Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 233–238

Mail sorting optimization: a case study of the French postal company La Poste

Emmanuelle Amann *,** Evgeny Gurevsky **
Arnaud Laurent ** Nasser Mebarki **

* *La Poste, Nantes, France*
** *LS2N, Université de Nantes, France*

Abstract: This paper addresses a mail sorting optimization problem encountered by a major French mail delivery provider La Poste. To simplify the operators' handling during the sorting process, it is necessary to balance the mail flow between the outputs of each sorting machine. This problem can be considered as a variant of the well-known assembly line balancing problem (ALBP) with a particular structure of precedence constraints. At first, to handle efficiently small- and medium-size academic instances of this problem, we propose a mixed-integer linear programming (MILP) model aiming to minimize the difference between the most and least loaded outputs. Finally, to deal successfully with real industrial large-size instances, we develop an efficient heuristic, inspired by simulated annealing, which focuses on minimizing the non-linear standard deviation between the output loads. Both approaches have demonstrated a very good overall performance for the respective instance categories. Computational results are reported as well.

Keywords: Mail sorting, balancing, smoothness, ALBP, MILP, heuristic, simulated annealing, industrial case.

1. INTRODUCTION

All over the world, incumbent mail delivery operators are facing a drastic decrease in the volume of mail exchanged. In France, La Poste, a public mail distribution operator, has seen the volume of mail fall from 18 billion in 2008 to 9 billion in 2018, with forecasts of 5 billion in 2025. To cope with this problem, La Poste intends to set up a new organization of mail sorting in order to optimize the use of its sorting machines. The first element of this new organization, discussed in this paper, concerns the operators' handling during the sorting process.

Currently, to facilitate postman rounds, the set of mail assigned to them is sorted in the order of geographical location of mailboxes and stored in containers. As the mails arrive without being ordered beforehand, sorting is carried out in two steps. During the first sorting step, the containers can be filled with mail from different rounds, while at the end of the second step, each container corresponds to a single round. At present, this sorting process is done with a static view based on the day of the week and not on the actual mail flow. This leads to unbalanced containers on the first step, with some full of mail and others almost empty. To simplify the operators' handling, it would be of great interest to obtain the load between containers as smoothly as possible at the end of the first step while satisfying the mail precedence constraints, linked to the geographical location of the mailboxes. This problem can be considered as a variant of the well-known assembly line balancing problem (ALBP) with a fixed number of workstations. This latter generally consists of distributing a given set of tasks (resp. mail batches in our case) to a given set of workstations (resp. containers) while satisfying some technological restrictions and optimizing certain production objectives. Among the restrictions used, one can notice task precedence (resp. mail precedence) and cycle time constraints. The first constraints are usually represented as a directed acyclic weakly connected graph, in which the nodes correspond to the tasks and the arcs express their partial execution order. The cycle time constraints are not mandatory. They determine a certain limit on the workload of each workstation that can not be exceeded. Concerning the production objectives, we can mention the following ones: the minimization of the workload of the busiest workstation, the maximization of the workload of the least busy workstation, or the optimization of certain smoothness indexes (SIs) between the workstation workloads. For more details, we refer the reader to the three well-known ALBP reviews as Boysen et al. (2007), Battaïa and Dolgui (2013), and Boysen et al. (2022).

There are two main differences between the ALBP and the problem studied in this article. The first difference consists in a particular structure of the mail precedence constraints, which are represented as a set of independent chains of the strict delivery order of the mail batches. Each chain corresponds to a certain postman round. The second one concerns the study of two new SIs: the difference between the most and least charged container loads, denoted f_1, and the standard deviation between the container loads, denoted f_2. In addition, to the best of our knowledge, ALBP variants have never been studied in the context of mail sorting.

The remainder of this paper is organized as follows. Section 2 presents a short literature review of ALBPs with a fixed number of workstations and related to the study

10.1016/j.ifacol.2023.10.1574

of SI. Section 3 describes some details of La Poste's mail sorting. Section 4 provides a mixed-integer linear programming (MILP) model for the problem addressed at La Poste that aims to minimize the linear objective f_1. Section 5 outlines an efficient heuristic, inspired by simulated annealing, which focuses on minimizing the non-linear objective f_2 for the same problem. Finally, Section 6 reports and analyzes the experimental results of the MILP model and the heuristic, performed on small and large size instances of La Poste.

2. RELATED WORKS

To the best of our knowledge, Rachamadugu and Talbot (1991) were historically the first authors introducing and studying several smoothness indexes (SIs) such as workload variance (WV) and mean absolute deviation (MAD) to be minimized for ALBPs. Here, WV is expressed as the average squared deviation of the real workloads from the *ideal* one, which is computed as the sum of the processing time of all tasks divided by the number of workstations. As concerns MAD, it is similar to WV, where the squared values are replaced by the absolute ones. Despite discussing these two SIs, the authors proposed a two-stage heuristic method only for a single MAD objective. For the same objective, Kim et al. (1998) proposed an evolutionary algorithm. Kim et al. (1996) was the first paper proposing a solution procedure (a genetic algorithm) for an ALBP with the WV objective. Pinnoi and Wilhelm (1997) develop a branch-and-cut procedure for an ALBP aiming to maximize the minimal workstation workload. Azizoğlu and İmat (2018) developed the first exact solution approach (task-oriented branch-and-bound) for an ALBP with a quadratic smoothness index aiming to minimize the sum of squared workloads.

It is also important to mention that the considered SIs have also been studied in multi-objective frameworks for ALBPs. For example, Nearchou (2008) and Otto and Scholl (2011) investigate SIs in a weighed sum of several objectives. Pastor (2011) and Eswaramoorthi et al. (2012) incorporate them as secondary objectives in the lexicographic approach. Finally, Nearchou (2011) explores one of SIs as a primary objective aimed at finding a set of non-dominated solutions in the Pareto sense.

It is relevant to note that the SIs have also been addressed in the context of scheduling with parallel machines/processors (see, *e.g.*, Ho et al., 2009; Ouazene et al., 2014).

3. LA POSTE MAIL SORTING

As mentioned in Section 1, mail sorting at La Poste is a two-step process, as mail is not handled ahead of time. To avoid the large disparity in output loads at the end of the first step and, therefore, to simplify operators' handling, it is important to conduct a preliminary analysis of all involved postman rounds (hereinafter rounds) before starting the sorting process. This analysis can help, among others, to determine the outputs for each mail batch (hereinafter batch) in advance such that the load of each output becomes as equal as possible. This latter represents the optimization problem handled in this paper.

At La Poste, each round is considered as a set of so-called distribution points (DPs), which are strictly ordered between them with respect to a geographical location of mailboxes. The number of DPs per round can be different from one round to another. Each DP is strictly associated with a particular unique address. However, the number of mails affiliated with this address can be substantial, constituting what is known as a batch of mail attached to that DP. The total thickness of a batch is also referred to as the batch volume, which can sometimes be equal to zero in the case where the address concerned has no mail to receive. In what follows, we refer to a round as an ordered set of batches of non-zero volume.

In order to better understand the distribution of batches during the first step of the sorting process, the table format is used, where the rows represent the rounds and the columns illustrate the potential outputs of the sorting machine. Table 1 shows an example of this initial distribution for a set of batches consisting of four rounds of different size on the five-output sorting machine. For this example, each batch is represented by its non-zero volume, occupies a specific cell of the corresponding row and satisfies precedence constraints with respect to other batches in the same round. Finally, the last row of the table combines for each machine output its load corresponding to the sum of the volume of batch assigned to it. Thus, for example, the third batch of the first round has a volume of 3, occupies the third cell of the first row and has to be delivered after the second batch having a volume of 7 and before the fourth batch having a volume of 8. According to this illustrative example, the sorting machine attributes the first batch of each round to the first machine output, the second batch of each round to the second machine output, etc. Consequently, this leads to the situation where the output loads are unbalanced, which is confirmed by the value of their standard deviation $f_2 = 12.13$. As a result, the transfer of the containers (with batches) corresponding to the machine outputs by the operators to the second sorting step will not be considered as an ergonomic operation because of this load imbalance. However, a more smoothed distribution of batches to outputs is possible.

Round	Out. 1	Out. 2	Out. 3	Out. 4	Out. 5
1	10	7	3	8	4
2	5	6	1	7	
3	8	9	3		
4	11	4	3		
$\sum$	34	26	10	15	4

Table 1. An example of allocation of batches to outputs with $f_2 = 12.13$

Indeed, it is sufficient to make a few horizontal moves of some batches to empty cells while respecting precedence constraints by round in order to lead to a more suitable solution, as shown in Table 2. The latter observation constitutes the main optimization challenge of this paper, which is addressed by the MILP formulation and the heuristic method, respectively, in the next two sections.

Round	Out. 1	Out. 2	Out. 3	Out. 4	Out. 5
1	10	7	3	8	4
2	5	6	1		7
3	8		9		3
4	11	4	3		
$\sum$	34	17	16	8	14

Table 2. The same example as in Table 1 with another allocation of batches to outputs with $f_2 = 9.71$

4. MILP FORMULATION

In this section, for the studied problem, we present a MILP formulation, which aims at minimizing the linear objective f_1. We first introduce some useful notations, then describe the decision variables used and finally explain the MILP model with the necessary statements for the objective function as well as for any constraint.

Notations:

- $R = \{1, \ldots, |R|)\}$ is the set of all rounds;
- $B^{(r)} = \{1, \ldots, |B^{(r)}|)\}$ is the set of batches for the round $r \in R$;
- $v_j^{(r)}$ is the volume of the j-th batch in the round $r \in R$. Here, $j \in B^{(r)}$.
- $O = \{1, 2, \ldots, |O|\}$ is the set of outputs for the first mail sorting step;
- $O_j^{(r)} = \{j, j+1, \ldots, |O| - |B^{(r)}| + j\}$ is the interval of potential outputs for the j-th batch in the round $r \in R$;
- $U_k^{(r)} = \{j \in B^{(r)} : k \in O_j^{(r)}\}$ is the set of mail bathes of round r, which can be potentially assigned to the output $k \in O$.

Variables:

- $x_{jk}^{(r)}$ is equal to 1 if the j-th batch of round r is assigned to the output $k \in O_j^{(r)}$, 0 otherwise;
- $L_{\min} \geq 0$ is the minimal load per output;
- $L_{\max} \geq 0$ is the maximal load per output.

Model:

$$\min \quad f_1 := L_{\max} - L_{\min} \tag{1}$$

subject to:

$$\sum_{k \in O_j^{(r)}} x_{jk}^{(r)} = 1, \quad \forall j \in B^{(r)}, \quad \forall r \in R \tag{2}$$

$$\sum_{j \in U_k^{(r)}} x_{jk}^{(r)} \leq 1, \quad \forall k \in O, \quad \forall r \in R \tag{3}$$

$$\sum_{k \in O_j^{(r)}} k \cdot x_{jk}^{(r)} \leq \sum_{k \in O_{j+1}^{(r)}} k \cdot x_{j+1,k}^{(r)}, \quad \forall j \in B^{(r)} \setminus |B^{(r)}|, \quad \forall r \in R \tag{4}$$

$$L_{\min} \leq \sum_{r \in R} \sum_{j \in U_k^{(r)}} v_j^{(r)} \cdot x_{jk}^{(r)} \leq L_{\max}, \quad \forall k \in O \tag{5}$$

$$x_{jk}^{(r)} \in \{0, 1\}, \quad \forall j \in B^{(r)}, \quad \forall k \in O_j^{(r)}, \quad \forall r \in R$$

$$L_{\min}, \; L_{\max} \geq 0$$

The principal goal of objective function (1) is to find a solution minimizing the difference between the most and least loaded outputs. Constraints (2) require that each batch j of round r be assigned to exactly one output. Inequalities (3) show that, for a given round, there can be at most one batch per output. The precedence mail constraints for each round r are modeled by (4). Constraints (5) express the total load for each output, which can not be less than $L_{\min}$ and greater than $L_{\max}$.

5. HEURISTIC APPROACH

In order to handle the actual postal traffic, La Poste wishes to perform the mail sorting process several times a day. Moreover, the MILP model, presented in the previous section, quickly finds its limits when the size of the instances to be handled increases. Therefore, a fast and powerful tool is needed to efficiently address the optimization problem that appears in the first sorting step. To this end, we implement a heuristic, inspired by the simulated annealing method, whose objective is to minimize the second smoothing index f_2. Its formal description is provided by Algorithm 1, named MIN-SMOOTH. The heuristic is composed of 3 stages during which it tries to improve the objective function, based on a technique of *admissible moves* for each round, formally depicted by Algorithm 2 and called as GREEDY-MOVE. For each round, this technique considers its batches in the non-increasing order of their corresponding output loads and moves a batch located at the currently busiest output to the least busy available one of the same round. This move has of course to respect the precedence constraints of the corresponding round. If the move is not possible, then the current batch is excluded from examination and the next one is considered. In its first stage, the heuristic is only interested in admissible moves that strictly improve the value of f_2 of the current solution. This stage ends when these moves are no longer possible. For the second stage, the heuristic accepts admissible moves that can deteriorate the quality of the current solution within a tolerance threshold that decreases each time when the value of f_2 is no longer improved. The third stage starts with the solution obtained at the end of the second stage and behaves like the first stage.

To illustrate how the heuristic works, the following two tables are presented. Table 3 provides the situation obtained at the end of the first stage of MIN-SMOOTH. As concerns Table 4, it shows the best solution achieved at the end of the third stage.

Round	Out. 1	Out. 2	Out. 3	Out. 4	Out. 5
1	10	7	3	8	4
2		5	6	1	7
3		8	9	3	
4	11			4	3
$\sum$	21	20	18	16	14

Table 3. Solution at the end of Stage 1 with f_2=2.86.

To simplify the description of Algorithm 2, the following notations are used:

Round	Out. 1	Out. 2	Out. 3	Out. 4	Out. 5
1	10	7	3	8	4
2		5	6	1	7
3			8	9	3
4	11	4			3
$\sum$	21	16	17	18	17

Table 4. Best solution obtained with f_2=1.92.

Algorithm 1 Heuristic MIN-SMOOTH

Input: A current solution s, a tolerance multiplier α and a decrement step Δ.

Output: The best known solution $s^{(B)}$.

- **Stage 1**.
  - 1.1. Set tolerance value $\tau := 0$ and $s^{(B)} := s$.
  - 1.2. Let s^* be a solution provided by GREEDY-MOVE(s, τ). If $f(s^*) < f(s)$, then reset $s := s^*$ and repeat Step 1.2.
- **Stage 2**.
  - 2.1. Set tolerance value $\tau := \alpha \cdot f(s)$.
  - 2.2. Let s^* be a solution provided by GREEDY-MOVE(s, τ). If $f(s^*) < f(s)$, then reset $s := s^*$ and repeat Step 2.2.
  - 2.3. Decrease the tolerance value $\tau := \tau - \Delta$ and reset $s := s^*$. If $\tau > 0$, then go to Step 2.2.
- **Stage 3**.
  - 3.1. Set tolerance value $\tau := 0$.
  - 3.2. Let s^* be a solution provided by GREEDY-MOVE(s, τ). If $f(s^*) < f(s)$, then reset $s := s^*$ and repeat Step 3.2. Otherwise, stop and return the best found solution $s^{(B)}$.

- $O^{(r)}(s)$ is the set of non-empty outputs in round r of solution s, where each of which has at least one direct right or left empty output. For the example presented in Table 2, $O^{(2)}(s) = \{3, 5\}$.
- $E^{(r,p)}(s)$ is the set of direct left and right empty outputs of the non-empty output p in round r of solution s. For the example presented in Table 2, $E^{(3,3)}(s) = \{2, 4\}$, $E^{(4,3)}(s) = \{4, 5\}$, $E^{(3,1)}(s) = \{2\}$ and $E^{(2,2)}(s) = \emptyset$.

6. NUMERICAL EXPERIMENTS

In this section, a description of the instances used and the input data for the numerical experiments is first given. The computational results obtained for the MILP formulation and the heuristic are then presented, compared and analyzed.

The MILP model was implemented using the solver CPLEX 12.6 and the heuristic was coded using Python language. The experiments were conducted on a computer having a 2.80 GHz Intel® Core™ i7-1165G7 processor with 16 GB RAM.

The heuristic method was examined on two categories of instances, named *academic* and *industrial*, respectively. As concerns the MILP model, it was investigated only on *academic* instances. The academic instances are randomly generated and consist of 12 series of 30 instances each. Their size varies from 12 to 120 rounds and from 20 to 90 sorting outputs. For the real life industrial instances, only 8 of them were provided by La Poste, where the number of rounds ranges from 112 to 228, and the number of sorting outputs is between 76 and 216.

A time limit of 600 seconds per instance was fixed for solving the MILP model by the CPLEX solver. For the heuristic parameters, the tolerance multiplier α was set to 0.05 and the decrement step Δ was established to 0.02.

The computational results for the academic instances are presented in Table 5. Each row of this table corresponds to a particular series of instances. Columns 2-6 are related to the MILP model and columns 7-9 concern the heuristic. This first column indicates the fixed parameters for each series, *i.e.*, the number of rounds and the used number of sorting outputs, respectively. The second column reports the number of instances solved to optimality by the MILP model for each series. The third column shows the average `GAP` per series only for the instances, for which no optimal solution was found. The fourth column specifies the average `CPU` time per series only for the optimally solved instances. The fifth and sixth columns indicate the average value per series for the best solutions found by the MILP model with respect to the smoothness indexes f_1 and f_2, respectively. Columns 7-9 are the same as for the MILP model, but concern the heuristic instead.

By analyzing the results obtained in Table 5, we can first observe that the MILP model achieves to optimally solve 71.67% of all academic instances which represents a remarkable performance. Moreover, for the first 7 series, almost every academic instance was optimally solved in 34.35 seconds on average. It is also important to notice that the MILP model has demonstrated better performance than the heuristic for each of these first 7 series in terms of the average value for both smoothing indices among the best found solutions. Thus, MILP provides a solution with an average value of f_1 equal to 9.86 (which corresponds to one mail) against 16.05 for the heuristic. This excellent performance of the MILP model is maintained until the 9th series. Starting from the 10th series, no optimal solution was found by the MILP model.

As concerns the heuristic, it has produced outstanding results for all academic instances. Thus, each instance was processed in less than a second on average and with an average value of f_1 of 12. Moreover, the heuristic clearly outperforms the MILP model from the 8th series for both smoothing indices.

This outstanding heuristic performance continues when it comes to industrial instances, see Table 6. This table deals with instances from 10,000 to 40,000 mails to be sorted. Thus, the heuristic succeeds in processing each instance in less than 45 seconds on average and provides a mean smoothing value of f_1 equal to 28.75. The latter corresponds to barely 3 mails.

The heuristic's promising performance is confirmed by Figure 1, where the convergence of the f_2 value at the end of each stage is demonstrated on three industrial instances.

7. CONCLUSION AND PERSPECTIVES

In this paper, a new industrial optimization problem was introduced. This latter is related to one of the important steps of the mail sorting process at the main French

Algorithm 2 Procedure Greedy-Move

Input: An initial solution $s^{(0)}$ and a tolerance value τ.
Output: A potentially new solution better than $s^{(0)}$.

1. Set $i := 0$ and $r := 1$.
2. Compute $O^{(r)}(s^{(i)})$ and set $P := O^{(r)}(s^{(i)})$.
3. If $P = \emptyset$, then go to Step 5. Otherwise, choose the most loaded output p from P and the least loaded output q from $E^{(r,p)}(s)$.
4. Let $s^{(T)}$ be a solution obtained from solution $s^{(i)}$ by shifting the batch from output p to output q for round r. If $f(s^{(T)}) < f(s^{(B)})$, then update the best known solution, *i.e.*, set $s^{(B)} := s^{(T)}$. If $f(s^{(T)}) < f(s^{(i)}) + \tau$, then move to a new current solution, *i.e.*, set $i := i + 1$ and $s^{(i)} := s^{(T)}$. Update $P := P \setminus \{p\}$ and go to Step 3.
5. If $r < |R|$, then set $r := r + 1$ and go to Step 2. Otherwise, stop and return solution $s^{(i)}$.

Table 5. Computational results for the MILP model and the heuristic on academic instances

Instances	MILP with f_1					Min-Smooth with f_2		
	#OPTI	Avg. GAP, (%)	Avg. CPU, (s.)	Avg. f_1	Avg. f_2	Avg. CPU, (s.)	Avg. f_1	Avg. f_2
12x20	30	⊕	3.34	**9.67**	**3.97**	0.02	22.33	6.42
20x20	30	⊕	3.09	**10**	**4.08**	0.05	16	5.26
30x15	30	⊕	0.91	**9.33**	**3.81**	0.04	22	5.96
30x30	30	⊕	16.36	**10**	**4.10**	0.10	13	4.59
40x25	30	⊕	7.61	**9.67**	**3.80**	0.12	15	4.26
40x40	29	50	97.51	**10.33**	**3.94**	0.29	13.33	4.27
50x40	30	⊕	113.76	**10**	**3.91**	0.50	10.67	4
60x50	25	50	334.69	11.67	4.89	0.79	**10**	**3.97**
75x50	24	87.5	427.26	12	5.10	0.97	**10.67**	**4.26**
80x55	0	100	⊖	34.33	12.18	1.64	**9.67**	**4.06**
100x60	0	100	⊖	179.33	53.83	1.64	**10**	**4.13**
120x90	0	100	⊖	1508.67	503.79	5.31	**12**	**3.84**

(⊕) All optimal solutions were found within the time limit of 600 seconds.
(⊖) No optimal solution was found within the time limit of 600 seconds.

Instances	CPU, (s.)	Value f_1	Value f_2
112x76	1.62	30.00	10.00
111x217	35.35	40.00	10.97
204x127	8.68	10.00	5.01
208x202	48.12	30.00	6.65
210x196	56.91	20.00	6.49
213x214	64.29	30.00	9.48
226x202	47.96	30.00	7.41
228x216	89.97	40.00	13.21

Table 6. Computational results for Min-Smooth with f_2 on industrial instances

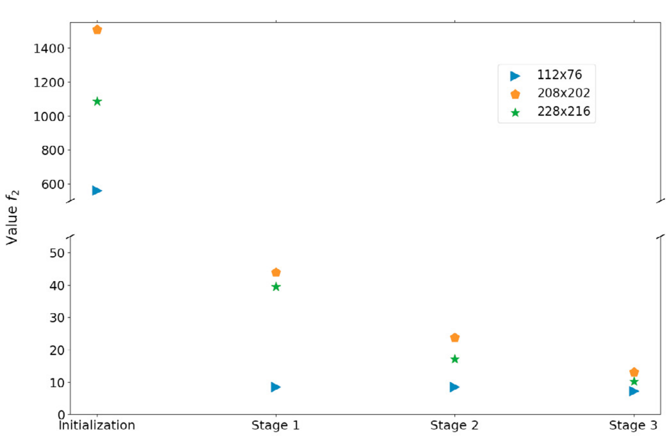

Fig. 1. Illustration of the performance of each heuristic stage for 3 industrial instances

mail delivery provider La Poste. The problem deals with the search for a smooth distribution of the mail batches between all the involved outputs of the sorting machines used. It was at first shown that such a problem can be considered as a particular variant of the well-known assembly line balancing problem of type 2, but with a specific structure of precedence constraints.

A MILP formulation and an appropriate heuristic, inspired by simulated annealing, were proposed and implemented to efficiently solve this problem. Two smoothing indices were considered as objective functions. Two categories of instances (academic and industrial ones) were used to validate the proposed approaches.

Both approaches have demonstrated good overall performance, especially the heuristic on industrial instances, thanks to which it is now successfully used in several sorting centers of La Poste and is being deployed throughout France.

For our future research work, some new ideas can be applied to further improve the performance of the heuristic method.

ACKNOWLEDGEMENTS

The authors would like to thank Nicolas Compère and Adrien Pichon, two M.S. students in Computer Science at the University of Nantes, for pointing out some errors in the realization of the original MILP model and for providing us with correct numerical results for it.

REFERENCES

Azizoğlu, M. and İmat, S. (2018). Workload smoothing in simple assembly line balancing. *Computers & Operations Research*, 89, 51–57.

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

Boysen, N., Fliedner, M., and Scholl, A. (2007). A classification of assembly line balancing problems. *European Journal of Operational Research*, 183(2), 674–693.

Boysen, N., Schulze, P., and Scholl, A. (2022). Assembly line balancing: What happened in the last fifteen years? *European Journal of Operational Research*, 301(3), 797–814.

Eswaramoorthi, M., Kathiresan, G.R., Jayasudhan, T.J., Prasad, P.S.S., and Mohanrama, P.V. (2012). Flow index based line balancing: a tool to improve the leanness of assembly line design. *International Journal of Production Research*, 50(12), 3345–3358.

Ho, J.C., Tseng, T.L., Ruiz-Torres, A.J., and López, F.J. (2009). Minimizing the normalized sum of square for workload deviations on m parallel processors. *Computers & Industrial Engineering*, 56(1), 186–192.

Kim, Y.J., Kim, Y.K., and Cho, Y. (1998). A heuristic-based genetic algorithm for workload smoothing in assembly lines. *Computers & Operations Research*, 25(2), 99–111.

Kim, Y.K., Kim, Y.J., and Kim, Y. (1996). Genetic algorithms for assembly line balancing with various objectives. *Computers & Industrial Engineering*, 30(3), 397–409.

Nearchou, A.C. (2008). Multi-objective balancing of assembly lines by population heuristics. *International Journal of Production Research*, 46(8), 2275–2297.

Nearchou, A.C. (2011). Maximizing production rate and workload smoothing in assembly lines using particle swarm optimization. *International Journal of Production Economics*, 129(2), 242–250.

Otto, A. and Scholl, A. (2011). Incorporating ergonomic risks into assembly line balancing. *European Journal of Operational Research*, 212(2), 277–286.

Ouazene, Y., Yalaoui, F., Chehade, H., and Yalaoui, A. (2014). Workload balancing in identical parallel machine scheduling using a mathematical programming method. *International Journal of Computational Intelligence Systems*, 7($1^{\mathtt{bis}}$), 58–67.

Pastor, R. (2011). LB-ALBP: the lexicographic bottleneck assembly line balancing problem. *International Journal of Production Research*, 49(8), 2425–2442.

Pinnoi, A. and Wilhelm, W.E. (1997). A branch and cut approach for workload smoothing on assembly lines. *INFORMS Journal on Computing*, 9(4), 335–350.

Rachamadugu, R. and Talbot, B. (1991). Improving the equality of workload assignments in assembly lines. *International Journal of Production Research*, 29(3), 619–633.