Stability factor for robust balancing of simple assembly lines under uncertainty

Evgeny Gurevsky[a,*], Andry Rasamimanana[a], Aleksandr Pirogov[b,c], Alexandre Dolgui[b], André Rossi[d]

[a]*LS2N, Université de Nantes, France*
[b]*IMT Atlantique, LS2N, Nantes, France*
[c]*CENTRAL, Lancaster University Management School, UK*
[d]*LAMSADE, Université Paris-Dauphine, PSL, France*

Abstract

This paper deals with an optimization problem, which arises when a new simple assembly line has to be designed subject to a fixed number of available workstations, cycle time constraint, and precedence relations between necessary assembly tasks. The studied problem consists in assigning a given set of tasks to workstations so as to find the most robust line configuration, which can withstand processing time uncertainty as much as possible. The line robustness is measured by a new indicator, called *stability factor*. In this work, the studied problem is proven to be strongly NP-hard, upper bounds are proposed, and the relation of the stability factor with another robustness indicator, known as *stability radius*, is investigated. A mixed-integer linear program (MILP) is proposed for maximizing the stability factor in the general case, and an alternative formulation is also derived when uncertainty originates in workstations only. Computational results are reported on a collection of instances derived from classic benchmark data used in the literature for the Simple Assembly Line Balancing Problem (SALBP).

Keywords: assembly line, balancing, robustness, robust optimization, stability radius, uncertainty, MILP

---

*Corresponding author.
Email address: `evgeny.gurevsky@univ-nantes.fr` (Evgeny Gurevsky)

1. Introduction

A robust balancing of simple assembly lines with a fixed number of workstations is considered. As for the classic SALBP problem (see, *e.g.*, Scholl and Becker, 2006; Battaïa and Dolgui, 2013), the restrictions such as precedence and cycle time constraints are taken into account. The cycle time constraint is enforced by the use of a conveyor belt, that moves all the products synchronously along the line. As a result, the considered production system is a *paced* assembly line (see, *e.g.*, Öner-Közen et al., 2017), and is free of buffer limitation issues by construction. In this work, the goal is not only to obtain an admissible assignment of a given set of assembly tasks to workstations, but also to focus on the best way to maintain its feasibility despite the presence of tasks, called *uncertain*, whose processing time can vary.

Classic formulation of SALBP captures basic features only and does not always reflect particular real-world situations in manufacturing, which often need to tackle the task processing time uncertainty. Indeed, as mentioned in Battaïa and Dolgui (2013), task processing times are not exactly known at the preliminary design stage of the line and only their nominal (or estimated) values are available. This may be caused by the following practical factors:

- for manual assembly lines, the performance of operators implementing tasks, depends on their work rate, skill level, fatigue and motivation;
- product specifications as well as workstation characteristics may be changed during the line life cycle. Such changes may be motivated by a customer demand or updating the market of materials;
- various delays and micro-stoppages when tasks are executed.

Any of these events may occur in any moment of the line exploitation implying a costly line interruption, if the cycle time is exceeded. As a consequence, to construct a robust line configuration for a long term usage, the task processing time uncertainty should be anticipated at the line balancing stage.

The choice of an appropriate approach for handling the processing time of tasks strongly depends on the available information related to uncertainty. Thus, among the ways widely-used in the literature, we can distinguish the following ones: *stochastic* or *fuzzy*. For the *stochastic approach*, task processing times are represented as independent random variables with known probability distributions (see, *e.g.*, Ağpak and Gökçen, 2007; Özcan, 2010, 2018; Bentaha et al., 2014). Concerning the *fuzzy approach*, the potential task processing time values are represented as a fuzzy set whose membership function describes their possibility distribution (see, *e.g.*, Hop, 2006; Zacharia and Nearchou, 2012, 2013).

However, it should be noted that the use of these two approaches in practice could be a difficult challenge. This is due to the fact that the available knowledge on the input data is not always sufficient to infer adequate probability or possibility distributions for all task processing times, especially if the design of the production system is planned for the first time. For example, in order to apply stochastic methods in this case, we should assume that the task processing time follows a particular probability distribution, but with unknown parameters. Then, based on scenario analysis, a discrete event simulation technique can be applied for uncertainty modeling (see, *e.g.*, Hopp and Spearman, 2011).

The use of the *discrete* or *interval* representation of uncertainty scenarios usually modifies the goal of the considered problem. Indeed, an optimal solution found for one scenario can loose its optimality and even its feasibility for another one. To get around this situation, a criterion named as *Bertsimas and Sim robustness* (Bertsimas and Sim, 2003, 2004; Bougeret et al., 2019) can be applied. Widely used in robust optimization, it aims to seek a solution remaining feasible for all scenarios and having the best performance for the worst of them provided that at most Γ uncertain problem parameters can deviate from their nominal values. For instance, this criterion with interval task processing times was studied in Gurevsky et al. (2013b) and Pereira and Álvarez-Miranda (2018) for the SALBP-1 problem, in Hazır and Dolgui (2013) for the SALBP-2 problem and in Hazır and Dolgui (2015) for U-shaped assembly line balancing. In these four papers, to find the solution mentioned above, the authors developed exact methods such as branch-and-bound, branch-and-price and Benders decomposition. A similar approach, named *min-max robustness*, was applied in Dolgui and Kovalev (2012) for SALBP-2, but with a discrete set of scenarios. In the latter work, the computational complexity of seeking the robust solution was presented for different types of precedence constraints.

The case where only a *set of uncertain tasks* (with variable processing times) can be identified without any additional information is less informative, but probably the most frequent in practice. Because of the lack of information, the methods used for the approaches referred earlier are not applicable. To evaluate a solution in such a situation, Sotskov et al. (2005, 2006), studying SALBP, have suggested a specific indicator, called as *stability radius*. Given a solution, it is calculated as the maximal magnitude of deviations of the nominal processing time of uncertain tasks for which the solution feasibility (or optimality) is maintained. The authors show that this indicator can serve as an appropriate robustness measure. Indeed, the greater its value for a solution studied, the greater the robustness of the line configuration engaged. This positive outcome has inspired several works concerning stability aspects under balancing production lines (Gurevsky et al., 2012, 2013a; Rossi et al., 2016). The paper of Rossi et al. (2016) studies a new optimization problem, which seeks a line configuration with the greatest stability radius, instead of computing it for an already existing line configuration as before. The authors prove the NP-hardness of this problem and formulate two MILP models for each norm ℓ_1 and ℓ_∞ of computing the stability radius.

The relative weakness of the approach, studied in Rossi et al. (2016), consists in the fact that the considered stability radius can be disproportional with respect to the nominal processing time of uncertain tasks, especially if there exists a large gap between their minimal and maximal values. In order to overcome this disadvantage, in this paper, we propose to study a new indicator of robustness, referred here to as *stability factor*. Compared to Rossi et al. (2016), this indicator is calculated as the maximum rate of increment (and not as the greatest absolute increment) of the nominal processing time applied for any uncertain task without compromising the admissibility of the corresponding line configuration. As a consequence, seeking a line configuration with the largest value of stability factor is the natural goal of the studied optimization problem in this paper. This choice is motivated by the fact that the tasks processed by human workers may have a processing time that varies significantly under a variety of factors, as shown in Jakob and Geyer (2006). Their study also suggests that the task processing time deviation is proportional to workload, which motivates the choice of focusing on the stability factor. Maximizing this indicator aims at preserving the assembly line throughput when inevitable variations of human worker performances occur.

The rest of the article is organized as follows. Basic definitions and properties are presented in Section 2. Section 3 is dedicated to developing a certain number of upper bounds on the optimal value of stability factor. Some relations with stability radius are provided in Section 4. Complexity aspects of the studied optimization problem, its variants and corresponding

MILP formulations are discussed in Section 5. Computational results by a MIP solver with illustrative examples are provided and analyzed in Section 6. Final remarks and conclusions are given in Section 7.

2. Basic definitions and properties

Let $V = \{1, 2, \ldots, n\}$ be the set of necessary assembly tasks and $W = \{1, 2, \ldots, m\}$ be the set of available workstations. It is also supposed here that there exist two sets of *uncertain tasks*: a set $\widetilde{V}^1 \subseteq V$ of *a priori uncertain tasks* whose processing time may deviate from its nominal value with regard to time without any additional information and a set $\widetilde{V}^2 \subseteq V$ of *a posteriori uncertain tasks* whose uncertainty is caused by a set $\widehat{W} \subseteq W$ of *uncertain workstations*. These workstations are such that any task allocated to them becomes uncertain (even if it belongs to $V \setminus \widetilde{V}^1$). Hereinafter, the set of all uncertain tasks is denoted as $\widetilde{V} = \widetilde{V}^1 \cup \widetilde{V}^2$ and any workstation from $W \setminus \widehat{W}$ is called *certain*. The presence of certain and/or uncertain workstations can be explained by the existence of assembly lines having simultaneously two types of workstations: workstations with automatic tasks executed by robots or machines and workplaces where tasks are operated manually, respectively. Supplementary notations related to the studied problem are given in Table 1.

Remark 1. *Since any decrease of task processing time cannot compromise the solution feasibility, it is sufficient to consider only non-negative task processing time deviations in this work, i.e., for all $j \in \widetilde{V}$ we have $\xi_j \in \mathbb{R}_+$.*

In order to measure the robustness of a feasible solution, we introduce the concept of stability factor by analogy with the stability radius (see Sotskov et al., 2006), which is also recalled below. Thus, the stability factor of a feasible solution $s \in F(t)$ can be defined as follows:

$$f(s, t) = \max\{\epsilon \geq 0 \mid \forall \xi \in H(\epsilon, t) \ \ (s \in F(t + \xi))\},$$

where $H(\epsilon, t) = \{\xi \in \Xi \mid \xi_j \leq \epsilon \cdot t_j, j \in \widetilde{V}\}$. In other words, $f(s, t)$ is defined as the proportionality factor defining the greatest closed n-dimensional hyperrectangle $H(\cdot)$, called *stability hyperrectangle*, representing the deviations of the uncertain task nominal processing times, for which s remains feasible.

G	is a directed acyclic graph (V, A) representing the precedence constraints between the tasks, where A is the set of arcs;
t_j	is a non-negative nominal processing time of task j;
t	is a vector expressing the nominal task processing times, *i.e.*, $(t_1, t_2, \dots, t_n)$;
$F(t)$	is the set of all feasible solutions with respect to a given vector t;
Ξ	is the set of vectors, where each of which presents possible non-negative processing time deviations for the uncertain tasks, *i.e.*, $\{\xi \in \mathbb{R}^n_+ \mid \xi_j = 0,\ j \in V \setminus \widetilde{V}\}$;
T	is the cycle time;
V_k	is the set of all tasks assigned to workstation k;
$\widetilde{V}_k$	is the set of all uncertain tasks assigned to workstation k, *i.e.*, $V_k \cap \widetilde{V}$;
$\widetilde{W}$	is the set of workstations having at least one a priori or a posteriori uncertain task;
$\mathcal{P}(j)$	is the set of all predecessors of j in G;
$\mathcal{S}(j)$	is the set of all successors of j in G;
$Q(j)$	is the interval of workstations that can process the task $j \in V$. It can be computed as follows (see Patterson and Albracht, 1975): $\left[\left\lceil \frac{t_j + \sum_{i \in \mathcal{P}(j)} t_i}{T} \right\rceil, m+1-\left\lceil \frac{t_j + \sum_{i \in \mathcal{S}(j)} t_i}{T} \right\rceil\right]$.

Table 1: Supplementary notations

Following Sotskov et al. (2006), the stability radius is defined as follows:

$$\rho(s,t) = \max\{\epsilon \geq 0 \mid \forall \xi \in B(\epsilon) \ \ (s \in F(t+\xi))\},$$

where $B(\epsilon) = \{\xi \in \Xi \mid \|\xi\| \leq \epsilon\}$. By contrast, $\rho(s,t)$ is defined as the value of the radius of the greatest closed ball $B(\cdot)$, called *stability ball*. Any element ξ of $B(\cdot)$ is evaluated based on a given norm $\|\cdot\|$. Two norms ℓ_1 ($\|\cdot\|_1$) and ℓ_∞ ($\|\cdot\|_\infty$) have been used for the stability radius in Rossi et al. (2016), where by definition $\|\xi\|_1 = \sum_{j \in \widetilde{V}} \xi_j$ and $\|\xi\|_\infty = \max_{j \in \widetilde{V}} \xi_j$. As a consequence, the notations $\rho_1(\cdot)$ and $\rho_\infty(\cdot)$ denote the stability radius in the ℓ_1 and ℓ_∞ norm, respectively. Finally, it can be observed that these robustness indicators can have an infinite value, typically when there are no uncertain tasks, and all certain tasks can be allocated to certain workstations only.

The following theorem presents a formula for computing the stability factor of any fixed line configuration in $O(n)$ time. The notation $t_j^{(\alpha)}$ is used to model a perturbed processing time with respect to a non-negative ratio α. Thus, $t_j^{(\alpha)} = (1+\alpha)t_j$, if $j \in \widetilde{V}$ and $t_j^{(\alpha)} = t_j$ otherwise.

Theorem 1. *The stability factor f for a given feasible solution is calculated as follows*

$$f = \min_{k \in \widetilde{W}} \frac{T - \sum_{j \in V_k} t_j}{\sum_{j \in \widetilde{V}_k} t_j} \tag{1}$$

and $f = +\infty$, if $\widetilde{W}$ is empty.

Proof. Let us denote the right-hand side of (1) as φ. To prove the present theorem, it needs to show that $f \geq \varphi$ and $f \leq \varphi$.

First start with $f \geq \varphi$. Let k be an uncertain workstation, exposed to stand the processing time perturbations, whose ratio β does not exceed φ. It is not difficult to see that its *perturbed* load can be expressed as follows

$$\sum_{j \in V_k} t_j^{(\beta)} = \sum_{j \in V_k} t_j + \beta \cdot \sum_{j \in \widetilde{V}_k} t_j. \tag{2}$$

Taking into account the fact that $\beta \leq \varphi$ and the inequality $\varphi \leq \frac{T - \sum_{j \in V_k} t_j}{\sum_{j \in \widetilde{V}_k} t_j}$, which is valid for any $k \in \widetilde{W}$ due to the definition of φ, we obtain

$$(2) \leq \sum_{j \in V_k} t_j + \frac{T - \sum_{j \in V_k} t_j}{\sum_{j \in \widetilde{V}_k} t_j} \cdot \sum_{j \in \widetilde{V}_k} t_j = T.$$

The latter shows that the load of the workstation k does not exceed the cycle time, provided that the perturbation ratio remains less than or equal to φ. This proves $f \geq \varphi$.

Now let us prove that $f \leq \varphi$. To do this, it is sufficient to check that any perturbation ratio $\beta > \varphi$ causes the considered feasible solution to be unfeasible.

As above, based on the definition of φ, we deduce that there exists an uncertain workstation k^* so that $\varphi = \frac{T - \sum_{j \in V_{k^*}} t_j}{\sum_{j \in \widetilde{V}_{k^*}} t_j}$. Then, we can notice that the *perturbed* load (with respect to the ratio β) of the workstation k^* violates the cycle time constraint, since

$$\sum_{j \in V_k^*} t_j^{(\beta)} = \sum_{j \in V_k^*} t_j + \beta \cdot \sum_{j \in \widetilde{V}_k^*} t_j > \sum_{j \in V_k^*} t_j + \frac{T - \sum_{j \in V_k^*} t_j}{\sum_{j \in \widetilde{V}_k^*} t_j} \cdot \sum_{j \in \widetilde{V}_k^*} t_j = T$$

that proves $f \leq \varphi$. $\square$

Theorem 1 shows how to compute the stability factor of a given solution. In the case where a solution that maximizes f is sought, we propose in the next section some upper bounds on the optimal value of stability factor.

3. Upper bounds on stability factor

The MILP formulation of the problem of maximizing the stability factor, introduced in Section 5, requires an upper bound on the stability factor. Such an upper bound can be computed as $\min\{UB_1, UB_2\}$, if $\sum_{j\in V\setminus\widetilde{V}^1} t_j \geq |W\setminus\widehat{W}| \cdot T$ and $\min\{UB_1, UB_3\}$ otherwise, where UB_1, UB_2 and UB_3 are defined below.

The first upper bound on f is denoted by UB_1 and is the minimum of the following three upper bounds denoted by UB_1^a, UB_1^b and UB_1^c, that can be computed only when $\widetilde{V}^1$ is nonempty. If $\widetilde{V}^1$ is empty, then UB_1 is set to $+\infty$ and none of the aforementioned three upper bounds can be computed. Indeed, when $\widetilde{V}^1$ is empty, deciding whether the stability factor is finite or not is strongly NP-complete, because it is equivalent to decide if all the tasks can be processed by certain workstations only, which is the decision problem of bin-packing. Assuming that $\widetilde{V}^1$ is nonempty, we define UB_1^a, UB_1^b and UB_1^c as follows. First, UB_1^a is defined as $\frac{T-\lambda_1^a}{\lambda_1^a}$, where $\lambda_1^a = \max_{j\in\widetilde{V}^1} t_j$. This upper bound is set by the longest uncertain task, whose processing time increase is limited by the cycle time. A second upper bound on f, denoted by UB_1^b, can be derived from the fact that total amount of work that the workstations can carry out is upper-bounded by $m\cdot T$. This can be written as $(1+f)\cdot\sum_{j\in\widetilde{V}^1} t_j + \sum_{j\in V\setminus\widetilde{V}^1} t_j \leq m\cdot T$. Hence f is upper-bounded by $UB_1^b = \left(m\cdot T - \sum_{j\in V\setminus\widetilde{V}^1} t_j\right) / \left(\sum_{j\in\widetilde{V}^1} t_j\right) - 1$. A third upper bound on f, denoted by UB_1^c, is based on the pigeonhole principle. It states that there exists at least one workstation that processes at least $\gamma_1 = \left\lceil \frac{|\widetilde{V}^1|}{m} \right\rceil$ uncertain tasks. Let π be a permutation of the set $\widetilde{V}^1$ such that $t_{\pi_1} \leq \cdots \leq t_{\pi_{|\widetilde{V}^1|}}$. It can then be deduced that the total processing time due to uncertain tasks in this workstation is at least $\lambda_1^c = \sum_{q=1}^{\gamma_1} t_{\pi_q}$. Hence, $UB_1^c = \frac{T-\lambda_1^c}{\lambda_1^c}$ is an upper bound on f and finally UB_1 is defined as $UB_1 = \min\{UB_1^a, UB_1^b, UB_1^c\}$.

If $\sum_{j\in V\setminus\widetilde{V}^1} t_j \geq |W\setminus\widehat{W}|\cdot T$, then certain workstations alone cannot process all certain tasks, which means that uncertain workstations have to process at least one certain task. Hence in that case, the minimum value of the maximum load over all the uncertain workstations is $\lambda_2 = \left(\sum_{j\in V} t_j - |W\setminus\widehat{W}|\cdot T\right) / |\widehat{W}|$. As a result, $UB_2 = \frac{T-\lambda_2}{\lambda_2}$ is an upper bound of f. Otherwise, if $\sum_{j\in V\setminus\widetilde{V}^1} t_j < |W\setminus\widehat{W}|\cdot T$, then an upper bound on f can be derived by observing that at most $\left\lfloor \sum_{j\in V\setminus\widetilde{V}^1} t_j/T \right\rfloor$ certain workstations have a load equal to T due to certain tasks only, which leaves at least $m - \left\lfloor \sum_{j\in V\setminus\widetilde{V}^1} t_j/T \right\rfloor$ workstations to process uncertain tasks, and by the pigeonhole principle, there exists a workstation that processes at least $\gamma_3 =$

$\left\lceil |\widetilde{V}^1| / \left(m - \left\lfloor \sum_{j \in V \setminus \widetilde{V}^1} t_j / T \right\rfloor\right) \right\rceil \geq \gamma_1$ uncertain tasks. As a result, there exists a workstation that processes at least γ_3 uncertain tasks whose total load is at least $\lambda_3 = \sum_{q=1}^{\gamma_3} t_{\pi_q}$. Hence, $UB_3 = \frac{T-\lambda_3}{\lambda_3}$ is an upper bound on f.

The upper bound described above can be computed in $O(n \log n)$ time because UB_1^c requires to sort the uncertain tasks by increasing processing time. It is implicitly defined on the full range of workstations, *i.e.*, $[1, m]$ as it always takes all the tasks and workstations into account. We extend this upper bound by computing it for all the workstation ranges of the form $[m_1, m_2]$ where $1 \leq m_1 \leq m_2 \leq m$, and m_1 is the lower bound of the allocation interval of some task, m_2 is the upper bound of the allocation interval of some task (possibly the same task as for m_1), and where we consider all the tasks $j \in V$ such that $Q(j) \subseteq [m_1, m_2]$. If none of these tasks is uncertain and $[m_1, m_2] \cap \widehat{W}$ is empty, then no upper bound on f can be derived from this particular range of workstations. This extension leads to compute the upper bound of the previous paragraph at most $n \times n$ times (because m_1 and m_2 cannot take more than n different values) and returning the lowest one, leading to an overall complexity of $O(n^3 \log n)$ for this extended upper bound on the stability factor. The computation of UB_1^c requires to sort the uncertain tasks by increasing processing time, which can be achieved in $O(n \log n)$ time. However, all the uncertain tasks can be sorted once, and the subsequent computations of UB_1^c do not require any additional sorting. As a result, the upper bound on the stability factor can be computed in $O(n^3)$.

4. Relations with other indicators

In the sequel of this section, the relations between the problems of maximizing the stability radii in the ℓ_1 and ℓ_∞ norms, introduced in (Rossi et al., 2016), and the problem of maximizing the stability factor are explored. It is first shown that in general the stability factor is not equivalent to the two aforementioned stability radii. The following two examples show that the problem of maximizing the stability factor f is neither equivalent to maximizing the stability radius in ℓ_∞-norm, nor in the ℓ_1-norm. This justifies the need to introduce a specific problem formulation in Section 5. It is also shown that when all the tasks are uncertain (or equivalently when all the workstations are uncertain), the problem of maximizing the stability radius in the ℓ_1-norm and the problem of maximizing the stability factor are equivalent.

4.1. Maximizing f is not equivalent to maximizing ρ_∞

Consider a problem instance consisting of $n = |\widetilde{V}^1| = 5$ tasks (all of them are uncertain) with nominal processing times $t = (1, 1, 1, 1, 4)$ to be processed on $m = 2$ certain workstations ($\widehat{W}$ is empty, so is $\widetilde{V}^2$, and $\widetilde{W} = W = \{1, 2\}$ since $V = \widetilde{V} = \{1, \ldots, 5\}$) with a cycle time equal to $T = 8$ units of time and no precedence constraints. In order to show that maximizing f is not equivalent to maximizing ρ_∞, we build $\mathcal{S}_1$, an optimal solution for f, and $\mathcal{S}_2$, an optimal solution for ρ_∞. Then, $\mathcal{S}_1$ (resp. $\mathcal{S}_2$) is shown to be non-optimal for ρ_∞ (resp. for f).

- The solution $\mathcal{S}_1$, shown in Figure 1, and defined by allocating tasks 1 to 4 to workstation 1 and task 5 to workstation 2 is optimal for f, as it reaches the upper bound UB_1^a introduced in Section 3. The stability factor of $\mathcal{S}_1$ is denoted by $f(\mathcal{S}_1)$ and is equal to 1, as all the tasks can have their processing time multiplied by two without compromising solution feasibility. The stability radius in the ℓ_∞-norm of $\mathcal{S}_1$, denoted by $\rho_\infty(\mathcal{S}_1)$ is 1, as one can increase the processing time of all the tasks simultaneously by 1 unit of time without compromising feasibility.

- The solution $\mathcal{S}_2$, shown in Figure 2, and defined by allocating tasks 1 to 3 to workstation 1 and tasks 4 and 5 to workstation 2 is optimal for ρ_∞, as can be checked by inspection: it is equal to 1.5. The corresponding value of the stability factor is $f(\mathcal{S}_2) = 0.6$ as the processing time of tasks 4 and 5 can be multiplied by 1.6 without violating the cycle time.

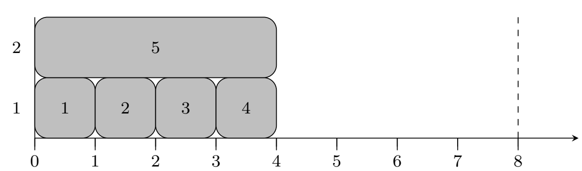

Figure 1: The solution $\mathcal{S}_1$, for which $f(\mathcal{S}_1) = 1$ and $\rho_\infty(\mathcal{S}_1) = 1$.

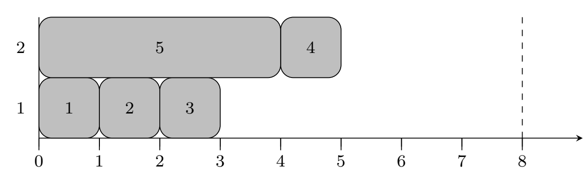

Figure 2: The solution $\mathcal{S}_2$, for which $f(\mathcal{S}_2) = 0.6$ and $\rho_\infty(\mathcal{S}_2) = 1.5$.

4.2. Maximizing f is not equivalent to maximizing ρ_1

The same problem instance is used but with $\widetilde{V}^1 = \{1, \ldots, 4\}$, which means that now, task 5 is certain. Upon this setting, the solution $\mathcal{S}_1$ defined above is optimal for ρ_1, whose value is 4. As previously, the stability factor of $\mathcal{S}_1$ is 1. The solution $\mathcal{S}_2$ has the stability radius in the ℓ_1-norm equal to 3 because the second workstation has a load equal to 5 and processes an uncertain task. However, $\mathcal{S}_2$ maximizes the stability factor which is $f(\mathcal{S}_2) = \frac{5}{3} \approx 1.667$, as the processing time of tasks 1 to 3 can be multiplied by $\frac{8}{3}$ while maintaining feasibility (the same holds for task 4 on workstation 2).

4.3. Maximizing f is equivalent to maximizing ρ_1 when all the tasks are uncertain

When $\widetilde{V} = V$ or $\widetilde{W} = W$, all the tasks are uncertain, and for any solution $\mathcal{S}$, the load of the workstation $k \in W$ is defined by $L_k(\mathcal{S}) = \sum_{j \in V_k(\mathcal{S})} t_j$, where $V_k(\mathcal{S})$ is the set of tasks that are allocated to workstation k in solution $\mathcal{S}$. Using this notation and Theorem 1 in Rossi et al. (2016), the stability radius in the ℓ_1-norm of $\mathcal{S}$ can be written as $\rho_1(\mathcal{S}) = \min\limits_{k \in W}(T - L_k(\mathcal{S})) = T - \max\limits_{k \in W} L_k(\mathcal{S})$. By Theorem 1 (see Section 2), $f(\mathcal{S}) = \min\limits_{k \in W} \left(\dfrac{T}{L_k(\mathcal{S})} - 1 \right) = \dfrac{T}{\max\limits_{k \in W} L_k(\mathcal{S})} - 1$. First, let us show by contradiction that if a solution maximizes the stability radius in the ℓ_1-norm, then it also maximizes the stability factor. Let $\mathcal{S}_1$ be such that $\rho_1(\mathcal{S}_1)$ is maximum, and assume that there exists $\mathcal{S}_2$ such that $f(\mathcal{S}_1) < f(\mathcal{S}_2)$. This implies that $f(\mathcal{S}_1) = \min\limits_{p \in W} \left(\dfrac{T}{L_p(\mathcal{S}_1)} - 1 \right) < \min\limits_{k \in W} \left(\dfrac{T}{L_k(\mathcal{S}_2)} - 1 \right) = f(\mathcal{S}_2)$. So $\dfrac{T}{\max\limits_{p \in W} L_p(\mathcal{S}_1)} - 1 < \dfrac{T}{L_k(\mathcal{S}_2)} - 1$ for all $k \in W$, hence $T - \max\limits_{p \in W} L_p(\mathcal{S}_1) < T - L_k(\mathcal{S}_2)$ for all $k \in W$, which leads to $\rho_1(\mathcal{S}_1) = T - \max\limits_{p \in W} L_p(\mathcal{S}_1) < \min\limits_{k \in W}(T - L_k(\mathcal{S}_2)) = \rho_1(\mathcal{S}_2)$. This is a contradiction, since $\rho_1(\mathcal{S}_1)$ was supposed to be maximum.

Second, let us show by contradiction that if a solution maximizes the stability factor, then it also maximizes the stability radius in the ℓ_1-norm. Let $\mathcal{S}_2$ be such that $f(\mathcal{S}_2)$ is maximum, and assume that there exists $\mathcal{S}_1$ such that $\rho_1(\mathcal{S}_2) < \rho_1(\mathcal{S}_1)$. This implies that $T - \max\limits_{p \in W} L_p(\mathcal{S}_2) < T - L_k(\mathcal{S}_1)$ for all $k \in W$, hence $\dfrac{T}{\max\limits_{p \in W} L_p(\mathcal{S}_2)} < \dfrac{T}{L_k(\mathcal{S}_1)}$ for all $k \in W$, which leads to $f(\mathcal{S}_2) = \dfrac{T}{\max\limits_{p \in W} L_p(\mathcal{S}_2)} - 1 < \min\limits_{k \in W} \left(\dfrac{T}{L_k(\mathcal{S}_1)} - 1 \right) = f(\mathcal{S}_1)$. This is a contradiction, since $f(\mathcal{S}_2)$ was supposed to be maximum.

This shows that maximizing the stability factor is equivalent to maximizing the stability radius in the ℓ_1-norm when all the tasks are uncertain. In the latter problem, the maximum load over all the workstations is $T - \rho_1$ when the tasks have their nominal processing times, so these two equivalent problems are also equivalent to SALBP-2.

4.4. Connection between f and ρ_∞

It is assumed that $\widetilde{V}^1$ is nonempty. Let $j^- \in \widetilde{V}^1$ (resp. j^+) be an uncertain task such that $t_{j^-} = \min_{j \in \widetilde{V}^1} t_j$ (resp. $t_{j^+} = \max_{j \in \widetilde{V}^1} t_j$), *i.e.*, j^- (resp. j^+) is an uncertain task of minimum (resp. maximum) processing time.

Whereas the problem of maximizing the stability factor and the stability radius in the ℓ_∞-norm have been shown to be non-equivalent, the following two properties show that when a feasible solution is available for one of these two problems, it provides a lower bound on the objective value of the other problem.

Property 1. *If the stability factor of a given feasible solution $\mathcal{S}$ is $f(\mathcal{S})$, then its stability radius in the ℓ_∞-norm is at least $t_{j^-} \cdot f(\mathcal{S})$.*

To justify Property 1, it suffices to check that increasing the nominal processing time t_j by $t_{j^-} \cdot f(\mathcal{S})$ for all $j \in \widetilde{V}^1$ yields this processing time less than or equal to $(1 + f(\mathcal{S})) \cdot t_j$. Consequently, the stability radius value in the ℓ_∞-norm of $\mathcal{S}$ is at least $t_{j^-} \cdot f(\mathcal{S})$.

Property 2 (corollary of Property 1)**.** *If the stability radius in the ℓ_∞-norm of a given feasible solution $\mathcal{S}$ is $\rho_\infty(\mathcal{S})$, then its stability factor is at least $\frac{\rho_\infty(\mathcal{S})}{t_{j^+}}$.*

The proof of Property 2 is obtained by the same argument as in the proof of Property 1.

5. Problem formulations

In this section, we formulate two mixed integer linear programs for building a line configuration that maximizes the stability. The following decision variables are used: f is the stability factor to maximize, x_{jk} is a binary variable that is set to one if and only if the task j is allocated to the workstation k, and α_{jk} is the increment rate of the nominal processing time of the task j on the workstation k.

The main idea of the first MILP formulation, referred to as **Model-1**, is that the nominal processing time of all uncertain tasks can be multiplied by $1+f$ while maintaining feasibility.

$$
\begin{aligned}
\text{Maximize} \quad & f \\
\sum_{k\in W} x_{jk} = 1 \qquad & \forall j \in V && (3)\\
\alpha_{jk} \le UB \cdot x_{jk} \qquad & \forall j \in V,\ \forall k \in W && (4)\\
f = \sum_{k\in W} \alpha_{jk} \qquad & \forall j \in V && (5)\\
\sum_{j\in V} t_j x_{jk} + \sum_{j\in V} t_j \alpha_{jk} \le T \qquad & \forall k \in \widehat{W} && (6)\\
\sum_{j\in V} t_j x_{jk} + \sum_{j\in \widetilde{V}^1} t_j \alpha_{jk} \le T \qquad & \forall k \in W \setminus \widehat{W} && (7)\\
\sum_{q=k}^{m} x_{iq} \le \sum_{q=k}^{m} x_{jq} \qquad & \forall (i,j) \in A,\ \forall k \in W \setminus \{1\} && (8)\\
x_{jk} = 0 \qquad & \forall j \in V,\ \forall k \notin Q(j) && (9)\\
f \ge 0 \qquad & \\
\alpha_{jk} \ge 0 \qquad & \forall j \in V,\ \forall k \in W \\
x_{jk} \in \{0,1\} \qquad & \forall j \in V,\ \forall k \in W
\end{aligned}
$$

Constraints (3) imply that any task has to be assigned to exactly one workstation. Constraints (4) together with (3) ensure that only for one $k \in W$, α_{jk} is non-zero for any fixed $j \in V$ and state that the increment rate of any uncertain task is upper-bounded by the constant UB. The numerical value of UB is discussed in the next paragraph. Equalities (5) state that the increment rate of any task is set to the stability factor value f. The perturbed load of each workstation cannot exceed the cycle time, as shown in constraints (6) and (7). The precedence constraints are expressed by constraints (8). Constraints (9) induce that the task j can only be allocated to a restricted set of workstations denoted by the interval $Q(j)$, introduced in Section 2.

It can be seen that the considered optimization problem is strongly NP-hard. This can be explained by the fact that the decision problem of seeking whether there exists a feasible assignment of tasks to workstations having the value of stability factor greater than or equal to zero is equivalent to the bin-packing problem.

In order to set the numerical value of UB, used in inequality (4), it should be recalled that the upper bound on the stability factor, introduced in Section 3, can be infinite: this may happen only when $\widetilde{V}^1$ is empty. When this bound is finite, UB is simply set to this value. But when it is not the case, setting UB to an overly large value like `DBL_MAX` (a constant defined in `float.h`, whose numerical value is around $1.79 \cdot 10^{308}$) is very likely to lead to wrong results caused by numerical instability. This issue is overcome by observing that the largest finite value of the stability factor is reached when processing the shortest task alone on an uncertain workstation, in that case $(1 + f) = T/\min_{j \in V} t_j$, so $T/\min_{j \in V} t_j$ is strictly larger than any possible finite value for f. We then set UB to $T/\min_{j \in V} t_j$ when the upper bound is infinite, and perform a simple post-optimization test once **Model-1** is solved. If its objective value is strictly less than UB, then the objective value is correct, as UB was actually a proper upper bound on f. But if the objective value of **Model-1** is equal to UB, then this indicates that f is infinite, as it is strictly larger than the largest finite value achievable for f. Such a situation happens only when all the tasks are certain, and are processed by certain workstations only, leaving uncertain workstations with no task to process. The x_{jk} variables computed by solving **Model-1** are then correct, but the objective value is not, as it is artificially upper bounded by the finite constant UB. Consequently, the objective value is simply set to $+\infty$. Section 6 reports some instances where such situations arise.

Model-1 is appropriate when tasks processing time increase may be due to both the presence of uncertain tasks and uncertain workstations. However, when uncertainty is due to uncertain workstations only, (*i.e.*, when $\widetilde{V}^1$ is empty) we propose a second model, referred to as **Model-2**, to address this particular case more efficiently.

Model-2 is an extension of the problem SALBP-2, where the maximum load L is minimized for uncertain workstations only, whereas the load of the workstations in $W \backslash \widehat{W}$ should be less than or equal to the constant cycle time T. The stability factor f is computed indirectly, because of its non-linear expression as a function of T. Indeed, if $L > T$, then the problem is infeasible as the cycle time constraint is not satisfied. Otherwise, $f = \frac{T-L}{L}$, as the processing time of all the tasks allocated to an uncertain workstation can be multiplied by $1 + f$ while keeping the solution feasible. Note that whenever $\widetilde{V}^1 = V$ or $\widehat{W} = W$, the problem is exactly SALBP-2 (Scholl and Becker, 2006). **Model-2** is formulated as follows.

$$
\begin{aligned}
\text{Minimize} \quad & L \\
\sum_{k \in W} x_{jk} = 1 \qquad & \forall j \in V \\
\sum_{j \in V} t_j x_{jk} \leq L \qquad & \forall k \in \widehat{W} \\
\sum_{j \in V} t_j x_{jk} \leq T \qquad & \forall k \in W \setminus \widehat{W} \\
\sum_{q=k}^{m} x_{iq} \leq \sum_{q=k}^{m} x_{jq} \qquad & \forall (i,j) \in A,\ \forall k \in W \setminus \{1\} \\
x_{jk} = 0 \qquad & \forall j \in V,\ \forall k \notin Q(j) \\
L \geq 0 \qquad & \\
x_{jk} \in \{0,1\} \qquad & \forall j \in V,\ \forall k \in W
\end{aligned}
$$

The problem formulations proposed in this section aim at finding optimal solutions. Since the problem is NP-hard, large instances may not be solved (or approached within an acceptable accuracy) in a reasonable amount of time. In such cases, one may resort to heuristics, and the reader is referred to Gurevsky et al. (2012), where fast heuristics have been proposed in the context of stability radius maximization. These heuristics may easily be adapted to the maximization of the stability factor.

6. Computational results

In order to test the proposed MILP formulations, we use the same set of 25 instances[1] as in Rossi et al. (2016). For these instances, we set respectively the number of workstations and the cycle time to $m = \left\lceil 1.2 \frac{\sum_{j \in V} t_j}{T} \right\rceil$ and $T = 1.5 \max_{j \in V} t_j$, except for the instance `WEE-MAG` where m has been set to 60 instead of 45, for the sake of feasibility. The set $\widetilde{V}^1$ (resp. $\widehat{W}$) is built by taking the first $|\widetilde{V}^1|$ (resp. $|\widehat{W}|$) elements of a random permutation of $\{1, \dots, n\}$ (resp. $\{1, \dots, m\}$) associated with each instance. We generate 13 series (each one derived from the 25 aforementioned instances) by varying the number of a priori uncertain tasks and the number of uncertain workstations such that $|\widetilde{V}^1| \in \{0, \lceil \frac{n}{4} \rceil, \lceil \frac{n}{2} \rceil, \lceil \frac{3n}{4} \rceil\}$ and $|\widehat{W}| \in \{0, \lceil \frac{m}{4} \rceil, \lceil \frac{m}{2} \rceil, \lceil \frac{3m}{4} \rceil, m\}$. The computational results are carried out on a laptop equipped with an Inter Core i7-8665U

[1] `http://pagesperso.ls2n.fr/~gurevsky-e/data/R-ALBP.zip`

processor at 1.90 GHz and 32 GB RAM. GUROBI 9.0.2 with default parameters is used as a MIP solver. The maximum solution time has been limited to 600 seconds per instance. The detailed results of each series are given in Appendix A. All these tables are built as follows. The first four columns indicate respectively the instance name, the number of tasks, the number of workstations, and the cycle time. The next two columns are respectively the lower bound (LB) and the upper bound (UB) of the stability factor found by GUROBI. Column 7 reports the `CPU` time in seconds for the instances solved to optimality. If an instance is not solved to optimality within the time limit, the abbreviation '**NF**' (not finished) is indicated. The last column displays the `GAP`, computed as $(UB - LB)/LB$. The last row of each table displays at first the number of instances solved to optimality, the average `CPU` time over all the 25 instances and the average `GAP`. Tables A.1 to A.3 give the results of solving **Model-1** for $|\widehat{W}| = 0$ with 3 combinations of $|\widetilde{V}^1|$. Similarly, Tables A.4 to A.6 and Tables A.7 to A.9 present respectively the results of **Model-1** for $|\widehat{W}| = \lceil \frac{m}{4} \rceil$ and $|\widehat{W}| = \lceil \frac{m}{2} \rceil$. Tables A.10 to A.13 show the results of solving **Model-1** with no uncertain tasks, and 4 combinations of $|\widehat{W}|$. Finally, Tables A.14 to A.17 give the results of solving **Model-2** on Series 10 to 13, which allows to compare the two problem formulations when $|\widetilde{V}^1| = 0$.

For the sake of illustrating the results presented in these tables, the solutions provided for the instance `JACKSON` with $n = 11$, $m = 6$ and $T = 10.5$ are shown as Gantt charts in Figures 4 – 7. Tasks are represented by rectangles. The white (resp. gray) rectangles corresponds to the certain (resp. uncertain) tasks. The width of a rectangle provides the processing time for the corresponding task. The hatched rectangles show the increment of the processing time of uncertain tasks, located on their left. The tasks in the instance `JACKSON` are subject to the precedence constraints given in Figure 3.

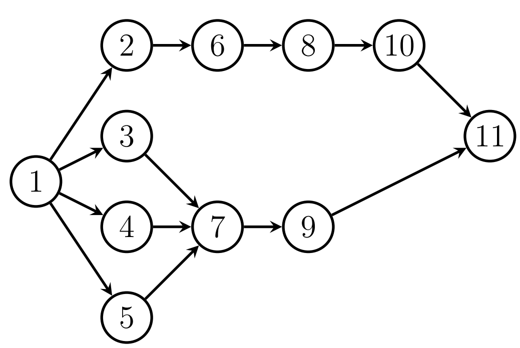

Figure 3: Precedence constraints of the instance `JACKSON`

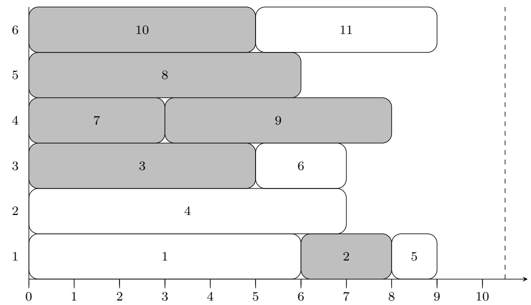

Figure 4: $f = 0.3$ for `JACKSON` with $\widetilde{V}^1 = \{2, 3, 7, 8, 9, 10\}$ and $\widehat{W} = \emptyset$.

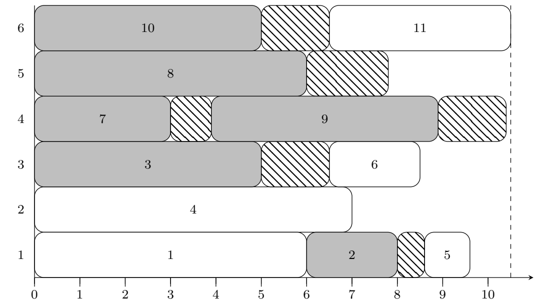

Figure 5: The solution shown in Figure 4 when all uncertain tasks reach their maximum processing time.

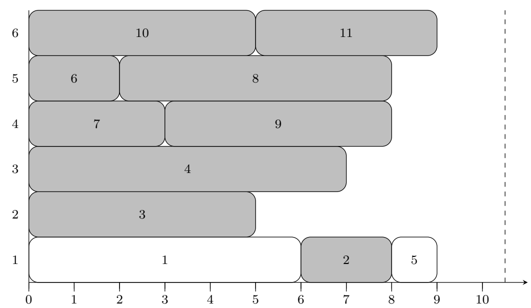

Figure 6: $f = 0.167$ for `JACKSON` with $\widetilde{V}^1 = \{2, 3, 7, 8, 9, 10\}$ and $\widehat{W} = \{3, 5, 6\}$.

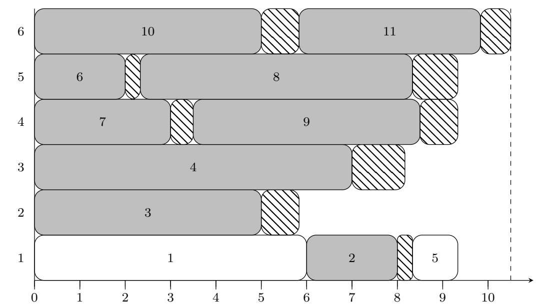

Figure 7: The solution shown in Figure 6 when all uncertain tasks reach their maximum processing time.

For Table A.2, half of tasks are uncertain and there is no uncertain workstation. Figure 4 shows an optimal solution, and Figure 5 illustrates the impact of an increase of processing times associated with the optimal value of the stability factor. It can be seen that the workstation 6 limits the value of f.

In Table A.8, $|\widetilde{V}^1| = \lceil \frac{n}{2} \rceil$ and $|\widehat{W}| = \lceil \frac{m}{2} \rceil$. As it can be remarked in Figure 6, the tasks 4, 6 and 11 become uncertain, since they are assigned to uncertain workstations. In Figure 7 the value of f is also limited by the workstation 6.

If the upper bound on the stability factor is simply computed on the full range of workstations only, (*i.e.*, from 1 to m), it may sometimes be found to be infinite when it is actually not. This happens with `BOWMAN8` when $|\widetilde{V}^1| = 0$ and $|\widehat{W}| = \lceil \frac{m}{4} \rceil$ because there are $m = 4$ workstations with a cycle time $T = 25.5$ and a single one is uncertain (workstation 3). Since the sum of the processing times is 75, we have $\lceil 75/T \rceil = 3 \leq m - |\widehat{W}|$ which does not allow to obtain a finite upper bound, as all the tasks may possibly be processed by certain workstations. However, when computing the upper bound on the stability factor on the range of workstations $[2, 4]$, the tasks allocation intervals are such that all the tasks except the first one have to be processed by workstations 2 to 4, which leads to $\lceil (75 - t_1)/T \rceil = 3$, and the upper bounds on f drops to 0.961, because now we know that the uncertain workstation has to process at least one task, which implies that the stability factor is finite. Since the question of deciding whether the stability factor is finite or not is strongly NP-complete, it may happen that we produce an infinite upper bound on f while it is finite. In order to illustrate how such a situation is dealt with, we assume in the sequel of this paragraph that the upper bound on f is computed on the full range of workstations only. In Table A.10 (when $|\widetilde{V}^1| = 0$ and $|\widehat{W}| = \lceil \frac{m}{4} \rceil$), the value of the upper bound on f is infinite for both `BOWMAN8` and `MANSOOR`, so the constant UB in constraint (4) of **Model-1** is set to $T/\min_{j \in V} t_j$, *i.e.*, 8.5 for `BOWMAN8` and 33.75 for `MANSOOR`. The solution value of **Model-1** obtained for `BOWMAN8` is 0.4166, so since it is strictly less than UB, this value is not modified. However, after solving **Model-1** with `MANSOOR`, the objective value is 33.75, and is equal to UB. Consequently, the stability factor for this instance is not 33.75, but it is infinite, as can be seen in Table A.10. The solution found by the solver is indeed such that all the tasks are allocated to three certain workstations, and the unique uncertain workstation does not process any task.

A synthesis of the results for **Model-1** reported in Tables A.1 to A.13 is shown in Table 2, where the first three columns display the main characteristics of each series, the fourth one presents the number of instances solved to optimality and the last two show the average `CPU` time (resp. the average `GAP`) over the 25 instances of the series. Table 2 clearly shows that the difficulty of seeking an optimal solution with the maximum stability factor increases with respect to the amount of uncertain tasks, *i.e.*, with $|\widetilde{V}^1|$ and $|\widehat{W}|$. Indeed, for Series 1 to 3, with $|\widehat{W}| = 0$, we can see that the number of obtained optimal solutions decreases when the number of uncertain tasks increases. The same observation can be done for Series 4 to 6, 7 to 9 and 10 to 13. Numerical results show that the used commercial solver can find an optimal solution with **Model-1** in less than 10 minutes for 65.8% of considered instances.

A synthesis of the results for **Model-2** reported in Tables A.14 to A.17 is shown in Table 3. It can be seen that the `CPU` time to solve **Model-2** is twice less than with **Model-1**. The gap is also 5 times less with **Model-2**. This confirms that whenever $|\widetilde{V}^1| = 0$, **Model-2** should be used instead of **Model-1**.

Series	$\lvert\widetilde{V}^1\rvert$	$\lvert\widehat{W}\rvert$	#OPT	Avg. CPU	Avg. GAP
1	$\lceil\frac{n}{4}\rceil$	0	23	49.58	0.004
2	$\lceil\frac{n}{2}\rceil$	0	18	172.83	0.052
3	$\lceil\frac{3n}{4}\rceil$	0	15	241.99	0.074
4	$\lceil\frac{n}{4}\rceil$	$\lceil\frac{m}{4}\rceil$	22	81.13	0.049
5	$\lceil\frac{n}{2}\rceil$	$\lceil\frac{m}{4}\rceil$	18	188.52	0.053
6	$\lceil\frac{3n}{4}\rceil$	$\lceil\frac{m}{4}\rceil$	17	213.52	0.082
7	$\lceil\frac{n}{4}\rceil$	$\lceil\frac{m}{2}\rceil$	18	179.10	0.110
8	$\lceil\frac{n}{2}\rceil$	$\lceil\frac{m}{2}\rceil$	17	212.57	0.126
9	$\lceil\frac{3n}{4}\rceil$	$\lceil\frac{m}{2}\rceil$	15	247.44	0.144
10	0	$\lceil\frac{m}{4}\rceil$	19	182.21	0.076
11	0	$\lceil\frac{m}{2}\rceil$	16	237.20	0.072
12	0	$\lceil\frac{3m}{4}\rceil$	15	261.25	0.084
13	0	m	14	265.95	0.163

Table 2: Summary of computational results for **Model-1**

Series	$\lvert\widetilde{V}^1\rvert$	$\lvert\widehat{W}\rvert$	#OPT	Avg. CPU	Avg. GAP
10	0	$\lceil\frac{m}{4}\rceil$	22	74.47	0.015
11	0	$\lceil\frac{m}{2}\rceil$	21	100.03	0.013
12	0	$\lceil\frac{3m}{4}\rceil$	20	122.70	0.011
13	0	m	20	124.16	0.009

Table 3: Summary of computational results for **Model-2**

7. Conclusion and perspectives

This paper deals with a robust balancing of simple assembly lines. It consists in finding a line configuration that maximizes the stability factor subject to restricted number of workstations, fixed cycle time, precedence constraints, and task processing time uncertainty. The corresponding problem has been proven to be strongly NP-hard, upper bounds were proposed, and the new problem has been shown to be non equivalent to the stability radius maximization. Two MILP formulations have been proposed and compared to address it, and promising experimental results have been obtained.

The proposed MILP models are a first attempt to address the studied problem. Our future research will be focused on the implementation of an efficient branch-and-bound procedure based on the developed upper bounds.

Acknowledgments

This work was partially supported by the council of the French region “Pays de la Loire”.

References

Ağpak, K., Gökçen, H., 2007. A chance-constrained approach to stochastic line balancing problem. European Journal of Operational Research 180 (3), 1098–1115.

Battaïa, O., Dolgui, A., 2013. A taxonomy of line balancing problems and their solution approaches. International Journal of Production Economics 142 (2), 259–277.

Bentaha, M. L., Battaïa, O., Dolgui, A., 2014. A sample average approximation method for disassembly line balancing problem under uncertainty. Computers & Operations Research 51, 111–122.

Bertsimas, D., Sim, M., 2003. Robust discrete optimization and network flows. Mathematical Programming 98 (1-3), 49–71.

Bertsimas, D., Sim, M., 2004. The price of robustness. Operations Research 52 (1), 35–53.

Bougeret, M., Pessoa, A. A., Poss, M., 2019. Robust scheduling with budgeted uncertainty. Discrete Applied Mathematics 261, 93–107.

Dolgui, A., Kovalev, S., 2012. Scenario based robust line balancing: Computational complexity. Discrete Applied Mathematics 160 (13-14), 1955–1963.

Gurevsky, E., Battaïa, O., Dolgui, A., 2012. Balancing of simple assembly lines under variations of task processing times. Annals of Operations Research 201 (1), 265–286.

Gurevsky, E., Battaïa, O., Dolgui, A., 2013a. Stability measure for a generalized assembly line balancing problem. Discrete Applied Mathematics 161 (3), 377–394.

Gurevsky, E., Hazır, O., Battaïa, O., Dolgui, A., 2013b. Robust balancing of straight assembly lines with interval task times. Journal of the Operational Research Society 64 (1), 1607–1613.

Hazır, O., Dolgui, A., 2013. Assembly line balancing under uncertainty: Robust optimization models and exact solution methods. Computers & Industrial Engineering 65 (2), 261–267.

Hazır, O., Dolgui, A., 2015. A decomposition based solution algorithm for U-type assembly line balancing with interval data. Computers & Operations Research 59, 126–131.

Hop, N. V., 2006. A heuristic solution for fuzzy mixed-model line balancing problem. European Journal of Operational Research 168 (3), 798–810.

Hopp, W. J., Spearman, M. L., 2011. Factory Physics, 3rd Edition. Waveland Press.

Jakob, M., Geyer, M., 2006. The influence of machine speed on human performance for simple and highly repetitive work processes: A pilot study. Agricultural Engineering International: CIGR Journal 8, 1–8.

Öner-Közen, M., Minner, S., Steinthaler, F., 2017. Efficiency of paced and unpaced assembly lines under consideration of worker variability - A simulation study. Computers & Industrial Engineering 111, 516–526.

Özcan, U., 2010. Balancing stochastic two-sided assembly lines: A chance-constrained, piecewise-linear, mixed integer program and a simulated annealing algorithm. European Journal of Operational Research 205 (1), 81–97.

Özcan, U., 2018. Balancing stochastic parallel assembly lines. Computers & Operations Research 99, 109–122.

Patterson, J. H., Albracht, J. J., 1975. Assembly-line balancing: Zero-one programming with Fibonacci search. Operations Research 23 (1), 166–172.

Pereira, J., Álvarez-Miranda, E., 2018. An exact approach for the robust assembly line balancing problem. Omega 78, 85–98.

Rossi, A., Gurevsky, E., Battaïa, O., Dolgui, A., 2016. Maximizing the robustness for simple assembly lines with fixed cycle time and limited number of workstations. Discrete Applied Mathematics 208, 123–136.

Scholl, A., Becker, C., 2006. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. European Journal of Operational Research 168 (3), 666–693.

Sotskov, Y., Dolgui, A., Portmann, M.-C., 2006. Stability analysis of an optimal balance for an assembly line with fixed cycle time. European Journal of Operational Research 168 (3), 783–797.

Sotskov, Y., Dolgui, A., Sotskova, N., Werner, F., 2005. Stability of optimal line balance with given station set. In: Dolgui, A., Soldek, J., Zaikin, O. (Eds.), Supply Chain Optimisation: Product/Process Design, Facility Location and Flow Control. Vol. 94 of Applied Optimization. Springer US, Ch. 10, pp. 135–149.

Zacharia, P. Th., Nearchou, A. C., 2012. Multi-objective fuzzy assembly line balancing using genetic algorithms. Journal of Intelligent Manufacturing 23 (3), 615–627.

Zacharia, P. Th., Nearchou, A. C., 2013. A meta-heuristic algorithm for the fuzzy assembly line balancing type-E problem. Computers & Operations Research 40 (12), 3033–3044.

Appendix A. Detailed computational results

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0.0	0.0	0.03	0
BOWMAN8	8	4	25.5	0.375	0.375	0.01	0
MANSOOR	11	4	67.5	1.812	1.812	0.01	0
JAESCHKE	9	5	9	0.2	0.2	0.00	0
JACKSON	11	6	10.5	0.3	0.3	0.01	0
MITCHELL	21	7	19.5	0.833	0.833	0.07	0
ROSZIEG	25	8	19.5	0.5	0.5	0.01	0
HESKIA	28	8	162	0.514	0.514	0.01	0
LUTZ1	32	9	2100	0.5	0.5	0.01	0
BUXEY	29	11	37.5	0.5	0.5	0.03	0
SAWYER30	30	11	37.5	0.974	0.974	0.31	0
GUNTHER	35	10	60	0.45	0.45	0.04	0
HAHN	53	7	2662.5	0.793	0.793	0.07	0
KILBRID	45	9	82.5	0.5	0.5	0.03	0
TONGE70	70	18	234	0.5	0.5	0.39	0
WARNECKE	58	24	79.5	0.529	0.529	0.98	0
ARC83	83	17	5536.5	1.021	1.046	**NF**	0.024
LUTZ3	89	18	111	0.5	0.5	0.41	0
BARTHOLD	148	12	574.5	0.5	0.5	0.70	0
MUKHERJE	94	20	256.5	0.5	0.5	0.34	0
ARC111	111	22	8533.5	0.687	0.687	3.15	0
LUTZ2	89	38	15	0.5	0.5	22.70	0
WEE-MAG	75	60	40.5	0.558	0.558	3.38	0
BARTHOL2	148	41	124.5	0.5	0.5	6.74	0
SCHOLL	297	41	2079	0.892	0.949	**NF**	0.064
				#OPT:	**23/25**	**49.58**	**0.004**

Table A.1: Results with **Model-1** for $|\widetilde{V}^1| = \lceil \frac{n}{4} \rceil$ and $|\widehat{W}| = 0$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0	0	0.02	0
BOWMAN8	8	4	25.5	0.214	0.214	0.00	0
MANSOOR	11	4	67.5	0.691	0.691	0.01	0
JAESCHKE	9	5	9	0.125	0.125	0.00	0
JACKSON	11	6	10.5	0.3	0.3	0.01	0
MITCHELL	21	7	19.5	0.312	0.312	0.05	0
ROSZIEG	25	8	19.5	0.312	0.312	0.13	0
HESKIA	28	8	162	0.434	0.434	4.60	0
LUTZ1	32	9	2100	0.5	0.5	0.04	0
BUXEY	29	11	37.5	0.389	0.389	1.35	0
SAWYER30	30	11	37.5	0.464	0.464	5.34	0
GUNTHER	35	10	60	0.341	0.341	4.68	0
HAHN	53	7	2662.5	0.401	0.401	0.14	0
KILBRID	45	9	82.5	0.5	0.5	0.05	0
TONGE70	70	18	234	0.345	0.345	90.30	0
WARNECKE	58	24	79.5	0.347	0.428	**NF**	0.232
ARC83	83	17	5536.5	0.425	0.431	**NF**	0.013
LUTZ3	89	18	111	0.382	0.382	7.20	0
BARTHOLD	148	12	574.5	0.436	0.439	**NF**	0.006
MUKHERJE	94	20	256.5	0.357	0.357	1.84	0
ARC111	111	22	8533.5	0.485	0.488	**NF**	0.007
LUTZ2	89	38	15	0.154	0.271	**NF**	0.764
WEE-MAG	75	60	40.5	0.5	0.5	5.10	0
BARTHOL2	148	41	124.5	0.325	0.383	**NF**	0.18
SCHOLL	297	41	2079	0.397	0.433	**NF**	0.093
				#OPT: **18/25**		**172.83**	**0.052**

Table A.2: Results with **Model-1** for $|\widetilde{V}^1| = \lceil \frac{n}{2} \rceil$ and $|\widehat{W}| = 0$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0	0	0.02	0
BOWMAN8	8	4	25.5	0.206	0.206	0.00	0
MANSOOR	11	4	67.5	0.5	0.5	0.00	0
JAESCHKE	9	5	9	0	0	0.00	0
JACKSON	11	6	10.5	0.3	0.3	0.01	0
MITCHELL	21	7	19.5	0.269	0.269	0.09	0
ROSZIEG	25	8	19.5	0.219	0.219	0.12	0
HESKIA	28	8	162	0.298	0.298	15.70	0
LUTZ1	32	9	2100	0.403	0.403	0.41	0
BUXEY	29	11	37.5	0.271	0.271	2.57	0
SAWYER30	30	11	37.5	0.25	0.25	3.85	0
GUNTHER	35	10	60	0.25	0.25	4.80	0
HAHN	53	7	2662.5	0.207	0.207	0.12	0
KILBRID	45	9	82.5	0.41	0.42	**NF**	0.023
TONGE70	70	18	234	0.255	0.268	**NF**	0.051
WARNECKE	58	24	79.5	0.242	0.318	**NF**	0.314
ARC83	83	17	5536.5	0.273	0.278	**NF**	0.019
LUTZ3	89	18	111	0.26	0.26	21.05	0
BARTHOLD	148	12	574.5	0.311	0.313	**NF**	0.006
MUKHERJE	94	20	256.5	0.248	0.248	1.00	0
ARC111	111	22	8533.5	0.309	0.315	**NF**	0.019
LUTZ2	89	38	15	0.125	0.173	**NF**	0.38
WEE-MAG	75	60	40.5	0.3	0.5	**NF**	0.667
BARTHOL2	148	41	124.5	0.221	0.278	**NF**	0.256
SCHOLL	297	41	2079	0.257	0.285	**NF**	0.109
				#OPT:	**15/25**	**241.99**	**0.074**

Table A.3: Results with **Model-1** for $|\widetilde{V}^1| = \lceil \frac{3n}{4} \rceil$ and $|\widehat{W}| = 0$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0	0	0.03	0
BOWMAN8	8	4	25.5	0.214	0.214	0.00	0
MANSOOR	11	4	67.5	1.75	1.75	0.01	0
JAESCHKE	9	5	9	0.125	0.125	0.01	0
JACKSON	11	6	10.5	0.3	0.3	0.01	0
MITCHELL	21	7	19.5	0.5	0.5	0.06	0
ROSZIEG	25	8	19.5	0.219	0.219	0.13	0
HESKIA	28	8	162	0.514	0.514	0.02	0
LUTZ1	32	9	2100	0.5	0.5	0.01	0
BUXEY	29	11	37.5	0.5	0.5	0.07	0
SAWYER30	30	11	37.5	0.786	0.786	1.66	0
GUNTHER	35	10	60	0.429	0.429	0.13	0
HAHN	53	7	2662.5	0.478	0.478	0.17	0
KILBRID	45	9	82.5	0.5	0.5	0.05	0
TONGE70	70	18	234	0.5	0.5	9.88	0
WARNECKE	58	24	79.5	0.529	0.529	4.86	0
ARC83	83	17	5536.5	0.665	0.71	**NF**	0.067
LUTZ3	89	18	111	0.426	0.426	5.43	0
BARTHOLD	148	12	574.5	0.5	0.5	1.35	0
MUKHERJE	94	20	256.5	0.5	0.5	1.20	0
ARC111	111	22	8533.5	0.687	0.687	39.33	0
LUTZ2	89	38	15	0.25	0.5	**NF**	1.0
WEE-MAG	75	60	40.5	0.558	0.558	4.35	0
BARTHOL2	148	41	124.5	0.5	0.5	159.49	0
SCHOLL	297	41	2079	0.817	0.948	**NF**	0.16
				#OPT:	**22/25**	**81.13**	**0.049**

Table A.4: Results with **Model-1** for $|\widetilde{V}^1| = \lceil \frac{n}{4} \rceil$ and $|\widehat{W}| = \lceil \frac{m}{4} \rceil$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0	0	0.02	0
BOWMAN8	8	4	25.5	0.214	0.214	0.00	0
MANSOOR	11	4	67.5	0.607	0.607	0.01	0
JAESCHKE	9	5	9	0.125	0.125	0.00	0
JACKSON	11	6	10.5	0.3	0.3	0.01	0
MITCHELL	21	7	19.5	0.312	0.312	0.07	0
ROSZIEG	25	8	19.5	0.219	0.219	0.17	0
HESKIA	28	8	162	0.429	0.429	2.62	0
LUTZ1	32	9	2100	0.5	0.5	0.22	0
BUXEY	29	11	37.5	0.342	0.342	1.53	0
SAWYER30	30	11	37.5	0.442	0.442	3.19	0
GUNTHER	35	10	60	0.341	0.341	4.28	0
HAHN	53	7	2662.5	0.307	0.307	0.09	0
KILBRID	45	9	82.5	0.5	0.5	0.18	0
TONGE70	70	18	234	0.34	0.34	472.90	0
WARNECKE	58	24	79.5	0.347	0.386	**NF**	0.112
ARC83	83	17	5536.5	0.34	0.341	**NF**	0.003
LUTZ3	89	18	111	0.306	0.306	5.98	0
BARTHOLD	148	12	574.5	0.436	0.439	**NF**	0.006
MUKHERJE	94	20	256.5	0.357	0.357	12.54	0
ARC111	111	22	8533.5	0.442	0.45	**NF**	0.018
LUTZ2	89	38	15	0.154	0.269	**NF**	0.746
WEE-MAG	75	60	40.5	0.5	0.5	9.21	0
BARTHOL2	148	41	124.5	0.289	0.383	**NF**	0.325
SCHOLL	297	41	2079	0.389	0.433	**NF**	0.114
				#OPT: **18/25**		**188.52**	**0.053**

Table A.5: Results with **Model-1** for $|\widetilde{V}^1| = \lceil \frac{n}{2} \rceil$ and $|\widehat{W}| = \lceil \frac{m}{4} \rceil$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0	0	0.02	0
BOWMAN8	8	4	25.5	0.206	0.206	0.00	0
MANSOOR	11	4	67.5	0.5	0.5	0.01	0
JAESCHKE	9	5	9	0	0	0.00	0
JACKSON	11	6	10.5	0.3	0.3	0.01	0
MITCHELL	21	7	19.5	0.269	0.269	0.08	0
ROSZIEG	25	8	19.5	0.219	0.219	0.08	0
HESKIA	28	8	162	0.296	0.296	3.60	0
LUTZ1	32	9	2100	0.36	0.36	0.54	0
BUXEY	29	11	37.5	0.25	0.25	1.86	0
SAWYER30	30	11	37.5	0.25	0.25	3.02	0
GUNTHER	35	10	60	0.25	0.25	3.46	0
HAHN	53	7	2662.5	0.207	0.207	0.15	0
KILBRID	45	9	82.5	0.394	0.394	301.20	0
TONGE70	70	18	234	0.251	0.259	**NF**	0.032
WARNECKE	58	24	79.5	0.242	0.309	**NF**	0.276
ARC83	83	17	5536.5	0.262	0.262	194.56	0
LUTZ3	89	18	111	0.233	0.233	12.47	0
BARTHOLD	148	12	574.5	0.31	0.313	**NF**	0.008
MUKHERJE	94	20	256.5	0.248	0.248	16.98	0
ARC111	111	22	8533.5	0.305	0.309	**NF**	0.015
LUTZ2	89	38	15	0.111	0.176	**NF**	0.587
WEE-MAG	75	60	40.5	0.3	0.5	**NF**	0.667
BARTHOL2	148	41	124.5	0.214	0.278	**NF**	0.297
SCHOLL	297	41	2079	0.245	0.285	**NF**	0.165
				#OPT: **17/25**		**213.52**	**0.082**

Table A.6: Results with **Model-1** for $|\widetilde{V}^1| = \lceil \frac{3n}{4} \rceil$ and $|\widehat{W}| = \lceil \frac{m}{4} \rceil$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0	0	0.02	0
BOWMAN8	8	4	25.5	0.214	0.214	0.01	0
MANSOOR	11	4	67.5	0.985	0.985	0.01	0
JAESCHKE	9	5	9	0	0	0.00	0
JACKSON	11	6	10.5	0.167	0.167	0.02	0
MITCHELL	21	7	19.5	0.3	0.3	0.07	0
ROSZIEG	25	8	19.5	0.219	0.219	0.07	0
HESKIA	28	8	162	0.514	0.514	0.03	0
LUTZ1	32	9	2100	0.5	0.5	0.01	0
BUXEY	29	11	37.5	0.5	0.5	0.18	0
SAWYER30	30	11	37.5	0.442	0.442	4.99	0
GUNTHER	35	10	60	0.364	0.364	2.50	0
HAHN	53	7	2662.5	0.467	0.467	0.13	0
KILBRID	45	9	82.5	0.5	0.5	0.23	0
TONGE70	70	18	234	0.376	0.376	248.84	0
WARNECKE	58	24	79.5	0.395	0.516	**NF**	0.308
ARC83	83	17	5536.5	0.438	0.466	**NF**	0.063
LUTZ3	89	18	111	0.321	0.321	11.88	0
BARTHOLD	148	12	574.5	0.5	0.5	2.37	0
MUKHERJE	94	20	256.5	0.425	0.433	**NF**	0.019
ARC111	111	22	8533.5	0.609	0.632	**NF**	0.037
LUTZ2	89	38	15	0.154	0.403	**NF**	1.619
WEE-MAG	75	60	40.5	0.558	0.558	6.17	0
BARTHOL2	148	41	124.5	0.353	0.496	**NF**	0.404
SCHOLL	297	41	2079	0.426	0.55	**NF**	0.292
				#OPT:	**18/25**	**179.1**	**0.110**

Table A.7: Results with **Model-1** for $|\widetilde{V}^1| = \lceil \frac{n}{4} \rceil$ and $|\widehat{W}| = \lceil \frac{m}{2} \rceil$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0	0	0.03	0
BOWMAN8	8	4	25.5	0.214	0.214	0.01	0
MANSOOR	11	4	67.5	0.534	0.534	0.02	0
JAESCHKE	9	5	9	0	0	0.01	0
JACKSON	11	6	10.5	0.167	0.167	0.06	0
MITCHELL	21	7	19.5	0.219	0.219	0.11	0
ROSZIEG	25	8	19.5	0.219	0.219	0.04	0
HESKIA	28	8	162	0.409	0.409	1.23	0
LUTZ1	32	9	2100	0.483	0.483	0.52	0
BUXEY	29	11	37.5	0.293	0.293	2.53	0
SAWYER30	30	11	37.5	0.339	0.339	4.03	0
GUNTHER	35	10	60	0.304	0.304	5.60	0
HAHN	53	7	2662.5	0.307	0.307	0.14	0
KILBRID	45	9	82.5	0.5	0.5	1.04	0
TONGE70	70	18	234	0.279	0.279	476.40	0
WARNECKE	58	24	79.5	0.347	0.42	**NF**	0.208
ARC83	83	17	5536.5	0.293	0.321	**NF**	0.094
LUTZ3	89	18	111	0.233	0.233	9.23	0
BARTHOLD	148	12	574.5	0.426	0.436	**NF**	0.025
MUKHERJE	94	20	256.5	0.336	0.356	**NF**	0.059
ARC111	111	22	8533.5	0.427	0.446	**NF**	0.043
LUTZ2	89	38	15	0.091	0.274	**NF**	2.018
WEE-MAG	75	60	40.5	0.5	0.5	13.16	0
BARTHOL2	148	41	124.5	0.284	0.383	**NF**	0.351
SCHOLL	297	41	2079	0.309	0.418	**NF**	0.35
				#OPT: **17/25**		**212.57**	**0.126**

Table A.8: Results with **Model-1** for $|\widetilde{V}^1| = \lceil \frac{n}{2} \rceil$ and $|\widehat{W}| = \lceil \frac{m}{2} \rceil$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0	0	0.03	0
BOWMAN8	8	4	25.5	0.206	0.206	0.00	0
MANSOOR	11	4	67.5	0.5	0.5	0.01	0
JAESCHKE	9	5	9	0	0	0.00	0
JACKSON	11	6	10.5	0.167	0.167	0.01	0
MITCHELL	21	7	19.5	0.219	0.219	0.13	0
ROSZIEG	25	8	19.5	0.219	0.219	0.12	0
HESKIA	28	8	162	0.29	0.29	2.94	0
LUTZ1	32	9	2100	0.314	0.314	0.88	0
BUXEY	29	11	37.5	0.25	0.25	2.95	0
SAWYER30	30	11	37.5	0.25	0.25	4.80	0
GUNTHER	35	10	60	0.25	0.25	3.48	0
HAHN	53	7	2662.5	0.207	0.207	0.23	0
KILBRID	45	9	82.5	0.384	0.393	**NF**	0.023
TONGE70	70	18	234	0.238	0.238	146.56	0
WARNECKE	58	24	79.5	0.242	0.298	**NF**	0.23
ARC83	83	17	5536.5	0.246	0.262	**NF**	0.067
LUTZ3	89	18	111	0.207	0.207	23.88	0
BARTHOLD	148	12	574.5	0.312	0.313	**NF**	0.003
MUKHERJE	94	20	256.5	0.233	0.246	**NF**	0.057
ARC111	111	22	8533.5	0.294	0.309	**NF**	0.051
LUTZ2	89	38	15	0.071	0.213	**NF**	1.983
WEE-MAG	75	60	40.5	0.3	0.5	**NF**	0.667
BARTHOL2	148	41	124.5	0.209	0.278	**NF**	0.332
SCHOLL	297	41	2079	0.239	0.285	**NF**	0.191
				#OPT:	**15/25**	**247.44**	**0.144**

Table A.9: Results with **Model-1** for $|\widetilde{V}^1| = \lceil \frac{3n}{4} \rceil$ and $|\widehat{W}| = \lceil \frac{m}{2} \rceil$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0.8	0.8	0.02	0
BOWMAN8	8	4	25.5	0.417	0.417	0.00	0
MANSOOR	11	4	67.5	$+\infty$	$+\infty$	0.01	0
JAESCHKE	9	5	9	0.125	0.125	0.00	0
JACKSON	11	6	10.5	1.1	1.1	0.01	0
MITCHELL	21	7	19.5	1.786	1.786	0.12	0
ROSZIEG	25	8	19.5	0.625	0.625	0.26	0
HESKIA	28	8	162	5.231	5.231	0.95	0
LUTZ1	32	9	2100	1.298	1.298	1.46	0
BUXEY	29	11	37.5	0.875	0.875	2.96	0
SAWYER30	30	11	37.5	2.125	2.125	3.17	0
GUNTHER	35	10	60	0.5	0.5	0.07	0
HAHN	53	7	2662.5	1.742	1.742	0.06	0
KILBRID	45	9	82.5	2.75	2.75	0.78	0
TONGE70	70	18	234	1.294	1.491	**NF**	0.152
WARNECKE	58	24	79.5	1.409	1.409	223.67	0
ARC83	83	17	5536.5	1.636	1.636	16.85	0
LUTZ3	89	18	111	0.85	0.85	5.70	0
BARTHOLD	148	12	574.5	2.636	2.715	**NF**	0.03
MUKHERJE	94	20	256.5	2.018	2.018	213.47	0
ARC111	111	22	8533.5	2.444	2.635	**NF**	0.078
LUTZ2	89	38	15	0.875	0.875	485.63	0
WEE-MAG	75	60	40.5	0.761	1.211	**NF**	0.592
BARTHOL2	148	41	124.5	1.075	1.733	**NF**	0.612
SCHOLL	297	41	2079	1.49	2.139	**NF**	0.436
				#OPT: **19/25**		**182.21**	**0.076**

Table A.10: Results with **Model-1** for $|\widetilde{V}^1| = 0$ and $|\widehat{W}| = \lceil \frac{m}{4} \rceil$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0.5	0.5	0.02	0
BOWMAN8	8	4	25.5	0.214	0.214	0.00	0
MANSOOR	11	4	67.5	0.985	0.985	0.01	0
JAESCHKE	9	5	9	0	0	0.00	0
JACKSON	11	6	10.5	0.312	0.312	0.02	0
MITCHELL	21	7	19.5	0.5	0.5	0.05	0
ROSZIEG	25	8	19.5	0.219	0.219	0.12	0
HESKIA	28	8	162	0.723	0.723	0.20	0
LUTZ1	32	9	2100	0.643	0.643	0.44	0
BUXEY	29	11	37.5	0.5	0.5	0.62	0
SAWYER30	30	11	37.5	0.562	0.562	4.03	0
GUNTHER	35	10	60	0.429	0.429	0.57	0
HAHN	53	7	2662.5	0.714	0.714	0.27	0
KILBRID	45	9	82.5	0.793	0.839	**NF**	0.057
TONGE70	70	18	234	0.472	0.5	**NF**	0.06
WARNECKE	58	24	79.5	0.472	0.594	**NF**	0.258
ARC83	83	17	5536.5	0.5	0.5	129.16	0
LUTZ3	89	18	111	0.5	0.5	15.82	0
BARTHOLD	148	12	574.5	0.57	0.575	**NF**	0.01
MUKHERJE	94	20	256.5	0.491	0.521	**NF**	0.06
ARC111	111	22	8533.5	0.623	0.651	**NF**	0.045
LUTZ2	89	38	15	0.25	0.45	**NF**	0.802
WEE-MAG	75	60	40.5	0.761	0.761	378.62	0
BARTHOL2	148	41	124.5	0.368	0.496	**NF**	0.346
SCHOLL	297	41	2079	0.483	0.555	**NF**	0.15
				#OPT: **16/25**		**237.20**	**0.072**

Table A.11: Results with **Model-1** for $|\widetilde{V}^1| = 0$ and $|\widehat{W}| = \lceil \frac{m}{2} \rceil$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0	0	0.03	0
BOWMAN8	8	4	25.5	0.159	0.159	0.00	0
MANSOOR	11	4	67.5	0.607	0.607	0.01	0
JAESCHKE	9	5	9	0	0	0.00	0
JACKSON	11	6	10.5	0.167	0.167	0.02	0
MITCHELL	21	7	19.5	0.219	0.219	0.07	0
ROSZIEG	25	8	19.5	0.219	0.219	0.19	0
HESKIA	28	8	162	0.385	0.385	1.50	0
LUTZ1	32	9	2100	0.404	0.404	0.78	0
BUXEY	29	11	37.5	0.25	0.25	1.80	0
SAWYER30	30	11	37.5	0.293	0.293	4.03	0
GUNTHER	35	10	60	0.25	0.25	3.80	0
HAHN	53	7	2662.5	0.253	0.253	0.13	0
KILBRID	45	9	82.5	0.473	0.489	**NF**	0.034
TONGE70	70	18	234	0.265	0.265	511.36	0
WARNECKE	58	24	79.5	0.242	0.291	**NF**	0.203
ARC83	83	17	5536.5	0.295	0.314	**NF**	0.065
LUTZ3	89	18	111	0.233	0.233	7.49	0
BARTHOLD	148	12	574.5	0.318	0.322	**NF**	0.013
MUKHERJE	94	20	256.5	0.282	0.298	**NF**	0.057
ARC111	111	22	8533.5	0.337	0.346	**NF**	0.027
LUTZ2	89	38	15	0.154	0.232	**NF**	0.508
WEE-MAG	75	60	40.5	0.306	0.558	**NF**	0.82
BARTHOL2	148	41	124.5	0.233	0.29	**NF**	0.246
SCHOLL	297	41	2079	0.284	0.319	**NF**	0.122
				#OPT:	**15/25**	**261.25**	**0.084**

Table A.12: Results with **Model-1** for $|\widetilde{V}^1| = 0$ and $|\widehat{W}| = \lceil \frac{3m}{2} \rceil$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0	0	0.02	0
BOWMAN8	8	4	25.5	0.159	0.159	0.00	0
MANSOOR	11	4	67.5	0.406	0.406	0.01	0
JAESCHKE	9	5	9	0	0	0.00	0
JACKSON	11	6	10.5	0.167	0.167	0.01	0
MITCHELL	21	7	19.5	0.219	0.219	0.06	0
ROSZIEG	25	8	19.5	0.219	0.219	0.07	0
HESKIA	28	8	162	0.256	0.256	5.01	0
LUTZ1	32	9	2100	0.282	0.282	0.64	0
BUXEY	29	11	37.5	0.172	0.172	4.30	0
SAWYER30	30	11	37.5	0.21	0.21	6.58	0
GUNTHER	35	10	60	0.2	0.2	3.85	0
HAHN	53	7	2662.5	0.14	0.14	0.14	0
KILBRID	45	9	82.5	0.331	0.345	**NF**	0.044
TONGE70	70	18	234	0.194	0.2	**NF**	0.032
WARNECKE	58	24	79.5	0.187	0.205	**NF**	0.096
ARC83	83	17	5536.5	0.221	0.229	**NF**	0.034
LUTZ3	89	18	111	0.194	0.194	28.16	0
BARTHOLD	148	12	574.5	0.222	0.224	**NF**	0.006
MUKHERJE	94	20	256.5	0.161	0.166	**NF**	0.035
ARC111	111	22	8533.5	0.243	0.248	**NF**	0.02
LUTZ2	89	38	15	0.071	0.154	**NF**	1.154
WEE-MAG	75	60	40.5	0.157	0.5	**NF**	2.182
BARTHOL2	148	41	124.5	0.153	0.206	**NF**	0.346
SCHOLL	297	41	2079	0.2	0.224	**NF**	0.121
				#OPT:	**14/25**	**265.95**	**0.163**

Table A.13: Results with **Model-1** for $|\widetilde{V}^1| = 0$ and $|\widehat{W}| = m$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0.8	0.8	0.02	0
BOWMAN8	8	4	25.5	0.417	0.417	0.00	0
MANSOOR	11	4	67.5	$+\infty$	$+\infty$	0.00	0
JAESCHKE	9	5	9	0.125	0.125	0.00	0
JACKSON	11	6	10.5	1.1	1.1	0.01	0
MITCHELL	21	7	19.5	1.786	1.786	0.01	0
ROSZIEG	25	8	19.5	0.625	0.625	0.01	0
HESKIA	28	8	162	5.231	5.231	0.29	0
LUTZ1	32	9	2100	1.298	1.298	0.16	0
BUXEY	29	11	37.5	0.875	0.875	0.24	0
SAWYER30	30	11	37.5	2.125	2.125	0.28	0
GUNTHER	35	10	60	0.5	0.5	0.02	0
HAHN	53	7	2662.5	1.742	1.742	0.02	0
KILBRID	45	9	82.5	2.75	2.75	0.07	0
TONGE70	70	18	234	1.294	1.294	7.84	0
WARNECKE	58	24	79.5	1.409	1.409	7.91	0
ARC83	83	17	5536.5	1.636	1.636	2.99	0
LUTZ3	89	18	111	0.85	0.85	1.00	0
BARTHOLD	148	12	574.5	2.683	2.683	29.91	0
MUKHERJE	94	20	256.5	2.018	2.018	2.56	0
ARC111	111	22	8533.5	2.452	2.532	**NF**	0.033
LUTZ2	89	38	15	0.875	0.875	6.10	0
WEE-MAG	75	60	40.5	0.761	0.761	2.39	0
BARTHOL2	148	41	124.5	1.394	1.649	**NF**	0.183
SCHOLL	297	41	2079	1.84	2.136	**NF**	0.161
				#OPT:	**22/25**	**74.47**	**0.015**

Table A.14: Results with **Model-2** for $|\widetilde{V}^1| = 0$ and $|\widehat{W}| = \lceil \frac{m}{4} \rceil$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0.5	0.5	0.02	0
BOWMAN8	8	4	25.5	0.214	0.214	0.01	0
MANSOOR	11	4	67.5	0.985	0.985	0.00	0
JAESCHKE	9	5	9	0	0	0.00	0
JACKSON	11	6	10.5	0.312	0.312	0.01	0
MITCHELL	21	7	19.5	0.5	0.5	0.01	0
ROSZIEG	25	8	19.5	0.219	0.219	0.01	0
HESKIA	28	8	162	0.723	0.723	0.06	0
LUTZ1	32	9	2100	0.643	0.643	0.06	0
BUXEY	29	11	37.5	0.5	0.5	0.04	0
SAWYER30	30	11	37.5	0.562	0.562	0.14	0
GUNTHER	35	10	60	0.429	0.429	0.03	0
HAHN	53	7	2662.5	0.714	0.714	0.07	0
KILBRID	45	9	82.5	0.793	0.793	0.25	0
TONGE70	70	18	234	0.49	0.49	22.23	0
WARNECKE	58	24	79.5	0.5	0.529	**NF**	0.058
ARC83	83	17	5536.5	0.5	0.5	10.26	0
LUTZ3	89	18	111	0.5	0.5	2.25	0
BARTHOLD	148	12	574.5	0.574	0.574	4.30	0
MUKHERJE	94	20	256.5	0.5	0.5	5.70	0
ARC111	111	22	8533.5	0.632	0.636	**NF**	0.007
LUTZ2	89	38	15	0.25	0.25	47.82	0
WEE-MAG	75	60	40.5	0.761	0.761	7.43	0
BARTHOL2	148	41	124.5	0.399	0.482	**NF**	0.209
SCHOLL	297	41	2079	0.528	0.554	**NF**	0.05
				#OPT:	**21/25**	**100.03**	**0.013**

Table A.15: Results with **Model-2** for $|\widetilde{V}^1| = 0$ and $|\widehat{W}| = \lceil \frac{m}{2} \rceil$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0	0	0.03	0
BOWMAN8	8	4	25.5	0.159	0.159	0.01	0
MANSOOR	11	4	67.5	0.607	0.607	0.02	0
JAESCHKE	9	5	9	0	0	0.01	0
JACKSON	11	6	10.5	0.167	0.167	0.02	0
MITCHELL	21	7	19.5	0.219	0.219	0.07	0
ROSZIEG	25	8	19.5	0.219	0.219	0.04	0
HESKIA	28	8	162	0.385	0.385	0.17	0
LUTZ1	32	9	2100	0.404	0.404	0.13	0
BUXEY	29	11	37.5	0.25	0.25	0.04	0
SAWYER30	30	11	37.5	0.293	0.293	0.20	0
GUNTHER	35	10	60	0.25	0.25	0.21	0
HAHN	53	7	2662.5	0.253	0.253	0.03	0
KILBRID	45	9	82.5	0.473	0.473	0.07	0
TONGE70	70	18	234	0.265	0.265	26.29	0
WARNECKE	58	24	79.5	0.242	0.262	**NF**	0.081
ARC83	83	17	5536.5	0.301	0.305	**NF**	0.016
LUTZ3	89	18	111	0.233	0.233	0.72	0
BARTHOLD	148	12	574.5	0.321	0.321	6.07	0
MUKHERJE	94	20	256.5	0.289	0.289	6.09	0
ARC111	111	22	8533.5	0.344	0.346	**NF**	0.007
LUTZ2	89	38	15	0.154	0.154	4.31	0
WEE-MAG	75	60	40.5	0.306	0.306	22.89	0
BARTHOL2	148	41	124.5	0.258	0.284	**NF**	0.101
SCHOLL	297	41	2079	0.299	0.318	**NF**	0.066
				#OPT: **20/25**		**122.70**	**0.011**

Table A.16: Results with **Model-2** for $|\widetilde{V}^1| = 0$ and $|\widehat{W}| = \lceil \frac{3m}{2} \rceil$

Instance	n	m	T	LB	UB	CPU	GAP
MERTENS	7	4	9	0	0	0.03	0
BOWMAN8	8	4	25.5	0.159	0.159	0.01	0
MANSOOR	11	4	67.5	0.406	0.406	0.02	0
JAESCHKE	9	5	9	0	0	0.01	0
JACKSON	11	6	10.5	0.167	0.167	0.01	0
MITCHELL	21	7	19.5	0.219	0.219	0.04	0
ROSZIEG	25	8	19.5	0.219	0.219	0.07	0
HESKIA	28	8	162	0.256	0.256	0.40	0
LUTZ1	32	9	2100	0.282	0.282	0.19	0
BUXEY	29	11	37.5	0.172	0.172	0.54	0
SAWYER30	30	11	37.5	0.21	0.21	0.40	0
GUNTHER	35	10	60	0.2	0.2	0.36	0
HAHN	53	7	2662.5	0.14	0.14	0.03	0
KILBRID	45	9	82.5	0.331	0.331	0.06	0
TONGE70	70	18	234	0.194	0.194	20.64	0
WARNECKE	58	24	79.5	0.205	0.205	46.17	0
ARC83	83	17	5536.5	0.223	0.228	**NF**	0.024
LUTZ3	89	18	111	0.194	0.194	2.38	0
BARTHOLD	148	12	574.5	0.222	0.222	2.67	0
MUKHERJE	94	20	256.5	0.161	0.166	**NF**	0.033
ARC111	111	22	8533.5	0.24	0.248	**NF**	0.033
LUTZ2	89	38	15	0.071	0.071	9.99	0
WEE-MAG	75	60	40.5	0.157	0.157	19.97	0
BARTHOL2	148	41	124.5	0.186	0.197	**NF**	0.061
SCHOLL	297	41	2079	0.21	0.224	**NF**	0.064
				#OPT: **20/25**		**124.16**	**0.009**

Table A.17: Results with **Model-2** for $|\widetilde{V}^1| = 0$ and $|\widehat{W}| = m$