



HAL
open science

Simple Assembly Line Balancing Problem with Power Peak Minimization

Paolo Gianessi, Xavier Delorme, Oussama Masmoudi

► **To cite this version:**

Paolo Gianessi, Xavier Delorme, Oussama Masmoudi. Simple Assembly Line Balancing Problem with Power Peak Minimization. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2019, Austin, TX, United States. pp.239-247, 10.1007/978-3-030-30000-5_31 . hal-02274953

HAL Id: hal-02274953

<https://hal-utt.archives-ouvertes.fr/hal-02274953>

Submitted on 24 Jan 2020

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

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



Distributed under a Creative Commons Attribution| 4.0 International License

Simple Assembly Line Balancing Problem with Power Peak Minimization

Paolo Gianessi[✉]¹[0000–0001–5457–3689], Xavier Delorme¹[0000–0002–9465–1255],
and Oussama Masmoudi²[0000–0002–0100–536X]

¹ Mines Saint-Étienne, Univ.Clermont Auvergne, CNRS, UMR 6158 LIMOS
Institut Henri Fayol, F-42023 Saint-Étienne France

`paolo.gianessi@emse.fr, delorme@emse.fr`

² ICD, LOSI (UMR-CNRS 6281), University of Technology of Troyes, Troyes, France
`oussama.masmoudi@utt.fr`

Abstract. The increased environmental awareness of these days urge companies towards energy efficient production. This work focuses on the consideration of power consumption since the design stage of a production system. A new Assembly Line Balancing Problem is introduced in which each task is associated with a power consumption. The aim is to find an assignment of tasks to workstations that minimizes the overall power peak while complying with given maximum cycle time and number of workstations. An Integer Linear Programming formulation is proposed and tested on benchmark instances enhanced with power features. Numerical results are discussed.

Keywords: Simple Assembly Line Balancing · Power Peak Minimization · Integer Linear Programming

1 Introduction

According to the International Energy Agency, in 2014 the industrial sector was responsible for 36% of the global total final energy consumption (TFEC), and since then its energy consumption has grown by 1.5% annually [1]. Hence, it is expected that in the future the most consistent reduction of greenhouse gas (GHG) emissions will come from an improved energy efficiency of industries [2]. Moreover, this period represents a transition phase for the industrial sector. The forth industrial revolution or *Industry 4.0* will respond to economical, technological, organizational, societal and environmental issues. Among the objectives due to the environmental concern is the reduction of carbon footprint by optimizing the energy consumption. Therefore, it is important to rise awareness of the necessity to control the related consumption of manufacturers, since the power demand and energy consumption are steadily increasing.

In order to address this problem, one of the existing solutions is to reorganize the energetic consumption in order to obtain a lower power peak. The growing attention paid in the last decades to energy-aware manufacturing and production systems has led to an increased scientific effort to design decision support

methods capable of achieving an optimized energy management. However, the number of works dealing with energy-driven optimization of scheduling or line balancing in manufacturing systems was still relatively scarce only a few years ago, as pointed out by [3].

In this paper, a new Assembly Line Balancing Problem with power peak minimization is considered. The rest of the paper is organized as follows. Section 2 introduces the literature review. Sections 3 and 4 present the new ALB problem and an integer linear program for it. Section 5 analyses the results of numerical tests. Finally, Section 6 presents the conclusion and future research perspectives.

2 Literature Review

Assembly line balancing (ALB) problems are among the most investigated optimization problems arising in production systems. ALB problems aim to optimally assign the elementary *tasks* involved in the assembly of a series of products to a set of *workstations* of a production line while complying with *precedence constraints* among tasks. These problems occur during the design of a production system and determine some of its main features, e.g. *cycle time* and number of workstations. The simplest and most studied variant is the *Simple ALB Problem* (SALBP) [4], in which the line is paced and synchronous, task process times are deterministic and independent of workstations and only one product is considered. In the case of *type I SALBP* (SALBP-1), the number of workstations m has to be minimized, given the cycle time c ; the minimum c for a given value of m is sought for in *type II SALBP* (SALBP-2); finally, SALBP-E (efficiency) aims at minimizing the product $c \cdot m$, or equivalently the total idle time, while SALBP-F (feasibility) tries to find out whether given values for both c and m admit a feasible solution. Despite its simple statement, SALBP is NP-hard and remains a challenging problem [5] and best known solution approaches are relatively recent [6,7,8]. A huge literature exists concerning ALB problem variants issued from several different application contexts. For more in-depth reading on ALB problems, the reader is referred to the excellent literature review of [9]. *Assembly line design* problems are also worth mentioning. In them, some special equipment must be assigned to workstations to enable them to perform tasks, as it is the case with robots in the *Robotic ALB* (RALB) [10].

For what concerns ALB problems with energy-driven optimization criteria, as far as the authors are aware of, few works can be found to date. In [11], a two-sided RALB is proposed, along with a Mixed-Integer Linear Programming (MILP) formulation. A simulated annealing-based metaheuristic algorithm is yielded to seek for Pareto-optimal solution w.r.t energy consumption and cycle time. In [12], two evolutionary algorithms, namely a Particle Swarm Optimization (PSO) and a Differential Evolution algorithm, are used to minimize the energy consumption of a U-shaped robotic assembly line. A RALB variant minimizing the total energy consumption, a Nonlinear Programming formulation and a PSO-based approach are the main contributions of [13]. The benchmark instances for type II RALB of [14] are used to test the effectiveness of the proposed approach.

This short literature review shows that few works exist that consider energy on a strategic level, i.e. since the design phase, in a production system, and in those works the concerned aspect is the minimization of energy consumption of equipment associated with workstations. The authors are not aware of works that tackle the power peak involved in the processing of the production tasks at a strategic level. However, power peak minimization or a power peak constraint can be found in works dealing with other production-related optimization problems at a more tactical or operational level, especially in scheduling.

In flow-shop scheduling, [15] proposes a mixed-integer linear program (MILP) to minimize energy consumption, makespan and carbon footprint, while [16] resort to a heuristic to minimize total tardiness and makespan of a flexible flow-shop, both under a power peak constraint. The scheduling of a job-shop with makespan minimization and power peak is addressed by [17] and tackled by means of a MILP, while [18] introduces a power peak constraint in a production system with parallel machines and proposes a heuristic method. Finally, [19] tries to minimize energy cost in a job-shop under makespan and power peak constraints. In this work, a new ALB problem is addressed which aims at minimizing the overall power consumption peak and does not seem to have yet received attention in the literature. An Integer Linear Program is proposed and tested.

3 Problem Definition

In the following the *Simple Assembly Line Balancing Problem with Power Peak Minimization* (SALB3PM) is introduced, along with the related notations and an example to help the reader understand the key aspects of power peak minimization.

The classic SALBP consists in assigning the tasks of a set \mathcal{O} to the workstations of a set \mathcal{M} of a paced, synchronous production line. Precedence constraints are defined that force each task $j \in \mathcal{O}$ to wait the end of all its direct predecessors before its processing can start: the notation $i \prec j$ indicates that i is such a direct predecessor. Each task j features a processing time t_j which is constant and independent of workstations. The goal is usually to optimize the cycle time or the number of workstations, the other being given, in such a way that precedence constraints are fulfilled, tasks are not preempted and each workstation processes at most one task at a time all along the time horizon. It is worth noting that the classic SALBP disregards scheduling aspects, because once tasks are optimally assigned to workstations, they can be scheduled at any date along the cycle time. This does not change the solution value, nor it affects its feasibility, as long as precedence constraints among tasks on the same workstation are complied with. In the SALB3PM, similarly to what is done in the SALBP-F, both the number of available workstations $m = |\mathcal{M}|$ and the maximum allowed cycle time c are given. Moreover, each task j features a power consumption W_j , constant and independent of workstations. The goal becomes to find a feasible assignment of tasks to workstations s.t. the highest peak of the overall power consumption profile is minimized. This objective requires to integrate scheduling decisions in the

SALB3PM, thus asking for dedicated decision variables. Most of all, the strong connection of this further decision layer with the other classical decisions make the SALB3PM harder than the SALBP.

Figure 1 illustrates the impact of power peak minimization by comparing two feasible solutions for a SALB3PM instance with cycle time $c = 18$, $m = 3$ workstations and $|\mathcal{O}| = 9$ tasks. The leftmost one is a generic solution that displays a very fluctuating overall power consumption profile. The rightmost one is an optimal solution, i.e. in which the maximum value of global power consumption over the time horizon is minimized: the overall consumption profile is much more smoothed and a power peak reduction of 36.6% is achieved.

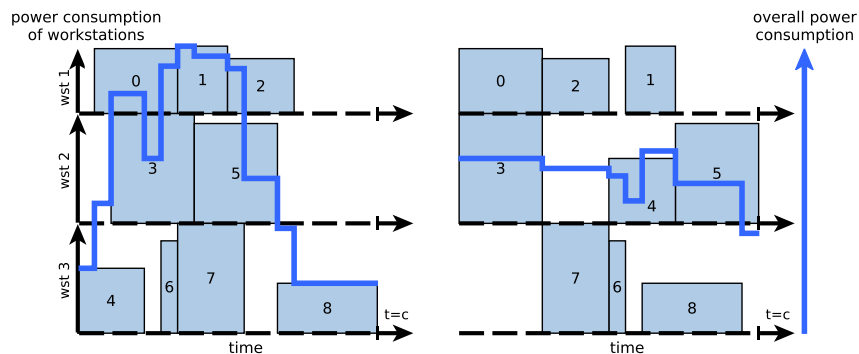


Fig. 1. A generic (left) and an optimal (right) solution of a SALB3PM instance with $c = 18$, $m = 3$, $|\mathcal{O}| = 9$. Dashed lines separate the schedule of workstations. Numbered boxes represent tasks: their width, height and position stand for associated processing times, power consumption values and starting times. A thick curve depicts the evolution of the overall power consumption profile.

4 An Integer Linear Program for the SALB3PM

In this section an Integer Linear Programming (ILP) model for the SALB3PM is presented. The model is *time-indexed*, i.e. its main feature is that the time horizon is subdivided in unit time slots and the variables describing the scheduling of tasks have a time index. This allows a detailed representation of the processing of tasks and hence of the cumulative power profile all along the time horizon. Let $\mathcal{T} = \{0, \dots, c - 1\}$ and $\mathcal{T}^i = \{0, \dots, c - t_i\}$ denote respectively the entire set of time slots, and the available time slots for the start of task i . The model has one integer nonnegative variable and two sets of binary variables:

- ▶ *power peak* variable W_{\max} , an upper bound on the power consumption peak;
- ▶ *assign* variables $X_{i,k}$, $i \in \mathcal{O}$, $k \in \mathcal{M}$, $X_{i,k} = 1 \Leftrightarrow$ task i assigned to workstation k ;
- ▶ *trigger* variables $S_{i,t}$, $i \in \mathcal{O}$, $t \in \mathcal{T}^i$, $S_{i,t} = 1 \Leftrightarrow$ processing of task i starts at time slot t of each cycle.

The proposed model is the following:

$$\min W_{\max} \quad (1)$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{M}} X_{j,k} = 1 \quad \forall j \in \mathcal{O} \quad (2)$$

$$\sum_{j \in \mathcal{O}} t_j \cdot X_{j,k} \leq c \quad \forall k \in \mathcal{M} \quad (3)$$

$$X_{j,k} \leq \sum_{h \in \mathcal{M}: h \leq k} X_{i,h} \quad \forall i, j \in \mathcal{O}: i \prec j, k \in \mathcal{M} \quad (4)$$

$$\sum_{t \in \mathcal{T}^j} S_{j,t} = 1 \quad \forall j \in \mathcal{O} \quad (5)$$

$$S_{j,t} \leq \sum_{\tau=0}^{t-t_i} S_{i,\tau} + 2 - X_{i,k} - X_{j,k} \quad \forall i, j \in \mathcal{O}: i \prec j, k \in \mathcal{M}, t \in \mathcal{T}^j \quad (6)$$

$$X_{i,k} + X_{j,k} + \sum_{\tau=t-t_i+1}^t S_{i,\tau} + \sum_{\tau=t-t_j+1}^t S_{j,\tau} \leq 3 \quad \forall i, j \in \mathcal{O}, k \in \mathcal{M}, t \in \mathcal{T} \quad (7)$$

$$\sum_{j \in \mathcal{O}} W_j \cdot \left(\sum_{\tau=t-t_j+1}^t S_{j,\tau} \right) \leq W_{\max} \quad \forall t \in \mathcal{T} \quad (8)$$

$$X_{i,k}, S_{i,t} \in \{0, 1\}, W_{\max} \in \mathbb{Z}_+$$

Constraints (2)-(4) are classical SALBP constraints. Each task j is assigned to exactly one workstation k by (2), and each $i \in \mathcal{O}$ s.t. $i \prec j$ is assigned to k or one of the upstream workstations by precedence constraints (4). Tasks assigned to one workstation have the sum of their processing times bounded by c due to (3). Constraints (5)-(8) are specific to SALB3PM. Each task j is processed at some date $t \in \mathcal{T}^j$ due to (5), which ensures its termination within given cycle time c . Constraints (6) add to (4) in enforcing precedence. If tasks i and j , $i \prec j$, are assigned to the same workstation, then (6) (jointly with (5)) force i to start at least t_i time slots sooner than j . Otherwise (6) are redundant as the right hand side is greater or equal than one. Relations (7) impose to have at most one task at a time running on the same workstation at date t of each cycle. Indeed, term $\sum_{\tau=t-t_i+1}^t S_{i,\tau}$ is 1 if task i has started at a date $\tau \in \{t-t_i+1, \dots, t\}$, in which case it is still running at date t . Hence, (7) state that either both i and j are running at date t , which forces $X_{i,k} + X_{j,k} \leq 1$, or both are assigned to $k \in \mathcal{M}$, i.e. $X_{i,k} + X_{j,k} = 2$ and at most one of the two can be running at date t . Finally, (8) state that for each date t the sum of the power consumption of the currently running tasks is bounded by W_{\max} . This is because the left hand side of (8) represents the overall power consumption at date t of each cycle. Along with the objective function (1), this smooths the overall power consumption profile, as shown in Figure 1.

5 Computational Experiments

In this section, the computational session conducted to assess the performance of the proposed ILP model is illustrated. The model is implemented in CPLEX 12.6 and solved by Branch&Bound (B&B). Tests are run on a Intel Core i7-5500U 2.40 Ghz machine with 15.55Gb RAM. All instances are given a 3600s time limit.

A testbed set of 19 SALB3PM instances is generated, inspired from 15 benchmark SALBP-1 datasets and 4 benchmark SALBP-2 datasets ¹ and completed with task power consumption values W_i randomly generated from the uniform distribution $U(5, 50)$. In the original datasets the number of tasks varies between 7 and 30, and either c is given and m is the computed optimal number of workstations, or m is given and c is the optimal cycle time. This guarantees the feasibility of the derived SALB3PM instances. From each SALB3PM instance, another one is obtained with cycle time augmented by 30% and rounded up, so as to stress the performances of the time-indexed model. To assess the power consumption improvement, a heuristic solution is generated by:

- ▶ solving (B&B) a reduced model (1)-(4) (i.e. removing scheduling constraints),
- ▶ sorting tasks on each workstation so as to comply with precedence constraints,
- ▶ schedule tasks at the earliest date, i.e. without idle times between them.

Table 1 reports the results for each instance. W_{\max} is the optimal power peak value, while $T/(\%)$ is the computational time (in seconds) to compute it, or the optimality gap still to close at time limit. W_{\max}^H is the value of the heuristic solution: the time to compute it is always less than 1s and thus neglected. Terms $\overline{W_{\max}^H}$, $\overline{W_{\max}}$, $\overline{T/(\%)}$ have the same meaning w.r.t longer cycle time $\bar{c} = \lceil 1.3 \cdot c \rceil$. Table 1 shows that in 23 instances out of 38 an optimal solution is achieved

Table 1. Computational results of the model alone.

| instance | $ \mathcal{O} $ | m | c | W_{\max}^H | W_{\max} | $T/(\%)$ | \bar{c} | $\overline{W_{\max}^H}$ | $\overline{W_{\max}}$ | $\overline{T/(\%)}$ |
|-------------|-----------------|-----|-----|--------------|------------|----------|-----------|-------------------------|-----------------------|---------------------|
| mertens-1 | 7 | 6 | 6 | 191 | 171 | 0.02 | 8 | 161 | 161 | 0.03 |
| mertens-2 | 7 | 2 | 18 | 77 | 57 | 0.18 | 24 | 87 | 48 | 0.27 |
| bowman-1 | 8 | 5 | 20 | 152 | 113 | 0.10 | 26 | 114 | 86 | 0.58 |
| jaeschke-1 | 9 | 8 | 6 | 249 | 249 | 0.02 | 8 | 178 | 178 | 0.08 |
| jaeschke-2 | 9 | 3 | 18 | 110 | 73 | 1.37 | 24 | 71 | 60 | 2.70 |
| jackson-1 | 11 | 8 | 7 | 217 | 188 | 0.05 | 10 | 146 | 121 | 0.52 |
| jackson-2 | 11 | 3 | 21 | 79 | 58 | 4.46 | 28 | 73 | 48 | 4.54 |
| mansoor-1 | 11 | 4 | 48 | 142 | 133 | 0.60 | 63 | 146 | 99 | 213.64 |
| mansoor-2 | 11 | 2 | 94 | 93 | 78 | 3.00 | 123 | 90 | 59 | 1305.11 |
| mittchell-1 | 21 | 8 | 14 | 241 | 211 | 11.97 | 19 | 210 | 149 | (1.36%) |
| mittchell-2 | 21 | 3 | 39 | 106 | 76 | 1383.80 | 51 | 116 | 62 | (10.71%) |
| roszieg-1 | 25 | 10 | 14 | 327 | 256 | 201.58 | 19 | 232 | 189 | (1.07%) |
| roszieg-2 | 25 | 4 | 32 | 144 | 119 | 878.39 | 42 | 147 | 88 | (3.53%) |
| heskiaoff-1 | 28 | 8 | 138 | 274 | – | – | 180 | 209 | – | – |
| heskiaoff-2 | 28 | 3 | 342 | 125 | – | – | 445 | 120 | – | – |
| buxey-1 | 29 | 14 | 25 | 468 | 395 | 3352.65 | 33 | 366 | 288 | (2.13%) |
| buxey-2 | 29 | 7 | 47 | 248 | – | – | 62 | 217 | 182 | (21.33%) |
| sawyer-1 | 30 | 14 | 25 | 389 | 326 | (1.56%) | 33 | 348 | 248 | (2.48%) |
| sawyer-2 | 30 | 7 | 47 | 202 | – | – | 62 | 191 | – | – |

within time limit. In most cases only few seconds are needed, except in six cases which require more than 200s. This is acceptable considering the strategic nature of the problem. For such instances, an average power peak gain of -19.8%

¹ <https://assembly-line-balancing.de/salbp/benchmark-data-sets-1993/>

is achieved w.r.t the heuristic solution (with $\frac{W_{\max}}{W_{\max}^H} < 1$ in 20 cases out of 23), in which scheduling decisions are greedy. In particular, the average gain is -22.5% for instances with high cycle time.

In 8 out of the remaining 15 instances, the B&B finds a feasible solution but the optimality gap is greater than 0 at time limit, even though less than 2.5% in five cases (6.31% on average). This is probably caused by the increased computation hardness, due to the growing number of tasks, and also the larger search space for instances with longer cycle time. Nevertheless, a considerable average power peak improvement of -27.1% is obtained w.r.t the heuristic solution.

In the remaining 7 instances, no feasible solution can be found. The heuristic solution can then be used as a warmstart for the B&B, as shown in Table 2.

Table 2. Computational results with warm start.

| instance | $ \mathcal{O} $ | m | c | W_{\max}^H | W_{\max} | $T/(%)$ | \bar{c} | $\overline{W_{\max}^H}$ | $\overline{W_{\max}}$ | $\overline{T/(%)}$ |
|-------------|-----------------|-----|-----|--------------|------------|----------|-----------|-------------------------|-----------------------|--------------------|
| heskiaoff-1 | 28 | 8 | 138 | 274 | 274 | (16.10%) | 180 | 209 | 209 | (15.47%) |
| heskiaoff-2 | 28 | 3 | 342 | 125 | 125 | (23.03%) | 445 | 120 | 120 | (52.40%) |
| buxey-2 | 29 | 7 | 47 | 248 | 223 | (12.63%) | 62 | 217 | - | - |
| sawyer-2 | 30 | 7 | 47 | 202 | 202 | (18.82%) | 62 | 191 | 150 | (16.28%) |

In two cases, an average power peak improvement of -15.8% is achieved. By considering the overall behavior of the $\frac{W_{\max}}{W_{\max}^H}$ ratio, the optimality gap still to close at time limit is probably mostly due to the distance between W_{\max}^H and the optimal value. This is promising in terms of further power peak gains that can be potentially obtained.

Finally, the results for the 9 instances for which an optimal solution is found with both c and \bar{c} show that the longer cycle time allows an average power peak gain of -21.6%. This suggests that from an industrial point of view it may make sense to decrease production pace as the consequences in terms of power consumption may be considerable.

6 Conclusion and Research Perspectives

In this work, a new Assembly Line Balancing (ALB) problem, the Simple Assembly Line Balancing Problem with Power Peak Minimization (SALB3PM), is studied in which the overall power consumption peak is minimized. The scheduling aspects that must be taken into account make this problem harder than classical ALB problems. An Integer Linear Program is proposed and tested on benchmark instances completed with power features. Preliminary results show that considerable gains can be achieved in terms of power consumption, and thus that SALB3PM deserves attention from researchers. In particular, it seems promising to study the extent of the interaction between balancing and scheduling decisions. This could lead to decomposition-based efficient metaheuristic approaches to hopefully deal with real-world instances.

References

1. International Energy Agency: Tracking Clean Energy Progress 2017: Industry. <https://www.iea.org/etp/tracking2017/industry/>, published: 2017-05-16
2. Lawrence, A., Thollander, P., Andrei, M., Karlsson, M.: Specific Energy Consumption/Use (SEC) in Energy Management for Improving Energy Efficiency in Industry: Meaning, Usage and Differences. *Energies* **12**(2), 247 (2019)
3. Dai, M., Tang, D., Giret, A., Salido, M.A., Li, W.D.: Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robot Cim-Int Manuf* **29**, 418–429 (2013)
4. Baybars, İ.: A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem. *Manage Sci* **32**(8), 909–932 (1986)
5. Scholl, A.: Balancing and sequencing of assembly lines. Tech. rep., Darmstadt Technical University – Institute for Business Studies (BWL) (1999)
6. Cerqueus, A., Delorme, X.: A branch-and-bound method for the bi-objective simple line assembly balancing problem. *Int J Prod Res* (2018). <https://doi.org/10.1080/00207543.2018.1539266>
7. Pape, T.: Heuristics and lower bounds for the simple assembly line balancing problem type 1: Overview, computational tests and improvements. *Eur J Oper Res* **240**, 32–42 (2015)
8. Sewell, E.C., Jacobson, S.H.: A branch, bound, and remember algorithm for the simple assembly line balancing problem. *INFORMS J Comput* **24**(3), 433–442 (2012)
9. Battaïa, O., Dolgui, A.: A taxonomy of line balancing problems and their solution approaches. *Int J Prod Econ* **142**, 259–277 (2013)
10. Borba, L., Ritt, M., Miralles, C.: Exact and heuristic methods for solving the Robotic Assembly Line Balancing Problem. *Eur J Oper Res* **270**, 146–156 (2018)
11. Li, Z., Tang, Q., Zhang, L.: Minimizing energy consumption and cycle time in two-sided robotic assembly line systems using restarted simulated annealing algorithm. *J Clean Prod* **135**, 508–522 (2016)
12. Nilakantan, M.J., Ponnambalam, S.G., Jawahar, N.: Design of energy efficient RAL system using evolutionary algorithms. *Eng Computation* **33**(2), 580–602 (2016)
13. Nilakantan, M.J., Huang, G.Q., Ponnambalam, S.G.: An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems. *J Clean Prod* **90**(2), 311–325 (2015)
14. Gao, J., Sun, L., Wang, L., Gen, M.: An efficient approach for type II robotic assembly line balancing problems. *Comput Ind Eng* **56**(3), 1065–1080 (2009)
15. Fang, K., Uhan, N., Zhao, F., Sutherland, J.W.: A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *J Manuf Syst* **30**(4), 234–240 (2011)
16. Bruzzone, A.G., Anghinolfi, D., Paolucci, M., Tonelli, F.: Energy-aware scheduling for improving manufacturing process sustainability: A mathematical model for flexible flow shops. *CIRP Ann-Manuf Techn* **61**(1), 459–462 (2012)
17. Kemmoe, S., Lamy, D., Tchernev, N.: Job-shop like manufacturing system with variable power threshold and operations with power requirements. *Int J Prod Res* **55**(20), 6011–6032 (2017)
18. Artigues, C., Lopez, P., Haït, A.: The energy scheduling problem: Industrial case-study and constraint propagation techniques. *Int J Prod Econ* **143**(1), 13–23 (2013)
19. Masmoudi, O., Delorme, X., Gianessi, P.: Job-shop scheduling problem with energy consideration. *Int J Prod Econ* **216**, 12–22 (2019)