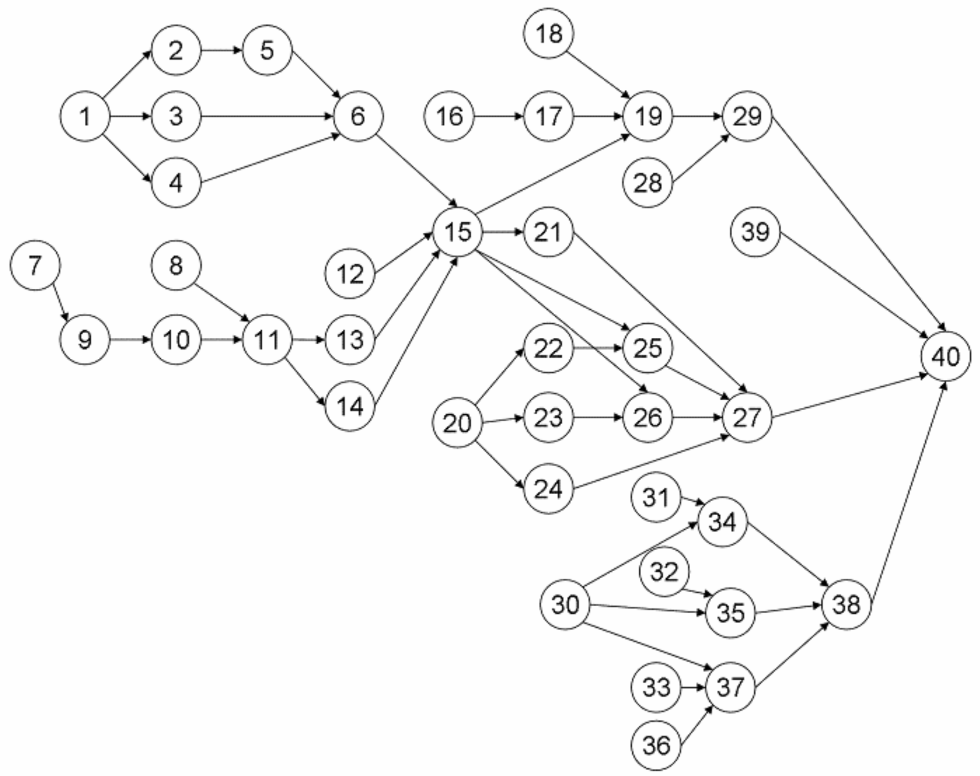


Figure 3: The precedence diagram of the new item manufactured in the line, in the case study.

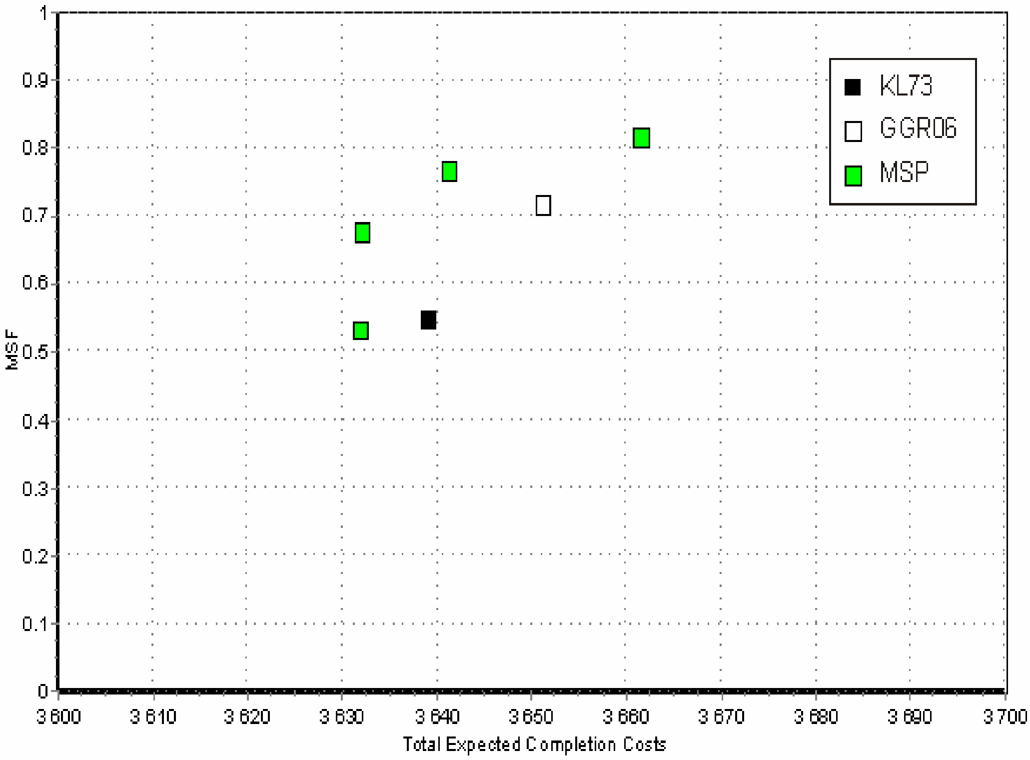


Figure 4: Alternative solutions for the new re-balanced line, in the case study.