

Chapter 2

Multi-objective Approaches for Design of Assembly Lines

X. Delorme, O. Battaïa and A. Dolgui

Abstract This chapter deals with the use of multi-objective approaches in the field of assembly line design. The design of assembly or transfer lines is a very important industrial problem, which involves various difficult and interconnected optimization problems. A review of the main multi-objective optimization methods used for these problems is presented and discussed. A case study is also described in order to highlight some interesting properties associated with such multi-objective problems.

Keywords Assembly lines · Line balancing · Multi-objective optimization · Design

2.1 Assembly Line Design

Assembly or transfer lines are production systems which are composed of several workstations organized in a serial manner. Each part successively visits each workstation by moving from one workstation to the next thanks to a linear transportation system, for example, a conveyor belt. Serial flow lines have been initially introduced for the production of large amounts of standardized products (mass-production), but are now also used for the production of families of products with low volume.

Assembly lines are intensively used in various industries (e.g., automotive or electronics) and their properties have been described in scientific literature (Nof et al. 1997).

X. Delorme (✉) · O. Battaïa · A. Dolgui
Henri Fayol Institute, CNRS UMR6158, LIMOS, Ecole des Mines de Saint-Etienne,
158 cours Fauriel 42023 Saint-Etienne Cedex 2, France
e-mail: delorme@emse.fr

2.1.1 Main Steps of Assembly Line Design

In order to design an assembly line, several important steps (see Fig. 2.1) are usually required:

1. Product(s) analysis: the aim of this step is to provide a complete description of the elementary operations to execute in order to obtain the product(s).
2. Process planning: it covers the selection of processes required to obtain the final product(s) and the definition of technological constraints. For instance, a partial order between operations (precedence constraints) is usually defined but various other restrictions have often to be considered. This step requires an accurate understanding of the functional specifications of the products as well as technological conditions for the operations.
3. Line configuration: this step defines the configuration design which implies the choice of the type of assembly line (e.g., pure serial flow line, hybrid flow shop with parallel stations or U-line), the selection of the equipment needed to perform the operations and the solution of a balancing problem, that is, the allocation of operations to workstations. It is imperative to consider all the technological constraints. At this step, a security margin often has to be considered in order to take into account failures, quality problems and also possible slight modifications of the product.
4. Line layout and transport system design: the material handling system is selected and the layout (placement of machines) is chosen. Products flow is analyzed, usually via simulation, to take into account random events and variability in production.
5. Detailed design and line implementation.

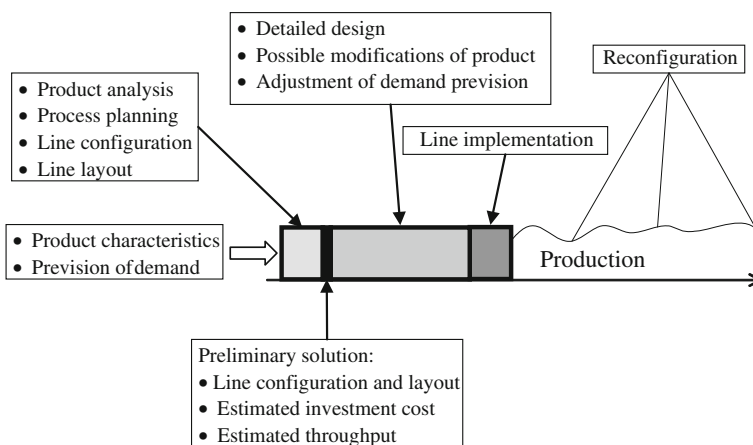


Fig. 2.1 Assembly lines design and reconfigurations

In addition, two other steps can eventually occur after the implementation of the line:

- When the line is designed for the production of several products, a scheduling problem has to be considered in order to determine the sequence of the mix of products.
- When the demand is subject to market fluctuations either in volume or characteristics of the product, the line has to undergo a reconfiguration. A reconfiguration has many similarities with the initial design but the existing line induces specific limitations and objectives.

Considering the complexity of the whole problem, these steps are usually considered sequentially. If the goal of the first two steps is to provide information on the process, the third step corresponds to a combinatorial problem whose objective is to optimize various objectives: minimizing investment costs or future labor costs, maximizing the production rate, minimizing idle times, and smoothing the workload among the workstations.

2.1.2 Line Configuration

Indeed, line configuration is of utmost importance since competitiveness and profitability depend directly on it and this problem has generated a massive amount of scientific publications. Initial studies on this problem have mostly focused on a simple version of the sole balancing problem (SALBP—simple assembly line balancing problem) with only one objective to optimize (number of workstations, production rate, or idle time) and no constraint outside of precedence. Considering the equipment selection and balancing problems independently can make sense, for example, when the equipment selection problem is trivial, either because the resources are interchangeable (e.g., operators without specific competence) or because a particular resource has to be used for each operation (e.g., tools in some lines), but it can be restrictive in many cases. Similarly, most industrial cases involve various technological constraints beside the precedence constraints.

Thus, recent studies have tried to reduce the gap between industrial applications and academic problems. Among the reasons explaining this gap, one of the most important reasons is the difficulty to effectively assess the global performance of a line configuration with a sole criterion. Some studies have considered the criterion of profit maximization to evaluate globally the solutions obtained, but such approaches seem more difficult to apply with increasingly shorter product runs and the growing uncertainty on future demand.

The rest of the chapter is organized as follows. [Section 2.2](#) presents a review of the main multi-objective optimization approaches used for the design of assembly lines such as aggregative and Pareto dominance-based methods. A case study

taken from automotive industry is considered in [Sect. 2.3](#). Conclusions and research perspectives are discussed in [Sect. 2.4](#).

2.2 Review of Literature in Multi-objective Assembly Line Optimization

The need to consider multi-objectives in research on assembly line has indeed grown progressively during the last 20 years. Obviously, considering several usually conflicting objective functions to evaluate the quality of feasible solutions instead of a sole one leads to more complex optimization problems.

In this section, we will discuss the main publications from literature on multi-objective decision-making in line balancing. The methods developed in these publications can be split into two main categories: aggregative methods and Pareto-dominance-based methods.

2.2.1 Aggregative Methods

With aggregative methods, authors actually make the assumption that the preferences of the decision-maker can be known a priori. As a consequence, they can define a relationship between the objectives which allows aggregating them. In assembly design, the most commonly used aggregation functions are:

- Lexicographic order, when no compensation is possible between objectives and a priority order can be defined among them.
- Weighted sum, when compensation is fully allowed between objectives with respect to pre-defined substitution rates (i.e., weights).
- Goal programming, when target values (i.e., goals) can be defined for each criterion and the objective is to minimize the deviation from these targets.

2.2.1.1 Lexicographic Order

The first papers using a lexicographic order to aggregate several objective functions for assembly line problems have focused on mixed-model line balancing problems or flexible manufacturing systems. (Gökçen and Erel 1997) studied a problem with three objectives to optimize: the number of workstations, the cycle time, and an objective related to a soft constraint on incompatible zoning. They proposed a preemptive goal programming model inspired from (Deckro and Rangachari 1990) (see [Sect. 2.2.1.3](#) for more details) but their methods actually acts as a lexicographic approach.

In (Sawik 1998), the author considered a line balancing problem combined with parts routing in a flexible assembly system. A two-phase heuristic was proposed to minimize lexicographically the cycle time and the total interstation transfer time.

Pastor et al. (2002) worked on an industrial application corresponding to a multiproduct assembly line balancing problem with four objectives considered in lexicographic order: overall production rate, workload smoothness, cycle time disparity between the different product types, and uniformity of tasks in each workstation. The problem was tackled with two tabu search algorithms which were applied sequentially.

Özcan and Toklu (2009a, b) studied a line balancing problem for the mixed-model two-sided assembly lines. Two performance objectives were considered simultaneously: maximizing the weighted line efficiency and minimizing the weighted smoothness index. They proposed tabu search and simulated annealing algorithms, respectively.

There are also publications on single-model assembly line balancing problems (ALBP). Baykasoğlu (2006) studied two versions of an assembly line balancing problem: straight and U-type line. A multi-rule simulated annealing algorithm was proposed in order to optimize a smoothness index as primary objective and the number of workstation as secondary objective.

The balancing problem associated with U-lines was also investigated by (Gökçen and Ağpak 2006). As in (Gökçen and Erel 1997), the proposed method was inspired from the goal programming model of (Deckro and Rangachari 1990) but was tuned to act as a lexicographic approach. The primary objective to minimize was the number of workstations, with the cycle time as secondary objective and a tertiary objective corresponding to the minimization of the maximal number of tasks assigned per workstation.

Özcan and Toklu (2010) considered two-sided assembly lines with sequence-dependent setup times between tasks. Performing a task directly before another task may influence the latter task inside the same station, because a setup for performing the latter task may be required. Furthermore, if a task is assigned to a station as the last one, then it may cause a setup for performing the first task assigned to that station since the tasks are performed cyclically. A mixed integer program (MIP) was proposed to model and solve the problem. The proposed MIP minimizes the number of mated-stations (i.e., the line length) as the primary objective and it minimizes the number of stations (i.e., the number of operators) as a secondary objective for a given cycle time. A heuristic approach (2-COMSOAL/S) for especially solving large-size problems based on COMSOAL (computer method of sequencing operations for assembly lines) was also presented.

Fattahi et al. (2011) extended the case of two-sided lines to multimanned lines where more than two workers can work at each workstation. A MIP was proposed to solve the balancing problem of the multimanned assembly lines optimally. This model minimizes the total number of workers on the line as the first objective and the number of opened multimanned workstations as the second one. A heuristic based on the ant colony optimization approach was developed to solve the medium- and large-size scales of this problem.

Pastor (2011) introduced a new type of assembly line balancing problem which consists to not only minimize the workload of the most heavily loaded workstation, but then the workload of the second most heavily loaded workstations, then the third, and so on. By nature, such a problem is obviously multi-objective.

Lastly, lexicographic order has also been considered in (Gupta and McGovern 2004) for a disassembly line balancing problem. An ant colony algorithm was proposed which primary sought to minimize the number of workstations, and then to optimize the workload smoothness and the position of hazardous parts in the sequence.

2.2.1.2 Weighted Sum

Contrary to the lexicographic order, the weighted sum was initially used on single-model ALBP. One of the first use of a weighted sum for assembly line balancing was presented by (Leu et al. 1994) as an extension of their work on the use of genetic algorithms to tackle line balancing problems with various objectives. They illustrated their idea with two objectives, namely, workload smoothness and idle time, and suggested to use different weights to obtain several different trade-offs.

In (Ponnambalam et al. 2000), a genetic algorithm was presented for a SALBP. The proposed algorithm sought to optimize a weighted sum of the number of workstations, the line efficiency and a smoothness index. Similarly, Suwannarongsri and Puangdownreong (2009) proposed a tabu search algorithm to optimize a weighted of the same objectives plus the idle time.

Hamta et al. (2011) formulated a flexible task time assembly line balancing problem. Task processing time could be between lower and upper bounds associated with each type of machine available. The machines could compress the processing time of tasks, but this action lead to higher cost. This cost was described in terms of task time via a linear function. A bi-objective nonlinear integer programming model was developed which comprises two inconsistent objective functions: minimizing the cycle time and minimizing the machine total costs. The LP-metric was used to combine these objectives. A genetic algorithm was developed to solve this problem.

Zacharia and Nearchou (2012) presented a fuzzy extension of the SALBP of type 2 with fuzzy task processing times formulated by triangular fuzzy membership functions. The total fuzzy cost function was formulated as the weighted-sum of two bi-objectives fuzzy objectives: (a) minimizing the fuzzy cycle time and the fuzzy smoothness index of the workload of the line; (b) minimizing the fuzzy cycle time of the line and the fuzzy balance delay time of the workstations. A multi-objective genetic algorithm was applied to solve the problem.

Purnomo et al. (2013) considered single model two-sided assembly line problem. The aim of the model was minimizing the cycle time for a given number of

mated-workstations and balancing the workstation simultaneously. Genetic algorithm and iterative first-fit rule were used to solve the problem. Based on the experiments, the iterative first-fit rule could take the advantage of finding the best position over many workstations and the genetic algorithm provided more flexible task assignment and was significantly faster than the iterative first-fit rule.

By comparison with single-model problems, the use of weighted sum on mixed-model assembly line problems is rather recent. The only exception corresponds to the work of (Sawik 1997) which considered a weighted sum for his combined balancing and routing problem previously to his work on lexicographic order (Sawik 1998). The main interest of the proposed approach was the interactive procedure with the decision-maker in order to set the value of the weights.

Kara et al. (2007) considered this approach for a combined balancing and sequencing problem in a mixed-model U-line. One objective (workload smoothness) was related to the balancing problem, but the two others (setup cost, smoothness of parts' usage rate) corresponded to the sequencing problem. A simulated annealing algorithm was used in this study. The same combined problem was also studied by (Hwang and Katayama 2010) but with different objectives: two associated with the balancing problem (line efficiency, workload smoothness) and one with the sequencing problem (difference between the actual and average workload).

Hwang and Katayama (2009), proposed an evolutionary approach to deal with workload balancing problems in mixed-model U-shaped lines. The performance objectives considered are the number of workstations and the variation of workload, simultaneously.

Simaria et al. (2009) worked on the same problem of mixed-model two-sided assembly lines than (Özcan and Toklu 2009a) and proposed an ant colony optimisation algorithm, which optimizes a weighted sum of line efficiency and smoothness index.

Kara et al. (2011) studied mixed-model assembly lines with the duplication of common tasks for several models. Three goals relevant to MALB-CD were considered in two pre-emptive goal programming models, one with precise and the other with fuzzy goals, namely minimizing the number of workstations, the cycle time and the total cost required to duplicate common tasks.

Beside these works on deterministic line balancing, (McMullen and Frazier 1998) has proposed a simulated annealing using a weighted sum to deal with a stochastic assembly line balancing problem with parallel stations. The objectives considered in this study were the total labor and equipment cost, workload smoothness, and a probability of lateness.

2.2.1.3 Goal Programming

Goal programming has been one of the first aggregation methods used on multi-objective assembly line problems. Indeed (Deckro and Rangachari 1990) studied a single-model assembly line balancing and proposed an integer linear program with

goals to reach for the number of workstations, the cycle time and some soft constraints. Alongside this work, (Malakooti 1991) suggested a goal programming approach for another line balancing problem with the same first two objectives and a third objective corresponding to operating costs instead of the soft constraints.

However, goal programming has actually generated interest in literature by comparison with lexicographic order or weighted sum, and these publications have merely focused on single-model ALBP. More recently, some works have considered a fuzzy extension of goal programming. (Toklu and Özcan 2008) presented a fuzzy goal programming model for the single-model U-line balancing problem with multiple objectives. The first fuzzy goal was the number of workstations in the U-line. The second fuzzy goal was the cycle time. The third fuzzy goal was the maximal number of tasks which were assigned to each workstation in the U-line. A similar approach was used by (Cheshmehgaz et al. 2012) for an assembly line balancing problem with specific objectives related to the ergonomics of the line (posture diversity and accumulated risk posture).

2.2.1.4 Other Aggregative Functions

More anecdotally, some other aggregation methods have been considered for line balancing problems. (Duta et al. 2003) studied a disassembly line balancing problem with two objectives which were aggregated using a ratio function: the outcomes resulting from the valorization of components were divided by the cycle time.

A more sophisticated approach was used in (Gamberini et al. 2006) for a stochastic assembly line reconfiguration problem. In this study, the two objectives considered (labor and incompleteness costs, task reassignment) were aggregated using a rank function based on TOPSIS which calculated a distance measure to the ideal and nadir values. Also using a distance measure, (Hamta et al. 2013) worked on an assembly line balancing problem with setup times and operational times varying according to a learning curve. They used a particle swarm optimization algorithm hybridized with a Variable Neighborhood Search to optimize a combination of cycle time, equipment cost and a smoothness index. The main peculiarity of this study was to aggregate these three objectives by using a weighted sum of the distances between the solution considered and a lower bound on each objective.

Finally, some other works could, to some extent, also be considered as multi-objective. For example, the works on SALBP-E deal with two objectives, cycle time and number of workstations, and use the multiplication operator to aggregate both objectives. Another important example of such an implicit aggregation of several objectives comes from profit oriented methods.

2.2.2 Pareto Dominance-based Methods

With Pareto dominance-based methods, authors suppose that the preferences of the decision-maker are unknown. As a consequence, they try to provide a list of interesting trade-offs between the objectives rather than a lone solution. These trade-offs are usually defined by using the Pareto dominance.

Malakooti (1991) was one of the first to study the Pareto front of some single-model ALBP as he tried to deduce some useful properties for a resolution. Following this work, Malakooti and Kumar (1996) proposed a multi-objectives decision support system for ALBP. In their study, they considered five objectives (number of workstations, cycle time, total cost of operations, production rate, and buffer size) but the article actually focused more on interactions with the decision-maker rather than on the optimization problem.

After these works, researches on Pareto-dominance based methods for single-model ALBP have mostly focused on algorithmic methods seeking to approximate the Pareto front. Indeed, in a study of several genetic operators for ALBP with various objectives, Kim et al. (1996) suggested an extension to multi-objective genetic algorithms (MOGA).

Nearchou (2008) considered two versions of the single-model ALBP with two bi-objective: (1) minimizing the cycle time of the assembly line and the balance delay time of the workstations; (2) minimizing the cycle time and the smoothness index of the workload of the line. A new population heuristic was proposed to solve the problem based on the general differential evolution method. The cost function was represented by a weighted-sum of multiple objectives functions with self-adapted weights. The efficiency of the algorithm MODE was compared to a weighted sum Pareto genetic algorithm (GA), and a Pareto-niched GA. The experimental comparisons showed a promising high quality performance for MODE approach. For the second version of the problem, a MODE approach with a new acceptance scheme based on the Pareto dominance concept and a new evaluation scheme based on TOPSIS was proposed by (Nourmohammadi and Zandieh 2011) and a particle swarm optimization algorithm was developed by (Nearchou 2011).

Hwang et al. (2008) presented a MOGA to solve the single-model U-shaped ALBP. The objectives considered were the number of workstations (the line efficiency) and the variation of workload. Chutima and Olanviwatchai (2010) extended the formulation of Hwang and Katayama (2009) by adding a third objective of minimum work relatedness and proposed an evolutionary method with coincidence algorithm. The same problem but with different objectives (cycle time, variation of workload and total operators cost) was studied by (Zhang and Gen 2011) who proposed a generalized Pareto-based scale-independent fitness function genetic algorithm (gp-siffGA) to solve it.

Chica et al. (2010) considered time and space ALBP with the joint minimization of the number and the area of the stations given a fixed cycle time limit and proposed a random greedy search algorithm. Other algorithms were also developed

for the same problem: NSGA-II (Chica et al. 2011a) and ant colony optimisation (Chica et al. 2011b). Finally, Chica et al. (2012) developed memetic versions of these algorithms with a multi-objective local search procedure. The memetic advanced NSGA-II showed its excellent performance, obtaining the best solutions.

Similarly, Rekiek et al. (2001) worked on a GA for a line balancing problem with equipment selection. The fitness evaluation of the proposed algorithm was based on the multi-objective decision analysis method Promethee II to order the solutions of the population.

In (Bukchin and Masin 2004), the authors studied a multi-objective ALBP with equipment selection for team oriented assembly systems. The objectives considered were the number of teams, the flowtime (which corresponded to inventory costs) and two objectives related to team oriented assembly systems. They proposed a branch-and-bound algorithm to generate the efficient set as well as some heuristics.

Chen and Ho (2005) considered a specific case of ALBP with equipment selection but without precedence constraints. A MOGA were proposed to obtain potentially efficient solutions for four objectives: total flow time, workload smoothness, cycle time, and tools cost.

Pekin and Azizoglu (2008) addressed the assembly line configuration problem of assigning tasks and equipment to workstations where several equipment alternatives are possible for each task. Minimizing the total equipment cost and the number of work stations were considered together. A branch-and-bound algorithm with powerful reduction and bounding mechanisms was developed.

A multi-objective evolutionary algorithm was also presented in (Shin et al. 2011) with three objectives under consideration: workload smoothness, part movements and tools changes. The approximation of Pareto front obtained by this approach was compared with the solutions from two classic MOGA (NSGA-II and SPEA 2).

Yoosefelahi et al. (2012) considered a robotic ALBP with the following objectives: to minimize the cycle time, robot setup costs and robot costs. Three versions of multi-objective evolution strategies were developed to solve this problem.

Recently, Yang et al. (2013) addressed the reconfiguration problem for a mixed-model assembly line with seasonal demands. The problem was to reassign assembly tasks and operators to candidate stations under the constraint of a given cycle time. The objectives were to minimize the number of stations, workload variation at each station for different models, and rebalancing cost. A MOGA was proposed to solve this problem. A non-dominated ranking method was used to evaluate the fitness of each chromosome. A local search procedure was developed to enhance the search ability of the proposed MOGA.

Chutima and Chimklai (2012) extended the problem of (Özcan and Toklu 2009a) by considering three objectives: (1) to minimize the number of mated-stations, (2) to minimizing the number of workstations or operators, and (3) the tertiary objective consisted of two conflicting sub-objectives to be optimized simultaneously, that is, to maximize work relatedness and minimize workload

smoothness. They developed a particle swarm optimization (PSO) algorithm with negative knowledge (PSONK) to solve this problem. In addition, a local search scheme (2-Opt) was embedded into PSONK (called M-PSONK) in order to improved Pareto frontiers obtained.

Another field that has generated some publications corresponds to stochastic ALBP. McMullen and Tarasewich (2006) studied this problem considering four objectives: total cost, probability of lateness, number of workers, and usage rate. They proposed a multi-objective method based on the ACO principles.

Gamberini et al. (2009) considered a stochastic assembly line reconfiguration problem with two joint objectives, total expected completion cost of the new line and similarity between the new and the existing line. A multiple single-pass heuristic algorithm was developed for the purpose of finding the most complete set of nondominated solutions representing the Pareto front of the problem. A multi-objective genetic algorithm was also developed but showed worse results than the heuristic algorithm.

Cakir et al. (2011) dealt with multi-objective optimization of a single-model stochastic ALBP with parallel stations. The objectives were as follows: (1) minimization of the smoothness index and (2) minimization of the design cost. To obtain Pareto-optimal solutions for the problem, an algorithm, based on simulated annealing (SA) was developed. It implemented a multinomial probability mass function approach, tabu list, repair algorithms and a diversification strategy.

Finally, Ding et al. (2010) extended the work of (Gupta and McGovern 2004) on disassembly line balancing by proposing a multi-objective ACO which used an evaluation based on the Pareto-dominance and a niching method.

2.3 Case Study of a Reconfigurable Transfer Line

2.3.1 Problem Description

In this section, we will study a multi-objective problem associated with the design of a reconfigurable machining line (RML). As introduced by (Koren et al. 1999), reconfigurable manufacturing systems are designed to allow easy changes in their physical configuration to answer market fluctuations in both volume and type of product. To achieve this goal, the main required characteristics are: modularity, integrability, customization, convertibility and diagnosability. The use of a RML is motivated by the increasingly shorter product runs and the need for more customization (see Fig. 2.2).

The line under consideration is paced and serial. As for classic assembly lines, the configuration of machining lines corresponds to the assignment of a set of processing operations to workstations which are equipped with a set of machines tools. The usual constraints (processing time of each operation, precedence) must be considered, but machining lines imply several other specific constraints:

- Some subsets of operations must be executed on the same workstation (inclusion constraints);
- Some subsets of operations cannot be executed on the same workstation (exclusion constraints).

Moreover, each workstation is composed of several identical computer numerical controller (CNC) machine-tools to facilitate a future reconfiguration of the line. Within a workstation, each CNC machine executes the same operations (in parallel on different units of products). A part is held at a machine with some fixtures in a given position (part fixing and clamping), but it is possible to rotate the part. However, even after the part rotation or displacement, some sides and elements of the part are not accessible for machining, and the operations which must be processed on these hidden or covered areas cannot be executed (see Fig. 2.3). Therefore, the choice of a part position for part fixing should be also considered in the optimization procedure because it generates specific restrictions for the assignment of operations to workstations.

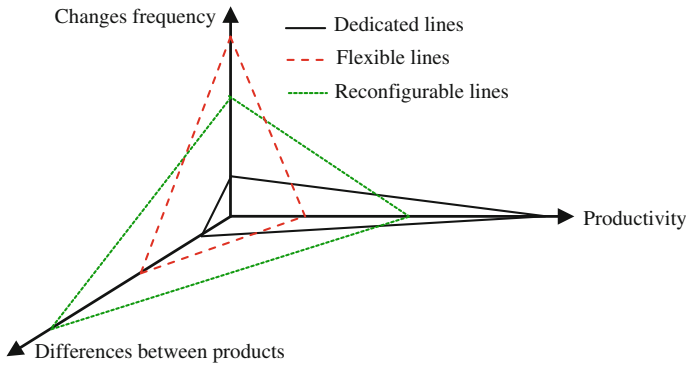
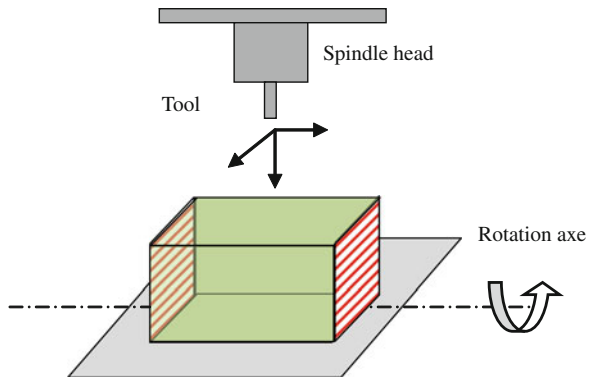


Fig. 2.2 Main differences between dedicated, flexible, and reconfigurable lines

Fig. 2.3 Processing restrictions due to part position



In such a line, all the operations assigned to a station are performed sequentially by the same spindle, and thus sequence-dependent set-up times must be considered. Set-up times are related to the rotation of the part but also change and displacement of the tool. Finally, the sequential execution of operations, as well as the setup times, usually implies large workload times and, as a consequence, parallel machines can be required at some stations in order to increase the production rate. The interest of these lines as well as the main steps of their design has been described in (Delorme et al. 2009).

Now let's introduce the notations used for the various data associated with this problem:

N	set of operations to be assigned;
P_i	set of all predecessors (direct and indirect) of operation i . Precedence constraints define a partial relation of order between operations;
ES	set of subsets $e \in N$ of operations which must be assigned to the same workstation. ES represents the inclusion constraints; that is, the need to carrying out fixed groups of operations on the same workstation;
\overline{ES}	set of pairs of operations (i, j) which cannot be assigned to the same workstation (exclusion constraints); that is, the impossibility of carrying out certain subsets of operations on the same workstation. All pairs (i, j) in \overline{ES} are defined such that $i < j$
A	set of possible part positions for part fixing in a machining center;
A_i	subset of part positions which allow to process operation i ($A_i \subseteq A$)
t_i	operational time to process operation i ;
$t_{i,j}$	setup time needed when operation j is processed directly after operation i . The time required for the execution of two sequential operations (i, j) is thus equal to $t_i + t_{i,j} + t_j$;
TMAX	maximal cycle time considered for the line;
CoS	cost associated with the opening of one workstation;
CoM	cost of one CNC machine;
CoMAX	maximal possible investment cost for the line;
$M = \{1, \dots, MaMAX\}$	set of possible numbers of machines on a workstation. When several identical CNC machines are installed on the same workstation, the local cycle time of the workstation is equal to the number of parallel machines multiplied by the line cycle time (takt time);
OpMAX	maximal number of operations which can be assigned to a workstation;
StMAX	maximal number of workstations on the line

A solution of this problem, that is, a feasible line configuration, is composed of several decisions:

- The number of workstations and the assignment of each operation to a workstation (balancing problem);
- The sequence of the assigned operations for each workstation (scheduling problem);
- The part fixing position and the number of CNC machines for each workstation (equipment problem).

Moreover, the choice of a solution among the different feasible configurations is actually a multi-objective problem since enterprises seek to minimize investment cost, which depends on the number of workstations and the number of CNC machines, and to maximize the throughput, which is equivalent to minimize the cycle time, at the same time.

The resulting optimization problem is NP-Hard since the sole balancing or scheduling problems are already NP-Hard even with only one objective. A mixed integer programming (MIP) formulation has been proposed for single objective version of this problem corresponding to the minimization of the investment cost with an upper bound on the cycle time of the line (Essafi et al. 2010), but only small size instances could be solved (less than 20 operations). As a consequence, several heuristics have been proposed to deal with this problem (Essafi et al. 2012; Borisovsky et al. 2012).

2.3.2 Some Interesting Properties for a Multi-objective Optimization

Despite the difficulty of this problem, it presents some very interesting properties which can be used during the optimization process. In this chapter, we will focus on three main properties and discuss how they can be used to tackle the multi-objective version of this problem.

2.3.2.1 Given the Sequence of Operations

Let's consider a given sequence of the operations of set N . In this case, the remaining decisions to be made are:

- The points in the sequence of operations where there is a change of workstation, which corresponds to a balancing problem with a strict total order on operations;
- The part fixing position and the number of CNC machines for each workstation.

These remaining decisions correspond to a multi-objective optimization problem which can be formulated with the following MIP:

$$\text{Minimize } T \quad (2.1)$$

$$\text{Minimize } \text{CoS} \times \sum_{i \in N, k \in M} x_{i,k} + \text{CoM} \times \sum_{i \in N, k \in M} k \times x_{i,k} \quad (2.2)$$

s.t.

$$\sum_{i \in N, k \in M} x_{i,k} \leq \text{StMAX} \quad (2.3)$$

$$\sum_{k \in M} x_{i,k} \leq 1, \forall i \in N | i < |N| \quad (2.4)$$

$$\sum_{k \in M} x_{n,k} = 1 \quad (2.5)$$

$$\sum_{j \in [i, i + \text{OpMAX} - 1], k \in M} x_{j,k} \geq 1, \forall i \in N | i \leq |N| - \text{OpMAX} + 1 \quad (2.6)$$

$$\sum_{l \in [i, j], k \in M} x_{l,k} \geq 1, \forall (i, j) \in \overline{ES} \quad (2.7)$$

$$\sum_{j \in [\min_{i \in e} \{i\}, \max_{i \in e} \{i\}], k \in M} x_{j,k} = 0, \forall e \in ES \quad (2.8)$$

$$\sum_{j \in [i, \max\{1 \geq i | \cap_{\lambda \in [i, \lambda]} A_{\lambda} \neq \emptyset\}], k \in M} x_{j,k} \geq 1, \forall i \in N \quad (2.9)$$

$$\tau_i \leq k \times T + \text{MaMAX} \times \text{TMAX} \times \left(1 - \sum_{k' \in M, k' \leq k} x_{i,k'}\right), \forall i \in N, k \in M \quad (2.10)$$

$$\tau_i \geq t_i, \forall i \in N \quad (2.11)$$

$$\tau_i \geq \tau_{i-1} + t_i + t_{i-1,i} - \sum_{j \in N, |i - \text{OpMAX}| \leq j \leq i} t_j \times \sum_{k \in M} x_{i-1,k}, \forall i \in N | i > 1 \quad (2.12)$$

$$x_{i,k} \in \{0, 1\}, \forall i \in N, k \in M \quad (2.13)$$

$$T \geq 0 \quad (2.14)$$

In this model (2.1–2.14), the decision variables $x_{i,k}$ are equal to 1 if operation i is the last operation assigned to a workstation equipped with k CNC machines, τ_i is the workload time accumulated on the current workstation up to operation i , and T correspond to the cycle time of the line. Note that a solution of this model does not contain a decision on the part fixing positions but the constraint (1.9) ensures that there exists at least one possible part position for each workstation so the remaining decision is trivial.

This MIP can be used to obtain new solutions by setting various upper bounds on one of the objective and optimizing the second objective. This method

corresponds to an ε -constraint approach. By this way, we can search for solutions in specific areas of the objectives space.

Another way to deal with this multi-objective problem is to use an aggregative function of the objectives. Various aggregative functions can be used for this purpose, for example, one commonly used aggregation function in line balancing is related to the notion of efficiency as in SALBP of type E. Here, maximizing the efficiency of the line corresponds to the minimization of the multiplication of the cost by the cycle time, which is equivalent to the minimization of the cost per unit of product. The corresponding optimization problem can be formulated with the following MIP:

$$\text{Minimize } \sum_{i \in N, k \in M} (\text{CoS} \times \sigma_i + \text{CoM} \times \mu_i) \quad (2.15)$$

s.t.

$$\sigma_i \geq T - \text{TMAX} \times \left(1 - \sum_{k \in M} x_{i,k}\right), \forall i \in N \quad (2.16)$$

$$\mu_i \geq k \times \left(T - \text{TMAX} \times \left(1 - \sum_{k' \in M, |k'| \geq k} x_{i,k'}\right)\right), \forall i \in N, k \in M \quad (2.17)$$

$$\text{CoS} \times \sum_{i \in N, k \in M} x_{i,k} + \text{CoM} \times \sum_{i \in N, k \in M} k \times x_{i,k} \leq \text{CoMAX} \quad (2.18)$$

$$T \leq \text{TMAX} \quad (2.19)$$

$$\sum_{i \in N, k \in M} x_{i,k} \leq \text{StMAX} \quad (2.20)$$

$$\sum_{k \in M} x_{i,k} \leq 1, \forall i \in N | i < n \quad (2.21)$$

$$\sum_{k \in M} x_{n,k} = 1 \quad (2.22)$$

$$\sum_{j \in [i, i + \text{OpMAX} - 1], k \in M} x_{j,k} \geq 1, \forall i \in N | i \leq n - \text{OpMAX} + 1 \quad (2.23)$$

$$\sum_{l \in [i, j], k \in M} x_{l,k} \geq 1, \forall (i, j) \in \overline{\text{ES}} \quad (2.24)$$

$$\sum_{j \in [\min_{i \in e} \{i\}, \max_{i \in e} \{i\}], k \in M} x_{j,k} = 0, \forall e \in \text{ES} \quad (2.25)$$

$$\sum_{j \in [i, \max\{1 \geq i | \cap_{\lambda \in [i, j]} A_\lambda \neq \emptyset\}], k \in M} x_{j,k} \geq 1, \forall i \in N \quad (2.26)$$

$$\tau_i \leq k \times T + \text{MaMAX} \times \text{TMAX} \times \left(1 - \sum_{k' \in M, |k'| \leq k} x_{i,k'}\right), \forall i \in N, k \in M \quad (2.27)$$

$$\tau_i \geq t_i, \forall i \in N \quad (2.28)$$

$$\tau_i \geq \tau_{i-1} + t_i + t_{i-1,i} - \sum_{j \in N, |i-\text{OpMAX}| \leq j \leq i} t_j \times \sum_{k \in M} x_{i-1,k}, \forall i \in N | i > 1 \quad (2.29)$$

$$x_{i,k} \in \{0, 1\}, \forall i \in N, k \in M \quad (2.30)$$

$$\sigma_i, \mu_i \geq 0, \forall i \in N \quad (2.31)$$

$$T \geq 0 \quad (2.32)$$

In the second model (2.15–2.32), two additional decision variables are used: σ_i is equal to the cycle time of the line if operation i is the last operation assigned to a workstation, and μ_i is equal to the cycle time of the line multiplied by the number of CNC machines which equip the current workstation if operation i is the last operation assigned to a workstation. The remainder of the model is similar to the first model (2.1–2.14), save for constraints (2.18) and (2.19) which bound the cost of the line and its cycle time, respectively.

2.3.2.2 Given the Assignment of Operations to Workstations

Let's now consider a given assignment of all operations to workstations. In this case, the remaining decisions to be made are:

- The sequence of operations for each workstation;
- The number of CNC machines for each workstation.

Note that the choice of the part fixing position is trivial as soon as the assignment of operations to workstations is set. Moreover, both decisions to be made can be considered sequentially: whatever number of CNC machines used in a workstation, a nonoptimal sequence of operations can only imply more setup time and thus cannot lead to a solution with a lower cost or cycle time.

The first decision corresponds to a single machine scheduling problem with sequence-dependent setup time and precedence constraints which has to be solved for each workstation. As shown by (Bigras et al. 2008), this problem is equivalent to a time-dependent traveling salesman problem for which various MIP formulations and algorithms have been proposed.

When the workload of each workstation is known, the second decision can be easily obtained with the following procedure:

1. Generate a first solution S_1 by assigning one CNC machine for each workstation;
2. Calculate the cost and the cycle time of S_1 . Note that the cycle time is determined by the workstation which has the larger workload; let's denote this workstation as w_1

3. Set index i to 1
4. If there are MaMAX CNC machines on workstation w_i go to step 9
5. Generate solution S_{i+1} by adding one CNC machine on workstation w_i
6. Calculate the cost and the cycle time of S_{i+1} . Determine the workstation which has the larger local cycle time; let's denote this workstation as w_{i+1}
7. Increment index i of 1
8. Go to step 4
9. Set index p to i
10. End

This procedure allows obtaining a set $S = \{S_1, \dots, S_p\}$ of solutions which are not dominated by each other. Moreover, any other solution with the same assignment of operations to workstations would necessary be dominated by at least one of the solutions of S .

2.3.2.3 Duplicating and Combining of Workstations

Finally, let's consider a solution corresponding to a feasible RML denoted X . The cost and the cycle time of this solution are denoted $C(X)$ and $T(X)$, respectively. Let's also suppose that we would like to find a solution with a lower cycle time than X . In this case, we can actually decide to consider a production system $X^{(2)}$ composed of two identical production lines X working in parallel. The cost of this production system would have be twice the cost of X but its cycle time would be half those of X . Such reasoning can be generalized with any number of parallel production line as soon as the cost of the corresponding production system don't exceed the upper bound:

$$\begin{cases} C(X^{(l)}) = l \times C(X) \\ T(X^{(l)}) = \frac{T(X)}{l} \end{cases}, \forall l \in \{2, \dots, \text{LiMAX}\} | l \times C(X) \leq \text{CoMAX}$$

where LiMAX corresponds to the maximum number of parallel lines.

As a consequence, duplication permits to generate $\frac{\text{CoMAX}}{C(X)} - 1$ new solutions from any solution X . Note that all the solutions generated by this way will have the same efficiency.

Similarly, we can decide to design a production system composed of two different production lines, X and Y , working in parallel. Let's denote the production system resulting from the combination of X and Y as $\langle X + Y \rangle$. The cost and cycle time of this production system can be calculated as follow:

$$\begin{cases} C(\langle X + Y \rangle) = C(X) + C(Y) \\ T(\langle X + Y \rangle) = \frac{T(X) \times T(Y)}{T(X) + T(Y)} \end{cases}$$

We can easily demonstrate that the cycle time of the combined solution is always lower than those of each initial line, which means that the solution generated by combination neither dominates nor is dominated by any of the initial solutions.

2.3.3 Illustration on a Didactic Example

We will now illustrate some of these properties on a didactic example. All the numerical data of the considered case are indicated in Tables 2.1 and 2.2.

Let's consider the sequence of operations used to present the operations in Table 2.1, that is, $\{1, 2, 3, \dots, 25, 26\}$. Note that this sequence is feasible since it respects all precedence constraints and there is no incompatible operation between any two operations in inclusions.

Using this sequence, we can use the model (2.15–2.32) to obtain a feasible solution of maximal efficiency. Using the solver IBM ILOG CPLEX 12.4 on a computer Intel® Core™ with 2.20 Ghz CPU and 8 Go of RAM, the optimal solution of this problem (for the given sequence) is obtained in less than 3 s. This solution is reported in Table 2.3. It corresponds to a single line with seven workstations and 15 CNC machines.

Knowing this solution, we can use the procedure described in Sect. 2.3.2.2 as a local search. The first step of this procedure is to determine the optimal sequence of operations for each workstation. Considering the very small size of the seven corresponding sequencing problems, their solution can be obtained nearly immediately. The resulting solution is reported in Table 2.4. This solution actually weakly dominates the initial solution in the sense of Pareto since its cycle time is lower.

We can now apply the second step of the procedure in order to obtain a set S of solutions. All the solutions generated are presented in Table 2.5. The last three columns in this table indicate which workstations are equipped with 1, 2, and 3 CNC machines, respectively. For each solution, the workstation which has the largest cycle time, and thus has no idle time, is indicated in bold. Note that the first two solutions (S_1 and S_2) are unfeasible since they don't respect the maximal cycle time (these values are in italic), so we have generated seven non-dominated solutions.

As indicated in Sect. 2.3.2.3, we can obtain additional solutions by duplicating the solutions of the set S . A total of 18 new solutions can indeed be generated (see Table 2.6); however, one of these solutions ($S_9^{(3)}$) is unfeasible since its cost exceeds the maximal value (this value is in italic).

Finally, we can also combine some of these solutions. For example, combining the lines S_3 and S_9 would result in a production system with an overall cycle time

Table 2.1 Numerical data of operations for the case study

Operation	Processing time	Direct predecessors	Incompatible operations	Operations on the same station	Possible part positions
1	10	–	–	3	1, 2
2	20	1	–	–	1, 2
3	15	–	–	1	1, 2
4	12	2	6	–	1
5	10	–	–	–	1, 2
6	15	4	4	–	1, 2
7	8	–	–	–	1, 3
8	16	–	–	10,13	2, 3
9	5	–	–	–	2, 3
10	7	–	–	8,13	2, 3
11	10	2	–	–	1, 2
12	4	–	–	–	1, 2
13	8	–	–	8,10	1, 2, 3
14	12	11	–	–	1, 3
15	10	2	–	–	1, 2
16	5	15	–	–	1, 2
17	7	–	18	–	2, 3
18	3	16	17	21	2, 3
19	4	–	–	–	1, 2, 3
20	6	–	–	–	1, 3
21	10	6, 14	22	18	2, 3
22	8	18, 21	21	–	1, 3
23	3	–	–	–	1, 3
24	4	22	–	–	1, 2, 3
25	3	–	–	–	1, 3
26	7	24	–	–	1, 2
Costs	Workstation		CoS = 50,000		
	CNC machine		CoM = 200,000		
Bounds	Cycle time		TMAX = 40		
	Investment cost		CoMAX = 10,000,000		
	Machines per workstation		MaMAX = 3		
	Operations per workstation		OpMAX = 10		
	Workstations per line		StMAX = 10		
	Parallel lines		LiMAX = 3		

of 13.93 and a total cost of 5,500,000. Similarly, we can also combine the solutions duplicated with a solution of set S (but not two solutions duplicated together because the maximal number of lines in parallel would be exceeded).

Obviously, many solutions generated by these different procedures are dominated but we can extract 32 non-dominated solutions. Figure 2.4 presents the corresponding Pareto front as well as the initial solution obtained by the model

Table 2.2 Setup times between operations of the case study

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	–	2	3	3	4	3	3	3	4	2	3	3	3	4	4	3	4	3	3	3	3	2	4	4	3	4
2	3	–	4	3	4	4	3	3	5	3	3	4	3	4	5	3	3	3	4	5	4	5	4	3	3	4
3	2	4	–	5	5	3	4	5	4	2	2	3	4	4	4	5	4	2	4	4	3	4	3	4	4	3
4	4	4	5	–	2	5	4	2	4	4	3	4	3	4	4	3	3	3	4	3	3	3	4	2	3	3
5	3	3	3	4	–	3	3	3	4	4	3	3	5	3	3	4	3	4	5	3	3	3	3	4	5	4
6	4	5	3	3	3	–	3	4	5	4	2	2	3	4	4	4	5	4	2	4	4	3	4	3	4	5
7	4	4	4	5	4	4	–	4	5	5	3	4	5	4	2	2	3	4	4	4	5	4	2	4	4	3
8	2	3	3	3	4	4	3	–	2	3	4	3	3	4	2	2	3	4	4	4	5	4	2	4	4	3
9	5	3	3	4	3	4	5	5	–	5	4	4	4	3	5	3	3	4	3	4	5	3	3	3	4	5
10	3	4	3	4	4	3	3	4	3	–	3	4	3	3	4	3	4	4	3	4	2	4	4	3	4	3
11	4	3	4	4	3	3	5	4	3	3	–	4	5	3	4	4	4	5	4	2	4	4	3	4	3	4
12	3	3	3	4	3	3	3	3	4	4	3	–	3	4	2	4	4	3	5	4	2	3	5	3	5	4
13	3	4	5	4	5	4	3	5	4	5	5	4	–	2	4	5	3	4	3	3	3	5	3	3	2	5
14	3	4	5	3	3	3	4	5	4	5	4	3	3	–	4	2	5	4	4	3	4	4	5	5	4	4
15	4	3	4	5	3	3	3	4	5	4	5	4	3	5	–	5	4	3	4	5	4	3	2	4	3	3
16	4	3	4	3	4	3	4	3	3	4	3	3	3	4	4	–	4	5	2	3	4	2	2	3	3	3
17	4	2	4	5	2	2	2	3	4	3	2	5	5	3	5	5	–	2	2	3	4	4	5	2	4	4
18	4	5	2	5	5	5	3	2	2	2	2	5	2	2	4	5	2	–	4	5	3	4	2	2	3	3
19	5	4	2	2	5	2	2	4	3	4	5	5	3	3	2	5	5	4	–	4	5	3	4	5	3	3
20	3	4	2	2	2	3	5	2	4	4	4	4	2	2	2	2	5	5	5	–	3	2	3	2	3	3
21	5	2	2	5	2	3	5	4	4	3	2	4	5	3	2	2	2	5	4	4	–	2	5	4	2	2
22	2	5	2	3	3	4	2	2	2	2	4	5	4	4	5	4	4	4	2	3	5	–	3	4	4	4
23	5	5	5	5	3	2	3	2	2	5	4	4	4	5	5	4	4	4	2	3	2	4	–	4	5	5
24	4	4	2	5	2	4	3	2	4	5	5	3	4	3	4	2	3	5	3	4	5	4	3	–	3	4
25	3	4	2	5	5	5	4	4	4	5	2	4	3	2	3	2	2	4	2	2	3	5	3	4	–	3
26	2	2	4	2	2	4	4	3	4	2	2	2	3	2	2	2	4	4	2	4	2	4	5	5	5	–

Table 2.3 Solution obtained with the model (2.15–2.32)

Station	Sequence of operations	Workload	Number of CNC machines
1	1, 2, 3, 4	68	3
2	5, 6, 7	39	2
3	8, 9, 10, 11, 12, 13	67	3
4	14	12	1
5	15, 16, 17	31	2
6	18, 19, 20, 21	34	2
7	22, 23, 24, 25, 26	38	2
Cycle time: 22.67			
Cost: 3,350,000			

(2.15–2.32). The procedure which has permitted to generate each solution is also indicated and a curve represents the best efficiency value obtained. Logically, the duplication and combination produce solutions with lower cycle time and larger cost, but all the solutions obtained seem to adequately cover the whole Pareto front.

Table 2.4 Solution obtained with optimal sequence in each workstation

Station	Sequence of operations	Workload	Number of CNC machines
1	3, 1, 2, 4	64	3
2	5, 6, 7	39	2
3	8, 9, 12, 11, 10, 13	65	3
4	14	12	1
5	15, 16, 17	31	2
6	18, 19, 20, 21	34	2
7	22, 23, 24, 25, 26	38	2
Cycle time: 21.67			
Cost: 3,350,000			

Table 2.5 Set S of solutions obtained with the procedure of Sect. 2.3.2.2

	Cycle time	Cost	Stations with one CNC machine	Stations with two CNC machines	Stations with three CNC machines
S_1	65	1,750,000	1, 2, 3 , 4, 5, 6, 7	–	–
S_2	64	1,950,000	1 , 2, 4, 5, 6, 7	3	–
S_3	39	2,150,000	2 , 4, 5, 6, 7	1, 3	–
S_4	38	2,350,000	4, 5, 6, 7	1, 2, 3	–
S_5	34	2,550,000	4, 5, 6	1, 2, 3, 7	–
S_6	32.5	2,750,000	4, 5	1, 2, 3 , 6, 7	–
S_7	32	2,950,000	4, 5	1 , 2, 6, 7	3
S_8	31	3,150,000	4, 5	2, 6, 7	1, 3
S_9	21.67	3,350,000	4	2, 5, 6, 7	1, 3

Table 2.6 Solutions obtained by duplication

	Cycle time	Cost		Cycle time	Cost
$S_1^{(2)}$	32.5	3,500,000	$S_1^{(3)}$	21.67	5,250,000
$S_2^{(2)}$	32	3,900,000	$S_2^{(3)}$	31.33	5,850,000
$S_3^{(2)}$	19.5	4,300,000	$S_3^{(3)}$	13	6,450,000
$S_4^{(2)}$	19	4,700,000	$S_4^{(3)}$	12.67	7,050,000
$S_5^{(2)}$	17	5,100,000	$S_5^{(3)}$	11.33	7,650,000
$S_6^{(2)}$	16.25	5,500,000	$S_6^{(3)}$	10.83	8,250,000
$S_7^{(2)}$	16	5,900,000	$S_7^{(3)}$	10.67	8,850,000
$S_8^{(2)}$	15.5	6,300,000	$S_8^{(3)}$	10.33	9,450,000
$S_9^{(2)}$	10.83	6,700,000	$S_9^{(3)}$	7.22	10,050,000

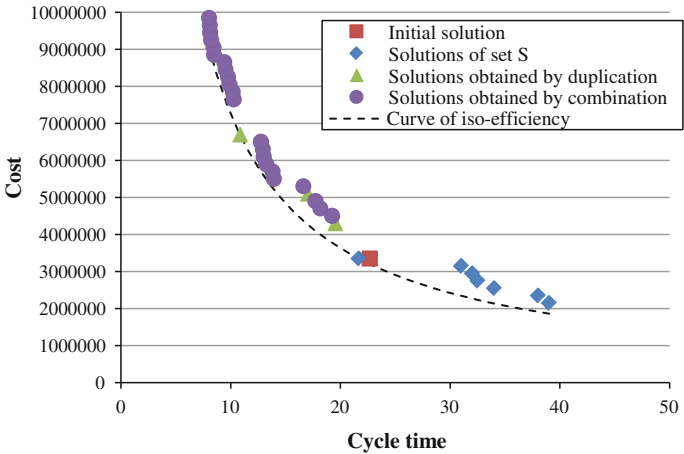


Fig. 2.4 Pareto front generated from the initial sequence

2.4 Conclusion and Perspectives

In this chapter, we have explained the interest to consider multiple criteria in order to support decision-making for the design of assembly lines, and we have presented a review of the main publications in this field.

As highlighted in this review, multi-objective optimization methods for assembly lines have been for a long time really scarce. Authors have mainly focused their research on different versions of the SALBP. The most known multi-objective version of this problem (with cycle time and number of workstations to minimize simultaneously) can be easily tackled with an ϵ -constraint method. However, this lack of interest is no longer true with the problems studied which involve the most known multi-objective complex constraints and objectives. As a result, publications on multi-objective line balancing problems have been booming in recent years (70 % of the publications referenced in this chapter have been published since 2006), and the number of research articles on this matter should continue to grow. A case study illustrated this trend as there is still a need to develop multi-objective optimization methods that can fully take advantage of the different properties we have described.

Finally, beside the development of efficient optimization methods, the integration of the corresponding models and algorithms within multi-objective decision support systems has not yet generated many contributions, but it is clearly a crucial question for years to come in order to allow a practical use in industry.

References

- Baykasoğlu A (2006) Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems. *J Intell Manuf* 17(2):217–232
- Bigras L, Gamache M, Savard G (2008) The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times. *Dis Opt* 5(4):685–699
- Borisovsky P, Delorme X, Dolgui A (2012) Genetic algorithm for balancing reconfigurable machining lines. *Comput Ind Eng*. doi:[10.1016/j.cie.2012.12.009](https://doi.org/10.1016/j.cie.2012.12.009)
- Bukchin J, Masin M (2004) Multi-objective design of team oriented assembly systems. *Eur J Oper Res* 156(2):326–352
- Cakir B, Altıparmak F, Dengiz B (2011) Multi-objective optimization of a stochastic assembly line balancing: a hybrid simulated annealing algorithm. *Comput Ind Eng* 60(3):376–384
- Chen JH, Ho SY (2005) A novel approach to production planning of flexible manufacturing systems using an efficient multi-objective genetic algorithm. *Int J Mach Tools Manuf* 45(7–8):949–957
- Cheshmehgaz H, Haron H, Kazemipour F, Desa M (2012) Accumulated risk of body postures in assembly line balancing problem and modeling through a multi-criteria fuzzy-genetic algorithm. *Comput Ind Eng* 63(2):503–512
- Chica M, Cordon O, Damas S (2011a) An advanced multi-objective genetic algorithm design for the time and space assembly line balancing problem. *Comput Ind Eng* 61(1):103–117
- Chica M, Cordon O, Damas S, Bautista J (2011b) Including different kinds of preferences in a multi-objective ant algorithm for time and space assembly line balancing on different Nissan scenarios. *Exp Syst Appl* 38(1):709–720
- Chica M, Cordon O, Damas S, Bautista J (2012) Multi-objective memetic algorithms for time and space assembly line balancing. *Eng Appl Artif Intell* 25(2):254–273
- Chica M, Cordon O, Damas S, Bautista J, Pereira J (2010) Multi-objective constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search. *Inf Sci* 180(18):3465–3487
- Chutima P, Chimklai P (2012) Multi-objective two-sided mixed-model assembly line balancing using particle swarm optimisation with negative knowledge. *Comput Ind Eng* 62(1):39–55
- Chutima P, Olanvithachai P (2010) Mixed-model U-shaped assembly line balancing problems with coincidence memetic algorithm. *J Softw Eng Appl* 3(4):347–363
- Deckro R, Rangachari S (1990) A goal approach to assembly line balancing. *Comput Oper Res* 17(5):509–521
- Delorme X, Dolgui A, Essafi M, Linxe L, Poyard D (2009) Machining lines automation. In: Nof DS (ed) *Handbook of automation*. Springer, New York, pp 599–617
- Ding LP, Feng YX, Tan JR, Gao YC (2010) A new multi-objective ant colony algorithm for solving the disassembly line balancing problem. *Int J Adv Manuf Technol* 48(5–8):761–771
- Duta L, Filip F, Henrioud JM (2003) A method for dealing with multi-objective optimization problem of disassembly processes. In: *Proceedings of the 5th IEEE international symposium on assembly and task planning*, Besançon, pp 163–168
- Essafi M, Delorme X, Dolgui A (2012) A reactive GRASP and path relinking for balancing reconfigurable transfer lines. *Int J Prod Res* 50(18):5213–5238
- Essafi M, Delorme X, Dolgui A, Guschinskaya O (2010) A MIP approach for balancing transfer line with complex industrial constraints. *Comput Ind Eng* 58(3):393–400
- Fattahi P, Roshani A, Roshani A (2011) A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem. *Int J Adv Manuf Technol* 53(1–4):363–378
- Gamberini R, Grassi A, Rimini B (2006) A new multi-objective heuristic algorithm for solving the stochastic assembly line re-balancing problem. *Int J Prod Econ* 102(2):226–243
- Gamberini R, Grassi E, Regattieri A (2009) A multiple single-pass heuristic algorithm solving the stochastic assembly line rebalancing problem. *Int J Prod Res* 47(8):2141–2164

- Gökçen H, Ağpak K (2006) A goal programming approach to simple U-line balancing problem. *Eur J Oper Res* 171(2):577–585
- Gökçen H, Erel E (1997) A goal programming approach to mixed-model assembly line balancing problem. *Int J Prod Econ* 48(2):177–185
- Gupta S, McGovern S (2004) Multi-objective optimization in disassembly sequencing problems. In: Second world conference on POM and 15th annual POM conference, Cancun
- Hamta N, Fatemi Ghomi S, Jolai F, Akbarpour Shirazi M (2013) A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect. *Int J Prod Econ* 141(1):99–111
- Hamta N, Fatemi Ghomi S, Jolai F, Bahalke U (2011) Bi-criteria assembly line balancing by considering flexible operation times. *Appl Math Model* 35(12):5592–5608
- Hwang R, Katayama H (2009) A multi-decision genetic approach for workload balancing of mixed-model U-shaped assembly line systems. *Int J Prod Res* 47(14):3797–3822
- Hwang R, Katayama H (2010) Integrated procedure of balancing and sequencing for mixed-model assembly lines: a multi-objective evolutionary approach. *Int J Prod Res* 48(21):6417–6441
- Hwang R, Katayama H, Gen M (2008) U-shaped assembly line balancing problem with genetic algorithm. *Int J Prod Res* 46(16):4637–4650
- Kara Y, Ozcan U, Peker A (2007) Balancing and sequencing mixed-model just-in-time U-lines with multiple objectives. *Appl Math Comp* 184(2):566–588
- Kara Y, Özgüven C, Seçme N, Chang C (2011) Multi-objective approaches to balance mixed-model assembly lines for model mixes having precedence conflicts and duplicable common tasks. *Int J Adv Manuf Techn* 52(5–8):725–737
- Kim Y, Kim Y, Kim Y (1996) Genetic algorithms for assembly line balancing with various objectives. *Comput Ind Eng* 30(3):397–409
- Koren Y, Heisel U, Jovane F, Moriwaki T, Pritschow G, Ulsoy G et al (1999) Reconfigurable manufacturing systems. *CIRP Ann Manuf Technol* 48(2):527–540
- Leu YY, Matheson L, Rees L (1994) Assembly line balancing using genetic algorithms with heuristic-generated initial populations and multiple evaluation criteria. *Dec Sci* 25(4):581–606
- Malakooti B (1991) A multiple criteria decision making approach for the assembly line balancing problem. *Int J Prod Res* 29(10):1979–2001
- Malakooti B, Kumar A (1996) A knowledge-based system for solving multi-objective assembly line balancing problems. *Int J Prod Res* 34(9):2533–2552
- McMullen P, Frazier G (1998) Using simulated annealing to solve a multi-objective assembly line balancing problem with parallel workstations. *Int J Prod Res* 36(10):2717–2741
- McMullen P, Tarasewich P (2006) Multi-objective assembly line balancing via a modified ant colony optimization technique. *Int J Prod Res* 44(1):27–42
- Nearchou AC (2008) Multi-objective balancing of assembly lines by population heuristics. *Int J Prod Res* 46(8):2275–2298
- Nearchou A (2011) Maximizing production rate and workload smoothing in assembly lines using particle swarm optimization. *Int J Prod Econ* 129(2):242–250
- Nof S, Wilhem W, Warnecke H (1997) *Industrial assembly*. Chapman Hall, London
- Nourmohammadi A, Zandieh M (2011) Assembly line balancing by a new multi-objective differential evolution algorithm based on TOPSIS. *Int J Prod Res* 49(10):2833–2855
- Özcan U, Toklu B (2009a) A tabu search algorithm for two-sided assembly line balancing. *Int J Adv Manuf Techn* 43(7):822–829
- Özcan U, Toklu B (2009b) Balancing of mixed-model two-sided assembly lines. *Comput Ind Eng* 57(1):217–227
- Özcan U, Toklu B (2010) Balancing two-sided assembly lines with sequence-dependent setup times. *Int J Prod Res* 48(18):5363–5383
- Pastor R (2011) LB-ALBP: the lexicographic bottleneck assembly line balancing problem. *Int J Prod Res* 49(8):2424–2442

- Pastor R, Andrés C, Duran A, Pérez M (2002) Tabu search algorithms for an industrial multi-product and multi-objective assembly line balancing problem, with reduction of the task dispersion. *J Oper Res Soc* 53(12):1317–1323
- Pekin N, Azizoglu M (2008) Bi criteria flexible assembly line design problem with equipment decisions. *Int J Prod Res* 46(22):6323–6343
- Ponnambalam S, Aravindan P, Mogileeswar Naidu G (2000) A multi-objective genetic algorithm for solving assembly line balancing problem. *Int J Adv Manuf Technol* 16:341–352
- Purnomo H, Wee H, Rau H (2013) Two-sided assembly lines balancing with assignment restrictions. *Math Comp Model* 57(1–2):189–199
- Rekiek B, De Lit P, Pellichero F, L'Eglise T, Fouda P, Falkenauer E et al (2001) A multiple objective grouping genetic algorithm for assembly line design. *J Intell Manuf* 12(5–6):467–485
- Sawik T (1997) An interactive approach to bicriterion loading of a flexible assembly system. *Math Comp Model* 25(6):71–83
- Sawik T (1998) A lexicographic approach to bi-objective loading of a flexible assembly system. *Eur J Oper Res* 107(3):656–668
- Shin K, Park JO, Kim Y (2011) Multi-objective FMS process planning with various flexibilities using a symbiotic evolutionary algorithm. *Comput Oper Res* 38:702–712
- Simaria A, Zanella de Sá M, Vilarinho P (2009) Meeting demand variation using flexible U-shaped assembly lines. *Int J Prod Res* 47(14):3937–3955
- Suwannarongsri S, Puangdownreong D (2009) Metaheuristic approach to assembly line balancing. *WSEAS transactions on systems* 2(8):200–209
- Toklu B, Özcan U (2008) A fuzzy goal programming model for the simple U-line balancing problem with multiple objectives. *Eng Optim* 40(3):191–204
- Yang C, Gao J, Sun L (2013) A multi-objective genetic algorithm for mixed-model assembly rebalancing. *Comput Ind Eng* 65(1):109–116
- Yoosefelahi A, Aminnayeri M, Mosadegh H, Davari Ardakahi H (2012) Type II robotic assembly line balancing problem: an evolution strategies algorithm for a multi-objective mode. *J Manuf Syst* 31(2):139–151
- Zacharia P, Nearchou A (2012) Multi-objective fuzzy assembly line balancing using genetic algorithms. *J Intell Manuf* 23(3):615–627
- Zhang W, Gen M (2011) An efficient multi-objective genetic algorithm for mixed-model assembly line balancing problem considering demand ratio-based cycle time. *J Intell Manuf* 22(3):367–378

Applications of Multi-Criteria and Game Theory
Approaches

Manufacturing and Logistics

Benyoucef, L.; Hennet, J.-C.; Tiwari, M.K. (Eds.)

2014, XVI, 408 p. 80 illus. in color., Hardcover

ISBN: 978-1-4471-5294-1