



**HAL**  
open science

## Simulated annealing for a bi-objective-based scalable balancing of a reconfigurable assembly line under uncertainty and limited resources

Siwar Arbi, Audrey Cerqueus, Evgeny Gurevsky, Alexandre Dolgui, Ali Siadat

### ► To cite this version:

Siwar Arbi, Audrey Cerqueus, Evgeny Gurevsky, Alexandre Dolgui, Ali Siadat. Simulated annealing for a bi-objective-based scalable balancing of a reconfigurable assembly line under uncertainty and limited resources. 50th International Conference on Computers & Industrial Engineering (CIE50), Oct 2023, Sharjah, United Arab Emirates. hal-04199608

**HAL Id: hal-04199608**

**<https://hal.science/hal-04199608v1>**

Submitted on 2 Dec 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Simulated annealing for a bi-objective-based scalable balancing of a reconfigurable assembly line under uncertainty and limited resources

Siwar Arbi<sup>a</sup>, Audrey Cerqueus<sup>a</sup>, Evgeny Gurevsky<sup>b</sup>, Alexandre Dolgui<sup>a</sup>, Ali Siadat<sup>c</sup>

<sup>a</sup>IMT Atlantique, LS2N, Nantes, France

<sup>b</sup>LS2N, University of Nantes, France

<sup>c</sup>LCFC, ENSAM, Metz, France

## Abstract

In this paper, we address the design of a reconfigurable assembly line while considering the presence of tasks with uncertain processing time and a limited number of extra resources. The principal goal is to find a line design that offers the most interesting scope in terms of productivity and robustness when the extra resources are used. In order to evaluate any line design in such a situation, a new bi-objective based scalability indicator is introduced. A simulated annealing algorithm is developed to find a line design that optimizes the proposed indicator. The algorithm is tested and approved on a set of instances derived from the literature. The results show that the proposed approach is stable and capable of finding a suitable design solution in a relatively short computational time.

*Keywords:* reconfigurable manufacturing systems, simulated annealing, scalability, robustness, productivity, balancing, bi-objective optimization

## 1. Introduction

The concept of reconfigurable manufacturing systems (RMS) was introduced by Koren et al. [1] as a response to the high volatility and unpredictability of market demand driven by global competition. Traditional manufacturing systems, including both dedicated and flexible ones, are not well-suited to meet the requirements of the current market (see, *e.g.*, Koren et al. [2]). These latter systems are unable to provide a rapid and cost-effective response to changes in production needs, which is an important capability for companies seeking to survive in a turbulent business environment. RMS have a number of key features (modularity, integrability, customization, convertibility, diagnosability and scalability) that enable them to quickly and effectively adjust to meet changing market demands (see, *e.g.*, Koren and Shpitalni [3]). The structure of RMS is usually composed of serial workstations, each of which is often equipped with extra resources. These resources are employed to enhance the production process. Designing such a system consists in assigning a set of manufacturing tasks and allocating extra resources to workstations while respecting technical constraints and aiming to optimize specific production objectives.

The design of an RMS has a significant impact on its operational performance. In the scope of this study, the productivity and robustness of an RMS receive special consideration. Productivity is defined here as the number of products manufactured within a given period of time, whereas robustness is the capacity of the system to maintain its productivity despite variability in task processing time (see, *e.g.*, Sotskov et al. [4]). Thus, while improving productivity aims to decrease the time between two successive finished products, enhancing robustness focuses on increasing this time to guarantee that the system can cope with the mentioned variability. These two goals are antagonistic, but both important since a higher level of productivity leads to greater profitability and a higher level of robustness reduces losses. In order to address the studied objectives properly, it is necessary to adopt a multi-objective approach. In this paper, inspired by Cerqueus and Delorme [5], we

propose a bi-objective based indicator to evaluate the evolution of productivity and robustness that an RMS could achieve at the design stage. The proposed indicator can also be viewed as a scalability measure. To find an RMS design with the largest value of the proposed indicator, an appropriate simulated annealing algorithm is developed and approved on a set of benchmark instances from the literature.

The remaining sections of the paper are organized as follows. Section 2 presents the literature related to scalability aspects and robustness in RMS. Section 3 introduces the studied problem. Section 4 provides some details on the use of extra resources in the design stage and an explanation of the employed scalability measure. The developed simulated annealing is detailed in Section 5. Numerical results are provided and analyzed in Section 6. Conclusion and perspectives are presented in Section 7.

## 2. Related literature review

Koren et al. [6] affirm that scalability could be considered as the most relevant characteristic of an RMS, since it has a significant impact on the profitability of manufacturing companies. According to Putnik et al. [7], scalability is defined as the ability of the production system to change its output quickly and progressively in order to follow market demand. This latter goal is generally achieved by adding or removing manufacturing resources. Several research studies have dealt with scalability in RMS, which we classify into two categories.

The first category aims to select an optimal system configuration or management policy in terms of capacity scalability. Koren et al. [8] and Spicer et al. [9] examine the effect of system configuration. Both studies show that configurations with parallel resources and crossover provide an interesting compromise between scalability and investment cost. Deif and ElMaraghy [10] develop a dynamic approach to determine an optimal scalability policy for various demand profiles. These policies are evaluated based on four following metrics: capacity level, WIP<sup>1</sup> level, inventory level and backlog level. In another contribution, Deif and ElMaraghy [11] propose a method for identifying an optimal scheduling of capacity scalability. This latter aims to determine when and how much to produce in order to meet demand at the lowest possible cost. Moghaddam et al. [12] develop a two-stage approach for designing a scalable RMS configuration. The first stage consists in designing an initial RMS configuration while minimizing total manufacturing costs and ensuring that demand is met. Once demand increases, the second stage enables the selection of the best possible reconfiguration with the lowest additional cost. The described approach was then extended for a family of products in Moghaddam et al. [13].

The second category, which focuses on scalability measures, is less studied in the literature. Only two papers can be noted. Thus, Koren et al. [6] suggested a metric for evaluating the scalability of a configuration based on its current state. The proposed metric reflects the minimal capacity increment that could be applied to adjust the system throughput according to market demand. The smaller the capacity increment, the more scalable the configuration. In a more recent contribution, Cerqueus and Delorme [5] provided a new measure for assessing the scalability of single-product RMS from the design phase. The proposed measure is based on a multi-objective analysis and enables to evaluate the scalability of each RMS balancing by considering all its possible configurations.

In addition to scalability, robustness is an important feature that is usually taken into account in the design phase of a manufacturing system. It has not been studied in the specific context of RMS, but has been largely investigated within the framework of SALBP<sup>2</sup> (see, *e.g.*, Battaïa and Dolgui [14]). This problem generally aims to assign a given set of assembly tasks to a given set of workstations, while satisfying some technical constraints and optimizing specific production objectives. The latter assignment is usually referred to as ‘balance’. The principal goal of the robust approach in the domain of SALBP is to find a balance that has the best performance for its worst-case realization of uncertain task processing times. It is also worth to mention that the main

---

<sup>1</sup>Work In Process

<sup>2</sup>Simple Assembly Line Balancing Problem

advantage of the robust approach over the two classical ones (stochastic and fuzzy) lies in the fact it does not require reliable historical data related to uncertainty. Thus, for example, Hazır and Dolgui [15] study a variant of the SALBP problem aiming to find a balance with the minimal load on the busiest workstation. For this problem, the number of workstations is fixed, no cycle time constraint is considered and all the tasks are supposed to be uncertain, with processing time represented as an interval. The lower (resp. upper) limit of the interval models the nominal (resp. maximal) processing time of the task involved. Based on Bertsimas and Sim's idea, the authors assume that, in the worst case, only  $\Gamma$  tasks can take their upper interval limits at the same time, and the other tasks keep their nominal processing time values. This leads to consider a particular bi-level optimization problem, for which the authors develop a decomposition approach. Rossi et al. [16] also study a SALBP variant with a fixed number of workstations and a given cycle time. However, unlike Hazır and Dolgui [15], the authors suppose that only some tasks (and not all of them) are considered as uncertain without any additional information. As an alternative robust approach, they look for a balance that can support (while remaining feasible) the greatest possible amplitude of nominal processing time increase for all uncertain tasks. The value of the mentioned amplitude is called the *stability radius*, which is considered as a possible robustness measure and can be calculated using different norms. Thus, to find such a balance, Rossi et al. [16] study the  $l_1$  and  $l_\infty$  norms for the stability radius and propose a mixed-integer linear programming (MILP) model as well as different theoretical upper bounds for each of them. A similar approach was done in Pirogov et al. [17], but for the case of transfer line balancing. Finally, in Gurevsky et al. [18], the authors, studying the same variant of SALBP as in Rossi et al. [16], propose a more appropriate robustness measure, called *stability factor*. Its main difference with respect to stability radius consists in analyzing the greatest percentage of increase and not its amplitude.

In this paper, we try to merge the ideas of Cerqueus and Delorme [5] and Gurevsky et al. [18] by integrating productivity and stability factor into one bi-objective based measure of scalability.

### 3. Problem description

Let  $\mathcal{V} = \{1, \dots, n\}$  be a set of manufacturing tasks and  $\mathcal{W} = \{1, \dots, m\}$  a set of available workstations. We consider an assembly line in a reconfigurable environment producing a single model of product. A line balance, denoted by  $B$ , is viewed as an assignment of  $n$  tasks to  $m$  workstations, such that each task is assigned to exactly one workstation and each workstation has at least one assigned task, while satisfying precedence constraints. The processing time of task  $j$  is noted as  $t_j$  and the sum of the processing time of the tasks assigned to workstation  $k$  is considered as its load and noted as  $l_k$ . Furthermore, the cycle time of workstation  $k$ , noted as  $c_k$ , refers to the theoretical minimum time required between two successive product outputs at this workstation. A workstation with the highest cycle time, known also as a bottleneck workstation, determines the production line rate, known as takt time, noted as  $T$  and computed as  $\max_{k \in \mathcal{W}} c_k$ . Productivity  $P$  is the inverse of takt time, which reflects the speed of the production process that we aim to maximize.

In the studied context, it is assumed that only a subset  $\tilde{\mathcal{V}}$  of so-called uncertain tasks can be identified. The processing time for each of these tasks is considered as variable without any additional information. The robustness of such a production line, referred to as  $R$ , is assessed on the basis of the measure proposed by Gurevsky et al. [18] and called *stability factor*. By analogy with  $l_k$ , we also introduce a notation  $\tilde{l}_k$ , which represents the sum of the processing time among the uncertain tasks assigned to workstation  $k$ . The time interval that a product spends between input at a given workstation and input at the next one is called the occupancy time, while the difference between the occupancy time and the load on that workstation is called the margin. The margin allows to cope with variations in task processing time: the greater the margin, the more robust the workstation. Following Gurevsky et al. [18], the stability factor of a workstation  $k$ , noted here as  $f_k$ , is computed as the ratio between its margin and  $\tilde{l}_k$ . In what follows, a workstation with the lowest stability factor is referred to as a "risky" workstation. This workstation determines the robustness of the whole production line, so that  $R = \min_{k \in \mathcal{W}} f_k$ .

**Example 1.** Figure 1 illustrates an example of precedence graph. Each node of this graph symbolizes a particular task. The task number (resp. nominal task processing time) is indicated inside (resp. above) the corresponding node. Grey nodes represent uncertain tasks belonging to  $\bar{\mathcal{V}}$ . Figure 2 shows a possible line balance with 5 workstations and no extra resources. The notations introduced above are depicted below each workstation. The takt time,  $T = 13$ , is determined by the bottleneck workstation 2. Thus, the productivity  $P$  of the line is equal to  $\frac{1}{13}$ . In terms of robustness, workstation 5 is the unique risky one with  $f_5 = \frac{13-9}{9} = \frac{4}{9}$  and therefore  $R = \frac{4}{9}$ , *i.e.*, the production line is able to maintain its productivity even though the processing time of every uncertain task is multiplied by  $1 + R$ .  $\square$

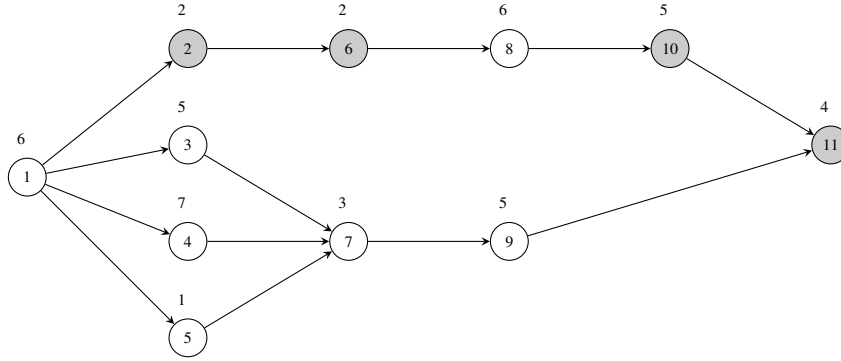


Figure 1: Precedence graph

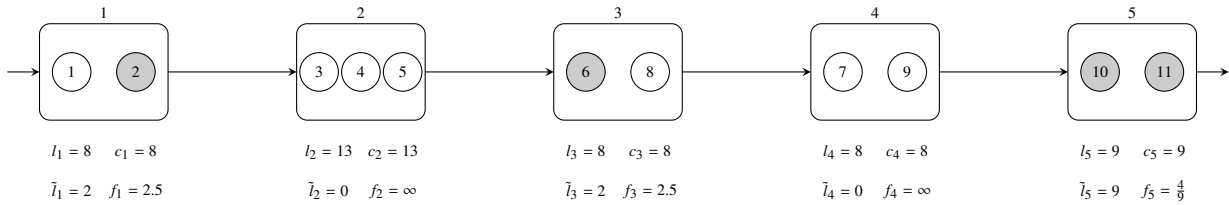


Figure 2: Example of a possible line balance

In this paper, we study the case where any RMS can be reconfigured by adding or removing extra resources in order to improve its productivity or robustness. In the following, a configuration refers to a line balance with extra resources added to workstations. Improving the productivity (resp. robustness) of the RMS starts with identifying bottleneck (resp. risky) workstations. Allocating an extra resource to such a workstation in terms of productivity (resp. robustness) can be interpreted as adding a new identical machine in parallel to (resp. adding a new buffer stock after) this workstation. The latter permits to reduce the cycle time (resp. increase the margin) of this workstation and potentially improve the productivity (resp. robustness) of the entire line. More formally, following Cerqueus and Delorme [5], Gurevsky et al. [18] and the interpretation mentioned above, the productivity  $P$  and robustness  $R$  of any configuration can be calculated as follows:

$$P = \min_{k \in \mathcal{W}} \left\{ \frac{1 + r_k^P}{l_k} \right\},$$

$$R = \min_{k \in \mathcal{W}} \left\{ \frac{(1 + r_k^P + r_k^R) \cdot T - l_k}{\bar{l}_k} \right\},$$

where  $r_k^P$  and  $r_k^R$  are the number of extra resources added to workstation  $k$  in terms of productivity and robustness, respectively.

**Example 2.** Based on the initial balance, presented in Figure 2, and having one extra resource, it is possible to generate one more productive or robust configuration. Indeed, by adding an extra resource in terms of productivity to the single bottleneck workstation 2, we reduce its cycle time to  $\frac{13}{2}$ , leading to the new bottleneck workstation 5 with  $c_5 = 9$ . This increases productivity  $P$  from  $\frac{1}{13}$  to  $\frac{1}{9}$ . On the other hand, by adding an extra resource in terms of robustness to the single risky workstation 5, we increase its stability factor from  $\frac{4}{9}$  to  $\frac{17}{9}$ , and, as a result, the robustness of the resulting configuration is also improved up to  $\frac{17}{9}$ .  $\square$

Thus, the principal goal of this paper is to design a line balance that offers the best trade-off between productivity and robustness among its possible configurations when distributing a limited number  $r_{\max}$  of extra resources. All the steps involved in such a design are presented in the next sections.

#### 4. Configurations and scalability measure

An efficient way in distributing  $r_{\max}$  extra resources for a given line balance  $B$  involves two main aspects. The first one consists in identifying bottleneck and risky workstations for  $B$  and for all its potential configurations. The second one comprises an iterative process of generating configurations using lists of configurations, noted  $\mathcal{L}_i$ ,  $i \in \{1, \dots, r_{\max}\}$ , sharing the same number  $i$  of extra resources. Each list is build from the previous one, taking into account the natural fact that  $\mathcal{L}_0 := \{B\}$ . The global process is more formally presented below by Algorithm 1.

---

##### Algorithm 1 GENERATION OF CONFIGURATIONS

---

**Input:** An initial balance  $B$  and a maximal number  $r_{\max}$  of extra resources that can be added.

**Output:** Lists of configurations.

1. Set  $i := 0$  (current number of added resources),  $\mathcal{L}_0 := \{B\}$ , and  $\mathcal{L}_i := \emptyset$ ,  $i \in \{1, \dots, r_{\max}\}$ .
  2. Choose a new configuration from  $\mathcal{L}_i$  and identify all its bottleneck and risky workstations.
  3. For each bottleneck (resp. risky) workstation, generate a new configuration by adding one resource in terms of productivity (resp. robustness) to this workstation. Then, add all the generated configurations to  $\mathcal{L}_{i+1}$ .
  4. If all the configurations from  $\mathcal{L}_i$  are explored, then go to Step 5. Otherwise, go to Step 2.
  5. Update  $i := i + 1$ . If  $i < r_{\max}$ , go to Step 2. Otherwise, stop and return  $\mathcal{L}_i$ ,  $i \in \{0, 1, \dots, r_{\max}\}$ .
- 

It is important to point out that the lists  $\mathcal{L}_i$ ,  $i \in \{1, \dots, r_{\max}\}$ , provided by Algorithm 1, can sometimes contain the configurations that are dominated in the Pareto sense (see, *e.g.*, Ehrgott [19]) by other configurations from the same list in terms of productivity and robustness. As a consequence, such dominated configurations should naturally be excluded from future consideration and from these lists. Thus, Figure 3 shows an example of the potential image of the first three lists  $\mathcal{L}_0$ ,  $\mathcal{L}_1$  and  $\mathcal{L}_2$  of non-dominated configurations in the bi-objective space of productivity  $P$  and robustness  $R$  to be maximized.

In order to compare the configurations obtained from different balances, a bi-objective based metric, noted as  $HV$ , is introduced in this paper. This metric is calculated as  $\sum_{i=0}^{r_{\max}} HV_i$ , where  $HV_i$  represents the well-known hyper-volume (see, *e.g.*, Ehrgott [19]) of  $\mathcal{L}_i$ . This latter is computed as the spatial value of the geometric area between the image of  $\mathcal{L}_i$  in the bi-objective space of productivity and robustness and a reference point, which usually represents lower bounds for these two objectives. The grey area in Figure 3 illustrates  $HV_2$ .

The higher the value of  $HV$ , the larger the scope of productivity and robustness covered by the configurations derived from the balance. Thus,  $HV$  can be viewed as an appropriate scalability measure. In the next section, we show how to identify a balance that maximizes  $HV$ .

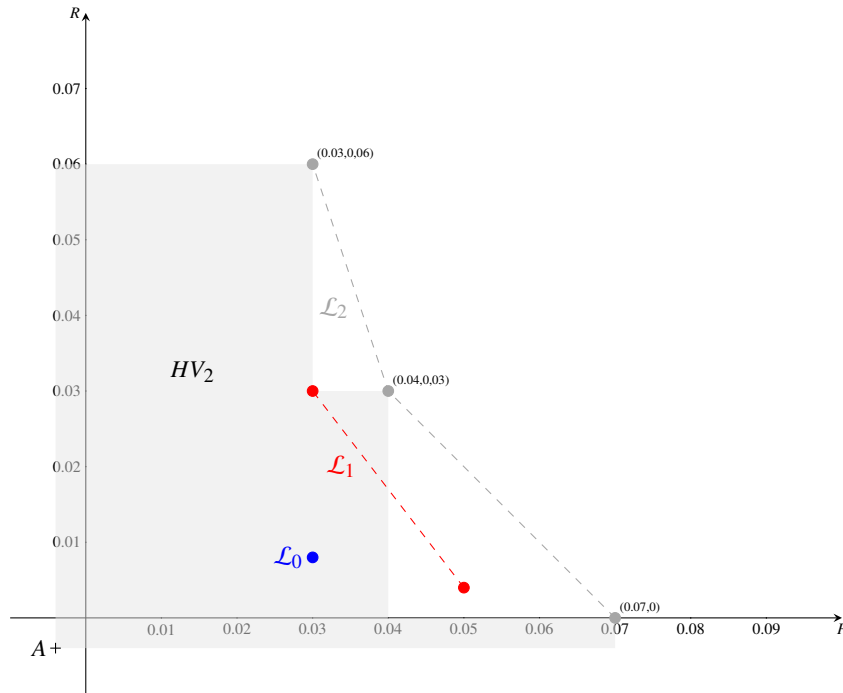


Figure 3: An example of three lists of non-dominated configurations and  $HV_2$  computation

### 5. Simulated annealing

In this study, we develop a simulated annealing algorithm for finding a line balance with the greatest value of  $HV$ . Inspired by the metallurgical annealing process (see, *e.g.*, van Laarhoven and Aarts [20, p. 7-15]), this approach is a widely used meta-heuristic for combinatorial optimization problems. The algorithm starts with an initial solution generated by a heuristic method. At each iteration a new solution is selected in the neighborhood of the current one, by applying a move. It is accepted by the algorithm if it is better according to a fitness function  $f(\cdot)$  or according to a probability depending on its quality. The algorithm stops when it reaches a given number of iterations and returns the best found solution. Algorithm 2 provides a formal description of the proposed simulated annealing.

For the studied problem, the initial solution is generated by an appropriate variant of the well-known COM-SOAL heuristic (see, *e.g.*, Arcus [21]). The fitness function  $f(\cdot)$  is represented by computing  $HV$ . The move used to generate a neighbor solution consists in randomly selecting a task and transferring it to a new workstation. A transfer move is performed without violating any precedence constraint. Figure 4 depicts an example of a transfer move where task 6 is moved from workstation 3 to workstation 1, satisfying the precedence constraints illustrated in Figure 1.

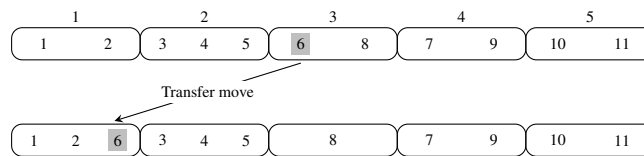


Figure 4: An example of moving task 6

---

**Algorithm 2** SIMULATED ANNEALING
 

---

**Input:** An initial temperature  $T$ , a cooling ratio  $\alpha$ , a maximum number of epochs  $E_{\max}$ , a maximum number of moves per epoch  $M_{\max}$  and an initial solution  $s^{(I)}$ .

**Output:** A new solution  $s^{(B)}$ .

1. Set  $i := 1$  (number of iterations in the epoch),  $j := 1$  (current epoch),  $s^{(B)} := s^{(I)}$  (best found solution) and  $s^{(C)} := s^{(I)}$  (current solution).
  2. If  $i > M_{\max}$ , then go to Step 3. Otherwise, construct a new feasible solution  $s^*$  by randomly and feasibly moving a task from one workstation to another. If  $f(s^*) > f(s^{(B)})$ , then update  $s^{(B)} := s^*$ . If  $f(s^*) \geq f(s^{(C)})$ , then update  $s^{(C)} := s^*$ . Otherwise, compute  $\Delta := f(s^*) - f(s^{(C)})$  and set  $s^{(C)} := s^*$  with probability  $\exp(\Delta/T)$ . Update  $i := i + 1$  and repeat Step 2.
  3. If  $j < E_{\max}$ , then consider the next epoch, *i.e.*, set  $j := j + 1$ ,  $i := 1$ ,  $T := \alpha T$  and go to Step 2. Otherwise, stop and return solution  $s^{(B)}$ .
- 

## 6. Experimental results

To experimentally evaluate our approach, we consider 525 instances with 20 tasks each, extracted from Otto et al. [22]. We generate a random subset of uncertain tasks such that  $|\tilde{\mathcal{V}}| = \lceil 0.2 \cdot |\mathcal{V}| \rceil$ . The instances are categorized into three subsets, based on the density, noted as  $OS$ , of their precedence graph: 226 instances with  $OS \approx 0.2$ , 226 instances with  $OS \approx 0.6$  and 73 instances with  $OS \approx 0.9$ . The number  $m$  of workstations is set to 5 and we limit the number of tasks assigned to a same workstation to 30% of  $n$ . Moreover, the maximum number  $r_{\max}$  of available extra resources is set to  $m$ .

The parameters of the simulated annealing are determined empirically. The initial temperature  $T$  is given as 0.028 and  $\alpha$  is fixed at 0.98. The maximum number of epochs  $E_{\max}$  is initialized to  $10n$ . The number of iterations per epoch  $M_{\max}$  depends on  $OS$ , *i.e.*,  $M_{\max} = \frac{5n}{OS}$ . Ten replications of the simulated annealing algorithm are performed for each instance. The experiments were implemented using the Python language and carried out on the LICCiA server, located in the LS2N laboratory at the University of Nantes and equipped with an Intel Xeon Platinum 8260 processor running at 2.4 GHz and having 2 GB RAM.

$OS$	#Instances	Avg. CPU, (s)	Avg. $HV$	Avg. #NDC	Avg. MRG	Avg. IMP
0.2	226	50.93	0.05	34.63	0.91	2.74
0.6	226	16.96	0.04	32.64	0.90	1.79
0.9	73	11.31	0.04	28.82	0.97	1.49
Global	525	30.80	0.04	32.97	0.91	2.16

Table 1: Computational results for the instances with  $n = 20$  and  $r_{\max} = 5$

Averaged results are summarized in Table 1. The first and second columns represent the subsets of instances with respect to  $OS$  and the total number of instances per subset, respectively. The average CPU time per instance is given in the third column. The fourth column indicates the average  $HV$  of the best found solution, while the fifth column provides the average number of non-dominated configurations. The column Avg. MRG is the merge indicator. It is computed for each instance as the average ratio of the  $HV$  of the returned solution over the  $HV$  of the best found one over the 10 replications. The last column is the improvement indicator. It shows the ratio between the  $HV$  of the best found solution over the  $HV$  of the initial solution.

The table shows that the resolution time is negatively correlated with  $OS$ , which was expected since the number of iterations per epoch depends on  $OS$ . Thus, the less dense the precedence graph, the longer it takes



to resolve a corresponding instance. The improvement indicator shows a significant enhancement of the  $HV$  metric, which is doubled on average. Moreover, the obtained balances present a large number of possible non-dominated configurations, offering the decision-maker multiple options. It is noticeable, however, that instances with higher  $OS$  values tend to have fewer non-dominated configurations.

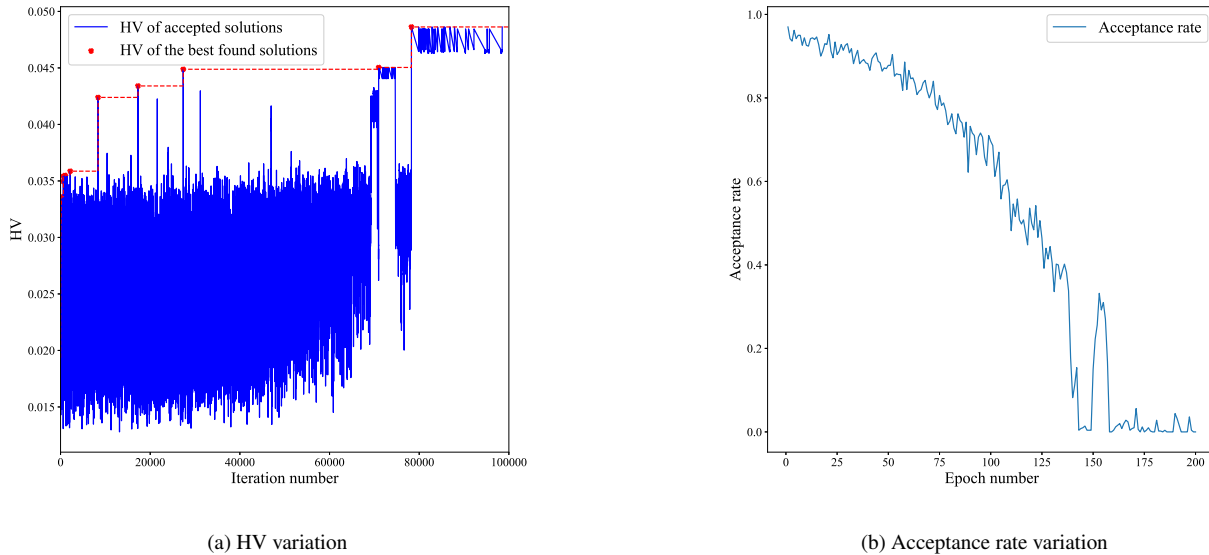


Figure 5: Simulated annealing performance

The merge indicator is on average 0.92, which confirms the stability of the simulated annealing algorithm. The blue-colored curve in Figure 5a illustrates the  $HV$  evolution of accepted solutions, while the red-colored curve presents the  $HV$  evolution of the best found ones, for a representative instance. We can observe that the algorithm tends to tolerate more degrading solutions during the initial iterations. This observation is confirmed by the evolution of the acceptance rate depicted in Figure 5b. The acceptance rate is computed, for each epoch of simulated annealing, as the ratio between the number of accepted solutions and the total number of generated ones. As shown in Figure 5b, this ratio is higher than 0.9 at the initial iterations, then decreases and converges to zero at the end of the algorithm.

Overall, the obtained results highlight the stability and efficiency of the proposed approach.

## 7. Conclusion and perspectives

The main contribution of this paper is the introduction and study of a new scalability measure for evaluating the feasible balances of any assembly line in a reconfigurable environment and in the presence of tasks whose processing time is uncertain. The studied measure has a bi-objective character, but provides a scalar value intended to represent, for a given line balance, the quality of all its attainable non-dominated configurations in terms of productivity and robustness when the extra resources are used. The second important contribution is the development of an appropriate simulated annealing algorithm aiming to find a line balance with the greatest value of the proposed measure. It is shown that the algorithm provides good quality line balances in a relatively short period of time for small-size instances.

A potential first extension of our study would be to test the proposed approach on larger-size instances. A deeper characterisation of non-dominated configurations would allow to fasten the generation process and thus the overall resolution procedure. To get closer to industrial reality, more complex contexts should also

be analyzed, for example multi-product lines or changes in product specifications. Furthermore, it would be interesting to integrate environmental and social objectives into the analysis.

### Acknowledgements

This work was financially supported by the French National Research Agency (project ANR-21-CE10-0019 RECONFIDURABLE).

### References

- [1] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, H. V. Brussel, Reconfigurable manufacturing systems, *CIRP Annals - Manufacturing Technology* 48 (1999) 527–540.
- [2] Y. Koren, X. Gu, W. Guo, Reconfigurable manufacturing systems: Principles, design, and future trends, *Frontiers of Mechanical Engineering* 13 (2018) 121–136.
- [3] Y. Koren, M. Shpitalni, Design of reconfigurable manufacturing systems, *Journal of Manufacturing Systems* 29 (2010) 130–141.
- [4] Y. N. Sotskov, N. Y. Sotskova, T.-C. Lai, F. Werner, *Scheduling under Uncertainty: Theory and Algorithms*, Belorusskaya nauka, Minsk, 2010.
- [5] A. Cerqueus, X. Delorme, Evaluating the scalability of reconfigurable manufacturing systems at the design phase, *International Journal of Production Research* 61 (2023) 8080–8093.
- [6] Y. Koren, W. Wang, X. Gu, Value creation through design for scalability of reconfigurable manufacturing systems, *International Journal of Production Research* 55 (2017) 1227–1242.
- [7] G. Putnik, A. Sluga, H. ElMaraghy, R. Teti, Y. Koren, T. Tolio, B. Hon, Scalability in manufacturing systems design and operation: State-of-the-art and future developments roadmap, *CIRP Annals - Manufacturing Technology* 62 (2013) 751–774.
- [8] Y. Koren, S. J. Hu, T. W. Weber, Impact of manufacturing system configuration on performance, *CIRP Annals - Manufacturing Technology* 47 (1998) 369–372.
- [9] P. Spicer, Y. Koren, M. Shpitalni, D. Yip-Hoi, Design principles for machining system configurations, *CIRP Annals - Manufacturing Technology* 51 (2002) 275–280.
- [10] A. M. Deif, H. A. ElMaraghy, Assessing capacity scalability policies in RMS using system dynamics, *International Journal of Flexible Manufacturing Systems* 19 (2007) 128–150.
- [11] A. M. Deif, W. ElMaraghy, Investigating optimal capacity scalability scheduling in a reconfigurable manufacturing system, *The International Journal of Advanced Manufacturing Technology* 32 (2007) 557–562.
- [12] S. K. Moghaddam, M. Houshmand, O. F. Valilai, Configuration design in scalable reconfigurable manufacturing systems (RMS); a case of single-product flow line (SPFL), *International Journal of Production Research* 56 (2018) 3932–3954.
- [13] S. K. Moghaddam, M. Houshmand, K. Saitou, O. F. Valilai, Configuration design of scalable reconfigurable manufacturing systems for part family, *International Journal of Production Research* 58 (2020) 2974–2996.



- [14] O. Battaïa, A. Dolgui, A taxonomy of line balancing problems and their solution approaches, *International Journal of Production Economics* 142 (2013) 259–277.
- [15] O. Hazır, A. Dolgui, Assembly line balancing under uncertainty: Robust optimization models and exact solution method, *Computers & Industrial Engineering* 65 (2013) 261–267.
- [16] A. Rossi, E. Gurevsky, O. Battaïa, A. Dolgui, Maximizing the robustness for simple assembly lines with fixed cycle time and limited number of workstations, *Discrete Applied Mathematics* 208 (2016) 123–136.
- [17] A. Pirogov, E. Gurevsky, A. Rossi, A. Dolgui, Robust balancing of transfer lines with blocks of uncertain parallel tasks under fixed cycle time and space restrictions, *European Journal of Operational Research* 290 (2021) 946–955.
- [18] E. Gurevsky, A. Rasamimanana, A. Pirogov, A. Dolgui, A. Rossi, Stability factor for robust balancing of simple assembly lines under uncertainty, *Discrete Applied Mathematics* 318 (2022) 113–132.
- [19] M. Ehrgott, *Multicriteria optimization*, 2 ed., Springer-Verlag Berlin Heidelberg, 2005.
- [20] P. J. M. van Laarhoven, E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, Mathematics and Its Applications, Springer Dordrecht, 1987.
- [21] A. L. Arcus, A computer method of sequencing operations for assembly lines, *International Journal of Production Research* 4 (1965) 259–277.
- [22] A. Otto, C. Otto, A. Scholl, Systematic data generation and test design for solution algorithms on the example of SALBPGen for assembly line balancing, *European Journal of Operational Research* 228 (2013) 33–45.