Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Line Balancing and Sequencing for Peak Power Minimization

Damien Lamy ** **Xavier Delorme** * **Paolo Gianessi** *

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

(e-mail: {damien.lamy, delorme, paolo.gianessi}@emse.fr)

Abstract: In the past years, environmental awareness started to bring new production paradigms based on energy efficiency. If it is possible to improve energy efficiency of existing production systems, it should be even more profitable to consider this objective at the design stage. In the context of Paced Production Lines, and given power requirements for operations, it becomes possible to assign more efficiently these operations to stations while respecting other constraints such as maximum takt time and number of workstations. The repetitive nature of paced lines implies that misconceptions implying a high peak power consumption will see this peak power repeated over and over without having large possibilities to correct it. In order to tackle peak power minimization objectives, this implies to consider sequencing of operations in addition to their assignment to workstation which is not classical in line balancing. In this paper, the problem under study is presented with a new specific feature that allows to consider semi-active sequence of operations at each station. In order to address large scale instances, a first metaheuristic approach is implemented and evaluated on an extended dataset from the literature. Results show that it is possible to improve energy efficiency at the design stage of production systems.

Keywords: Line Balancing, Sequencing, Energy Efficiency, Peak Power, Metaheuristics.

1. INTRODUCTION

The industrial sector is known to be the first energy consumer and greenhouse gas emitter in the world (Wang and Li, 2013). Due to climate change, the past years have seen a transition of the industry. The forth industrial revolution, referred to as Industry 4.0, intends to answer economical, technological, organizational, societal and environmental issues industry is facing. Several objectives can be mentioned in order to improve the environmental impact of production, such as reducing carbon footprint by optimizing the energy consumption, knowing that power demand and energy consumption are steadily increasing. Improving energy efficiency of production systems can be done by several technological or organisational actions at the operational level (Duflou et al., 2012). In the literature, the latter is generally addressed through minimization of total energy consumption, time of use pricings, or peak power minimization (Giret et al., 2015). If the first two are largely addressed in the literature (Akbar and Irohara, 2018) still few research projects deal with reducing peak power consumption. This latter allows easier smoothening of energy production from the provider side and reduce operating costs (better dimensioning of power requirements, avoiding penalties for large peak power consumptions, etc.). This is even more important in the context of paced production lines where the peak power is repeated at each cycle. If several papers concerning energy-efficient scheduling are available in the literature, these research projects refer to situations related to the exploitation of the production systems. Actually, very few papers deal with such issues at the design level of production systems, in particular concerning Line Balancing (LB) problems, as stressed in Gianessi et al. (2019). The same paper defines the Simple Assembly Line Balancing Problem with Power Peak Minimization (SALB3PM), one of the first examples of line balancing problem with minimization of peak power, based on the Simple Assembly Line Balancing Problem (SALBP), a reference problem in LB. More specifically, the authors showed that in order to consider peak power minimization in the design of paced production lines, it is necessary not only to consider classical constraints, such as precedence constraints, but also the scheduling of operations on workstations, i.e. their starting times.

In this paper, a particular case of the SALB3PM is dealt with in which tasks assigned to a workstation are executed without idle times between operations, therefore making scheduling decisions actually become sequencing decisions. This version of SALB3PM better fits the case of manual or semi-automatized production systems, where human resources are involved and an earliest starting date of tasks without intermediate idle times is more appropriate. Hence, an integrated Balancing and Sequencing Problem with Peak Power Minimization is investigated here. A

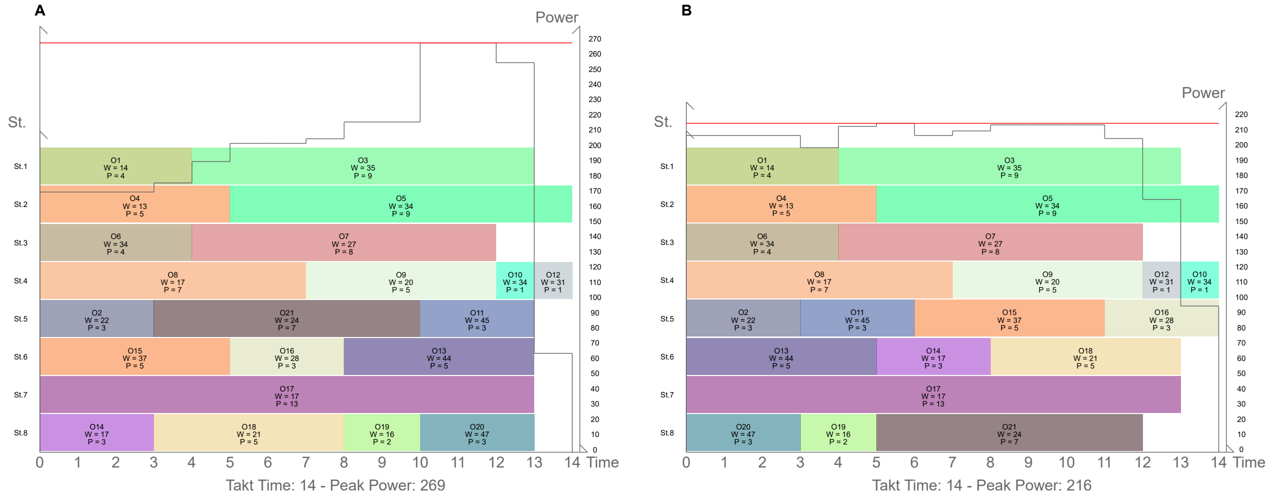

Fig. 1. Exemple of a solution presenting Gantt chart and power consumption profile

production system with given number of workstations and takt time is considered. A set of operations is also given, each featuring constant processing time and power requirement. Operations must be assigned to workstations, and their sequence must be decided, so that precedence relations are complied with and the resulting peak of the overall power profile is minimized when operations are scheduled at the earliest date.

An example of the problem at hand is given in Figure 1. In it, *St.*, *Power* and *Time* axis correspond to stations, required power to process operations and takt time. Each operation is represented with a rectangle in which O is its index, W its power requirement and P its processing time. Parts (A) and (B) represent two solutions of an instance with 8 workstations and takt time equal to 14. However, if assignments and sequencing of operations on stations 1, 2, 3 and 7 are similar, it is not the case on other stations. This results in a different power profile for the two solutions: in particular, the power peak is 269 in solution (A) and 216 in the much more smoothened power profile of solution (B). This example clearly states the necessity of integrating balancing and sequencing at the conception stage of production lines.

The rest of the paper is organized as follows. A literature review is provided in Section 2. Section 3 introduces a metaheuristic approach for problem solving. Section 4 analyses the results of numerical experiments, and Section 5 consists in the conclusion and presentation of future research prospects.

2. LITERATURE REVIEW

Line Balancing problems are among the most investigated optimization problems arising in production systems. LB problems consists in optimally assign operations required by production to a set of workstations of a production line while complying with precedence constraints among operations. These problems may occur during the design or reconfiguration of a production system and determine some of its main features, e.g. takt time and number of workstations. The simplest and most studied variant is the Simple ALB Problem (SALBP) (Baybars, 1986), in which the line is paced and synchronous, operation processing times are deterministic and independent of workstations and the line allows only one product to be realized. Mainly four different SALBP can be encountered : (i) SALBP-1, where the number of workstations m has to be minimized, given the takt time c; (ii) SALBP-2 where the minimum c for a given value of m is searched for; (iii) SALBP-E (efficiency) which aims at minimizing the total idle time, and (iv) the SALBP-F (feasibility) which tries to determine whether or not a solution exists considering given values for both c and m. Despite its simple statement, SALBP is NP-hard and remains a challenging problem (Scholl and Becker, 2006) and several best known solution approaches are relatively recent (Cerqueus and Delorme, 2019; Pape, 2015; Sewell and Jacobson, 2012). Existing literature on ALB is very large, and several extensions are considered in order to fit industrial needs and applications. The reader can refer to the literature review provided by Battaïa and Dolgui (2013) for more information on assembly line balancing problems. However, as far as the authors are aware of, few works address joint LB and sequencing, as in Andrés et al. (2008) in which this is done to take into account sequence-dependent setup times. Neither did the consideration of energy efficiency in LB problems receive a large amount of attention from the research community. Actually, most of existing papers considering energy-efficiency concern assembly line design problems, where special equipment must be assigned to workstations to enable them to perform tasks. This is the case with robots in Robotic ALB (Borba et al., 2018) since criteria on tools selection may include their energy consumption and hence it is naturally that this selection is intended to minimize total energy consumption. Li et al. (2016) considered a two-sided RALB. A Mixed-Integer Linear Programming (MILP) formulation is given and a simulated annealing-based metaheuristic algorithm is developed to search for Pareto-optimal solution considering both energy consumption and takt time as objectives. Nilakantan et al. (2016) explored the use of two evolutionary algorithms, namely a Particle Swarm Optimization (PSO) and a Differential Evolution algorithm, in order to minimize the energy consumption of a U-shaped robotic assembly line. Nilakantan et al. (2016) studied a RALB variant where the objective is to minimize the total energy consumption. Energy expenditures during idle times are also considered. A Nonlinear

Programming formulation is given and a PSO algorithm is designed to obtain solutions. The approach is evaluated on problems based on the benchmark instances provided by Gao et al. (2009). From this short literature review, it can be seen that most addressed problems concern total energy consumption rather than peak power minimization. Actually, and to the best of authors knowledge, considering energy consumptions and power requirements of operations at the design stage of production systems is not spread in the literature on ALB problems. However, this kind of constraints and/or objectives are more present in works dealing with other production-related optimization problems, especially in scheduling. Even in these problems, the total energy consumption is prevalent and few works consider either peak power as an objective or a constraint in problem formulations. Fang et al. (2011) proposed a mixed-integer linear program (MILP) to minimize energy consumption, makespan and carbon footprint in the context of a Flow-shop Scheduling Problem. Bruzzone et al. (2012) proposed a MILP and a heuristic approach to minimize peak power consumption in a Flexible Flow-shop where an initial sequence of operation is considered. This sequence of operations is not modified but starting dates of operation are changed in order to reduce energy consumption. A Job-shop with a variable power threshold is addressed by Kemmoe et al. (2017) and several metaheuristics are designed. Artigues et al. (2013) introduce the energy scheduling problem where an electric power limitation is considered. A two step integer/constraint programming approach is designed and applied on an industrial problem. Finally, Masmoudi et al. (2019) intend to minimize energy cost in Job-shop Scheduling Problem under makespan and power peak constraints. In Gianessi et al. (2019) a first mathematical formulation is introduced in which idle times can be inserted between operations on workstations in order to get the lowest peak power consumption possible. Although instances with up to 25 operations, 10 workstations and a takt time of 40 are solved to optimality within a one hour time limit, for bigger instances the proposed exact method fails either to prove optimality (with large optimality gap values at time limit) or even to find feasible solutions. This restricts its suitability for instances of industrial interest.

3. METAHEURISTIC APPROACH

As it is difficult to obtain optimal solutions, a metaheuristic approach is investigated. The applied metaheuristic is based on a Multi-start Evolutionary Local Search (MS×ELS), which is close to the algorithm proposed by Prins (2009). This metaheuristic relies on a randomized construction heuristic and on the Evolutionary Local Search (ELS, Wolf and Merz (2007)). Such metaheuristics has shown efficiency in solving several problems (Chassaing et al., 2014; Kemmoe et al., 2017). A template algorithm of the MS×ELS is proposed in Figure 2, where two phases can be highlighted: the construction phase which is applied nb_G times, and the ELS phase. At each ELS iteration (limited by nb_E), neighbours of the previous solution are generated nb_N times and improved through a local search process. The mutation consists in changing the position and/or workstation of a random number of operations. All these procedure rely on a representation of solutions which is presented in the following.

```
MS×ELS(nb_G, nb_E, nb_N)
argtype  nb_G, nb_E, nb_N: different loop sizes;
returns  S*: best found solution;
declare  S, nS, nS*: local solutions;
S* ← ∅
for g ← 1 to nb_G
    S ← CONSTRUCTION_PHASE()
    for e ← 1 to nb_E
        for n ← 1 to nb_N
            nS ← MUTATION_PHASE()
            nS ← LOCAL_SEARCH_PHASE()
            nS* ← best found nS
        next n
        S ← nS*; S* ← best found solution
    next e
next g
return S*
```

Fig. 2. MS×ELS pseudo-code.

3.1 Encoding Solution

Representation of solutions is one of the key features of metaheuristics. In the present approach, an indirect representation of solutions is considered. Such approaches are common and widely spread in the literature from scheduling to routing problems (Duhamel et al., 2011). Two vectors are used in order to represent a solution. The first one (A) is an assignment vector that allows to know which operation is assigned to which workstation (i.e. indexed on operation number).The second vector (V) is a sequence of operations. The decoding procedure assigns all operations to workstations according to A, and sequences these operations without delay following the respective order given in V. Taking the solution from Fig. 1(B) as an example, the two vectors could be: $A = \{1;5;1;2;2;3;3;4;4;4;5;4;6;6;5;5;7;6;8;8;8\}$ and $V = \{1;3;4;5;6;7;8;9;12;10;2;11;15;16;13;14;18;17;20;19;21\}$. Obviously, vectors in Fig. 1(A) are different. The decoding procedure returns the starting and ending dates of operations. As it can be stressed from such representations, vectors that do not respect precedence constraints could appear. However such situations are avoided during generation of initial solution and during Neighbourhood exploration.

3.2 Fitness Evaluation

The objective of the problem is to minimize the power consumption. However, if the number of workstations is easily managed and set at the beginning of the solving approach, some infeasible vectors can be generated because of the takt constraint. To deal with this issue, a fitness value is used to compare solutions. This fitness value is given by the required power supply to process operations, plus a penalty in case of infeasible vectors. This way the search process tries to move towards valid solutions without forbidding them. This penalty corresponds to the sum of time exceeding the takt on each station.

3.3 Construction Heuristic

In order to build initial solutions, a construction heuristic is developed. Considering the number of operations and number of stations, the procedure starts by partitioning operations into balanced groups while respecting precedence constraints. At the end of this stage, vector A is fully defined. Then, starting from station 1 to the last one, at each iteration, an operation that respects precedences is taken randomly from the remaining operations to sequence on the station. At each step, the selected operation is inserted at the end of vector V. At each stage, the starting time and ending time of the operation is computed, and the power consumption during this time window is increased by the power requirement of the operation. When the takt time on a station exceeds the constraint, the duration that exceeds the takt is added into the penalty variable.

3.4 Neighbourhood Structure

A single neighbourhood structure is considered in this paper. It is based on insertion of operations. Let S be a solution and O be an operation. Let a be the position of the closest predecessor of O in V and k its corresponding station. Let b be the position of its closest successor and h its station. (if no predecessor exists then $a = 1$ and $k = 1$; if no successor exists $b = n$ where n denotes the number of operations and $h = m$ with m the number of workstations). A neighbouring solution is obtained by removing O from its current station and by assigning it to a station in [k;h] and inserting it in the range [a;b] in vector V. For instance, in problem instance of Fig. 1(B), operation 11's closest predecessor is operation 9, and its closest successor is operation 15. Hence, operation 11 can be inserted in any position from operation 9 in workstation 4 (i.e. after O9), to operation O15 in workstation 6 (i.e. before O15). This may violate takt time but it allows to move from a solution to another (Connected Neighbourhood). This basis neighbourhood is used for diversification and intensification phases. In Mutation_Phase, a neighbouring solution of a given solution S is taken randomly in N(S). For diversification of solutions, at each step of second loop in MS×ELS, the neighbourhood is applied as many times as loop iteration number.

3.5 Local Search Procedure

In order to improve solutions a local search procedure is used. This local search consists in exploring neighbourhood of a given solution based on N. At each step of the search process, a neighbour of the current solution is obtained. This neighbour is taken randomly in N(S). If the fitness value of the new solution is better than the previous one, it becomes the current solution. The search process is given in Figure 3.

```
LocalSearch(S, N, n)
argtype S: solution; N: neighborhood;
argtype n: number of local search iterations;
returns S: neighbouring solution;
declare lS: local solution; i: int;
i ← 0
while(i < n)
   lS ← random solution in N(S)
   if(lS fitness value is better than S)
      S ← lS
      i ← 1
   else
      i ← i + 1
endwhile
return S
```

Fig. 3. Local Search pseudo-code.

4. NUMERICAL EXPERIMENTS

4.1 Benchmark

In this section, the computational experiments conducted to evaluate the performance of the proposed metaheuristic is given. We extend the instance set proposed in Gianessi et al. (2019), which is based on a dataset of 19 problems that are from 15 SALBP-1 benchmark datasets and 4 SALBP-2 benchmark datasets, by adding 4 further instances based on 4 more SALBP-2 benchmarks. Power consumption values W_i are randomly generated from the uniform distribution U(5; 50). The number of tasks varies from 7 to 45, and either c is given and m is the computed optimal number of workstations, or m is given and c is the optimal takt time. Hence, all the considered instances are feasible. From each instance, another one is obtained by considering a takt time increased by 30% and rounded up. Numerical experiments have been run on an Intel Xeon E3-1505M 3Ghz machine with 16Gb RAM. All the instances are given a maximum of 120 seconds, and 50 replications are considered. Before computing results, a design of experiments (DOE) has been conducted on a problem instance, sawyer-2, as it is one of the instance with largest operation number in the dataset and seemed hardly addressed by linear solvers. This instance is thus not included in the conducted computational sessions. The total number of considered instances is thus 44.
In this DOE, nb_G is taken in [40, 100] with a 20 steps, nb_E is taken in [50, 150] with a 25 steps, and nb_N is taken in [10, 20] with a 2 steps. This DOE resulted in following parameters: $nb_G = 80$, $nb_E = 75$, $nb_N = 14$.

4.2 Comparison with an Exact Method

In order to assess the performances of the proposed MS×ELS, we consider the mathematical formulation proposed in Gianessi et al. (2019) with an additional constraint in order to forbid idle times between any two operations on a machine. For the sake of brevity the extended model is not fully reported here. In the same work, a naive heuristic method solution is defined to provide SALB3PM which consist in solving the balancing subproblem and, in a postprocessing phase, sequencing operations on workstations to comply with precedence relations and scheduling them at the earliest possible starting date.
When solved by Branch&Bound, the extended model can find optimal or near-optimal solutions within the same time limit of one hour considered in Gianessi et al. (2019) for small-sized instances, i.e. with up to 30 operations,

Table 1. Results of MS×ELS on dataset for takt and 1.3 · takt.

instance	n	m	c	LB	$W^H_{\max}$	MS×ELS $W_{\max}$	MS×ELS %gap	MS×ELS σ	MS×ELS T(s)	c	LB	$W^H_{\max}$	MS×ELS $W_{\max}$	MS×ELS %gap	MS×ELS σ	MS×ELS T(s)
mertens-1	7	6	6	191⋆	191	191.00	0.00	0.00	0.0	8	161⋆	161	161.00	0.00	0.00	0.0
mertens-2	7	2	18	65⋆	77	65.00	0.00	0.00	0.0	24	50⋆	87	50.00	0.00	0.00	0.0
bowman-1	8	5	20	129⋆	152	129.00	0.00	0.00	0.0	26	106⋆	114	106.00	0.00	0.00	0.0
jaeschke-1	9	8	6	249⋆	249	249.00	0.00	0.00	0.0	8	178⋆	178	178.00	0.00	0.00	0.0
jaeschke-2	9	3	18	86⋆	110	86.00	0.00	0.00	0.0	24	66⋆	71	66.00	0.00	0.00	112.0
jackson-1	11	8	7	205⋆	217	205.00	0.00	0.00	0.0	10	130⋆	146	130.00	0.00	0.00	1.0
jackson-2	11	3	21	67⋆	79	67.00	0.00	0.00	0.0	28	57⋆	73	57.00	0.00	0.00	0.0
mansoor-1	11	4	48	133⋆	142	133.00	0.00	0.00	0.0	63	122⋆	146	122.00	0.00	0.00	0.0
mansoor-2	11	2	94	78⋆	93	78.00	0.00	0.00	0.0	123	75⋆	90	75.00	0.00	0.00	0.0
mitchell-1	21	8	14	216⋆	241	216.00	0.00	0.00	0.3	19	147	210	156.00	6.12	0.00	1.6
mitchell-2	21	3	39	86⋆	106	86.00	0.00	0.00	12.5	51	55	116	71.00	29.09	0.00	2.0
roszieg-1	25	10	14	265⋆	327	265.00	0.00	0.00	8.2	19	187	232	194.00	3.74	0.00	12.2
roszieg-2	25	4	32	120⋆	144	120.00	0.00	0.00	0.7	42	85	147	92.04	8.28	0.28	14.4
heskiaoff-1	28	8	138	236	274	246.48	4.44	0.70	20.2	180	181	209	188.22	3.99	1.20	33.4
heskiaoff-2	28	3	342	95	125	101.42	6.76	0.49	29.5	445	73	120	79.04	8.27	0.20	24.8
buxey-1	29	14	25	402⋆	468	406.86	1.21	4.87	39.1	33	282	366	294.66	4.49	0.76	45.3
buxey-2	29	7	47	198	248	772.18	289.99	490.95	36.2	62	150	217	157.52	5.01	0.61	31.7
sawyer-1	30	14	25	321	389	326.00	1.56	0.00	20.1	33	242	348	250.38	3.46	0.82	42.6
gunther-1	35	14	40	421⋆	438	421.00	0.00	0.00	0.9	52	289	374	300.60	4.01	0.87	37.8
gunther-2	35	7	72	209	240	223.18	6.78	0.62	37.2	94	160	247	167.04	4.40	0.80	34.6
kilbridge-1	45	11	55	272	362	283.76	4.32	1.07	53.7	72	208	280	219.34	5.45	0.84	39.7
kilbridge-2	45	7	79	190	247	200.28	5.41	0.98	48.7	103	146	191	155.94	6.81	0.99	43.3
mean							14.57	22.71	14.0					4.23	0.34	16.6

15 workstations and a takt time of up to 30 time units. However, for bigger instances, the model fails either to prove optimality (with large optimality gap values at time limit) or even to find feasible solutions.
This motivates the proposal of a metaheuristic approach.

4.3 Results Analysis

Table 1 reports the results for each instance. In it, LB represents a lower bound on power peak obtained by running Branch&Bound on the extended formulation mentioned in Section 4.2: a star ($\star$) denotes if such lower bound is the value of the optimal solution, or it is the best value of the dual bound at time limit. Terms $W_{\max}$, %gap, σ and T(s) are average values of, respectively, power peak, gap from lower bound, standard deviations of solutions and computation time required to reach the best found solution on each replication. Finally, $W^H_{\max}$denotes the value of the solution of the heuristic approach of Gianessi et al. (2019).
In all the cases for which LB is the optimal solution value, MS×ELS always find the optimal solution (%gap=0) in all the replications (σ=0), with the only exception of buxey-1 with basic takt time value ($c = 25$), for which the average gap is 1.21% with small differences among replications.
In the cases when the Branch&Bound is not able to provide an optimal solution, the difference between the optimal power peak value and LB is greater than 0: such difference is incorporated in %gap, which partly explains its higher values. In these cases, when the takt time is at its base value (left part of the table), the average computation time is under 15 seconds, while the average gap between $W_{\max}$and LB is under 15%, but drops under 1.5% if we do not consider instance buxey-2. Indeed, this instance is clearly not well addressed by the current MS×ELS, as stressed by standard deviation σ, and most of all by the fact of being the only instance for which MS×ELSbehaves worse than the naive heuristic of Gianessi et al. (2019), which is otherwise always largely outperformed. For this instance, $W_{\max}$ is valued 772 over the 50 replications, the optimum being not greater than 248: this means that the constraint related to the takt has been violated and penalized in the computation of fitness.
In instances for which LB is not the optimal value and the takt time is increased by 30% (rightmost part of the table), the values of %gap are on average higher, which is expected due to the enlarged solution space; however, computation times are still very short (on average under 17s), while values of σapproach 0, which shows that the metaheuristic can find solutions that are closer. Also, $W_{\max}$ values are lowered, showing that the power consumption of production systems can be substantially improved. To this respect, it is worth mentioning that the augmented takt time is not necessarily fully exploited, e.g. considering bowman-1 instance, $W_{\max}$ is 129 with takt time bounded to 20, and 106 (i.e. a reduction of more than 17%) when the takt time is bounded to 26, but the solution has a takt time of 22 and one unused station. This means that important reduction in power consumption can be obtained even for a low takt increase.

5. CONCLUSIONS

In this research project an extension/particular case of the Single Assembly Line Balancing Problem with Peak Power Minimization is studied in which operations are processed without delay on workstations. In order to cope with the Peak Power objective, sequencing decisions are considered in addition with the balancing decisions. As this problem is harder than classical Simple Assembly Line Balancing Problem, a metaheuristic approach based on a MS×ELS is explored. Results show that the approach is effective in finding valuable solutions (i.e. close or equal to the solutions returned by CPLEX solver), even though further computational experiments are required in order to properly tune the proposed MS×ELSand avoid some unstable behaviour. To this end, the naive heuristic approach of Gianessi et al. (2019) could be used to provide initial solutions for the MS×ELS so as to prevent it from returning solutions violating the takt constraint.
However, some instances remain difficult to solve. Hence, in future works, the different procedures will be enhanced by considering other neighbourhood structures and guided

local searches, taking more advantage of structure of solutions. Currently, semi-active sequences are considered, having all operations set to the left. An extension of the problem could be to consider either operations set to the left on some stations, or set to the right, keeping no-idle times constraint between operations. It could also be interesting to consider reconfigurable manufacturing systems and provide solutions to move from one configuration to another when considering peak power constraints and/or time-of-use objective. For instance, such problems would require the fewest modifications possible from one configuration to another while respecting a power threshold.

Another interesting possibility is to address the bi-objective optimisation problem in order to provide pareto of possible solutions and measure the distance from a solution to another not only from the viewpoint of peak power consumption but also in terms of number of required modifications to go from a solution to another (i.e. minimize number of operations that are changed from one station to another). Such an approach could be used in reconfigurable manufacturing systems in order to know how to manage modules and machine tools when peak shaving are considered.

REFERENCES

Akbar, M. and Irohara, T. (2018). Scheduling for sustainable manufacturing: A review. J. of Cleaner Production, 205, 866–883.

Andrés, C., Miralles, C., and Pastor, R. (2008). Balancing and Scheduling tasks in assembly lines with sequence-dependent setup times. European J. of Operational Research, 187, 1212–1223.

Artigues, C., Lopez, P., and Haït, A. (2013). The energy scheduling problem: Industrial case-study and constraint propagation techniques. Int. J. of Production Economics, 143(1), 13–23.

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

Baybars, İ. (1986). A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem. Management Science, 32(8), 909–932.

Borba, L., Ritt, M., and Miralles, C. (2018). Exact and heuristic methods for solving the Robotic Assembly Line Balancing Problem. European J. of Operational Research, 270(1), 146–156.

Bruzzone, A., Anghinolfi, D., Paolucci, M., and Tonelli, F. (2012). Energy-aware scheduling for improving manufacturing process sustainability: A mathematical model for flexible flow shops. CIRP Annals - Manufacturing Technology, 61(1), 459–462.

Cerqueus, A. and Delorme, X. (2019). A branch-and-bound method for the bi-objective simple line assembly balancing problem. Int. J. of Production Research, 57(18), 5640–5659.

Chassaing, M., Fontanel, J., Lacomme, P., Ren, L., Tchernev, N., and Villechenon, P. (2014). A GRASP×ELS approach for the job-shop with a web service paradigm packaging. Expert Systems with Applications, 41(2), 544–562.

Duflou, J.R., Sutherland, J.W., Dornfeld, D., Herrmann, C., Jeswiet, J., Kara, S., Hauschild, M., and Kellens, K. (2012). Towards energy and resource efficient manufacturing: A processes and systems approach. CIRP Annals - Manufacturing Technology, 61(2), 587–609.

Duhamel, C., Lacomme, P., and Prodhon, C. (2011). Efficient frameworks for greedy split and new depth first search split procedures for routing problems. Computers & Operations Research, 38(4), 723–739.

Fang, K., Uhan, N., Zhao, F., and Sutherland, J.W. (2011). A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. J. of Manufacturing Systems, 30(4), 234–240.

Gao, J., Sun, L., Wang, L., and Gen, M. (2009). An efficient approach for type II robotic assembly line balancing problems. Computers & Industrial Engineering, 56(3), 1065–1080.

Gianessi, P., Delorme, X., and Masmoudi, O. (2019). Simple Assembly Line Balancing Problem with Power Peak Minimization. In F. Ameri, K.E. Stecke, G. von Cieminski, and D. Kiritsis (eds.), Advances in Production Management Systems, volume 566, 239–247. Springer Int. Publishing, Cham.

Giret, A., Trentesaux, D., and Prabhu, V. (2015). Sustainability in manufacturing operations scheduling: A state of the art review. J. of Manufacturing Systems, 37, 126–140.

Kemmoe, S., Lamy, D., and Tchernev, N. (2017). Job-shop like manufacturing system with variable power threshold and operations with power requirements. Int. J. of Production Research, 55(20), 6011–6032.

Li, Z., Tang, Q., and Zhang, L. (2016). Minimizing energy consumption and cycle time in two-sided robotic assembly line systems using restarted simulated annealing algorithm. J. of Cleaner Production, 135, 508–522.

Masmoudi, O., Delorme, X., and Gianessi, P. (2019). Job-shop scheduling problem with energy consideration. Int. J. of Production Economics, 216, 12–22.

Nilakantan, M.J., Ponnambalam, S.G., and Jawahar, N. (2016). Design of energy efficient RAL system using evolutionary algorithms. Engineering Computations, 33(2), 580–602.

Pape, T. (2015). Heuristics and lower bounds for the simple assembly line balancing problem type 1: Overview, computational tests and improvements. European J. of Operational Research, 240(1), 32–42.

Prins, C. (2009). A GRASP x evolutionary local search hybrid for the vehicle routing problem. In Bio-inspired algorithms for the vehicle routing problem, 35–53. Springer.

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

Sewell, E.C. and Jacobson, S.H. (2012). A Branch, Bound, and Remember Algorithm for the Simple Assembly Line Balancing Problem. INFORMS Journal on Computing, 24(3), 433–442.

Wang, Y. and Li, L. (2013). Time-of-use based electricity demand response for sustainable manufacturing systems. Energy, 63, 233–244.

Wolf, S. and Merz, P. (2007). Evolutionary Local Search for the Super-Peer Selection Problem and the p-Hub Median Problem. Lecture Notes in Computer Science, 4771, 1–15.