XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTIONAL METHODS IN ENGINEERING

Flexible job shop scheduling problem with non-linear routes, energy awareness and position-based learning effect[1]

Débora P. Ronconi[1], Ernesto G. Birgin[2], José Angel Riveaux[2]

[1]*Department of Production Engineering, Polytechnic School, University of São Paulo*
Av. Prof. Luciano Gualberto, 1380, Cidade Universitária, 05508-010 São Paulo, SP, Brazil
dronconi@usp.br
[2]*Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo*
egbirgin@ime.usp.br, jangel.riveaux@usp.br

Abstract. This paper summarizes the main results and conclusions presented in Birgin et al. [1]. The Flexible Workshop Scheduling Problem is addressed with the objective of minimizing energy consumption, considering the possibility of turning off machines between operations. To better capture real-world conditions, this version of FJSP incorporates nonlinear routing and a position-based learning effect. A constructive heuristics and two neighborhood-based local search strategies are presented: SRRN, which reinserts single operations, and SRDRRN, which removes and reconstructs segments of the schedule. Additionally, three metaheuristics—Variable Neighborhood Search, Greedy Randomized Adaptive Search Procedure, and Simulated Annealing —are customized for the problem. Experimental results demonstrate that VNS is effective on large instances, while GRASP achieves high-quality solutions on small instances.

Keywords: flexible job shop, nonlinear routes, learning effect, energy-aware scheduling, metaheuristics.

1 Introduction

Sustainability has become a key objective in manufacturing. This paper addresses the Flexible Job Shop Scheduling Problem (FJSP) with a focus on minimizing energy consumption. The FJSP involves scheduling operations for multiple jobs across machines with flexibility in routing and assignment. In particular in this work several features are tackled: *i*) jobs have multiple feasible processing sequences (nonlinear routing); *ii*) machines may process different operations with varying durations and energy costs; *iii*) the processing time is reduced when operations are scheduled later in a machine's sequence due to a position-based learning effect; and *iv*) machines can be turned off to save idle energy, but restarting incurs a cost. These features reflect real-world industrial settings and add significant complexity to the problem. The goal is to schedule all operations minimizing total energy consumption, considering processing, idle, and start-up energy.

We present a constructive list scheduling heuristic and two neighborhood structures: one based on removing and reinserting a single operation, and another involving destruction and reconstruction strategies. These neighborhoods support two local search algorithms and three metaheuristics—Simulated Annealing, GRASP, and GVNS. The remainder of the paper includes a literature review (Section 2), the proposed heuristic and local search methods (Section 3), and the metaheuristics (Section 4). Section 5 presents computational results, and Section 6 concludes the work with future research directions.

2 Literature review

Previous studies have addressed energy-aware scheduling in the FJSP context. While a few consider learning effects, none incorporate nonlinear routing. Several works model machine on/off decisions to reduce idle energy consumption, such as Meng et al. [2], Zhang et al. [3, 4, 5], Ham et al. [6]. The MILP model used in Birgin et al. [1] extends those by explicitly modeling nonlinear routes via arbitrary precedence graphs and learning effects through position-dependent processing times, building on Araujo et al. [7].

Other contributions include hybrid metaheuristics such as a hybrid pigeon-inspired optimization and simulated annealing algorithm (Wu and Sun [8]), multi-objective algorithms for makespan and emissions reduction using sparrow search (Li and Chen [9]), MILP-based models with metaheuristics like migrating birds optimization (Li et al. [10]), water wave optimization (Lu et al. [11]), shuffled frog-leaping (Lei et al. [12]), and NSGA variants (Gong et al. [13]). These works address objectives like carbon emissions, machine load balancing, labor costs, and degradation effects due to tool wear (Wu et al. [14]), further highlighting the relevance and novelty of incorporating learning and nonlinear routes in the problem.

3 Greedy Constructive Heuristic (GCH) and Local Searches

GCH is list-based scheduling heuristic that selects one operation per iteration to add to the partial solution. In each iteration, it identifies "candidate" operations (whose predecessors have already been scheduled). For each candidate and each compatible machine, it calculates the most economical start time, considering turning the machine on/off or keeping it idle. The machine operation that results in the lowest energy consumption is chosen and scheduled. This heuristic serves as a starting point for local search and metaheuristic strategies. Heuristics of this type have already been successfully employed in the FJS environment. See for example Birgin et al. [15, 16].

Two local search strategies are introduced, which generate neighbors by removing and reinserting an operation in the current solution:
- SRRN (Single Operation Removal and Reinsertion Neighborhood): removes a single operation and attempts to reinsert it in all possible positions of all compatible machines, ensuring that no cycles are formed in the resulting precedence graph. After reinsertion, the effective processing time of the operation and its successors in the machine list is recalculated due to the learning effect.
- SRDRRN (Single Operation Removal, Destruction, Reinsertion, and Reconstruction Neighborhood): when an operation is removed, all operations accessible from it (its successors) are also removed ("destruction"). If the operation is reinserted into the position of another operation, that operation and its successors are also removed. The partially "destroyed" solution is then completed using the constructive heuristic GCH.

The local search iterates, using only one of the two types of neighborhood, until it finds a solution that cannot be improved by a neighbor. The method that consists of calculating an initial solution using the constructive heuristic GCH and applying the local search with the SRRN neighborhood was denominated as GCH-LS-SRRN. Similarly, we call GCH-LS-SRDRR the method using the local search with the SRDRR neighborhood.

4 Metaheuristics

In this section, we briefly describe the three metaheuristics considered, namely, greedy randomized adaptive search procedure (GRASP), simulated annealing (SA) and general variable neighborhood search (GVNS). GRASP (Feo and Resende [17]) starts with an initial solution provided by GCH. Then, it iterates, building an initial solution with a random version of GCH and performing a local search from that solution. In the randomized version of GCH, all $(operation, machine)$ pairs with their respective energy consumption are stored in a list of candidates $\mathcal{L}$ and one pair is drawn from $\max\{1, \lfloor\alpha|\mathcal{L}|\rfloor\}$ best pairs, where $\alpha \in [0,1]$ is a randomization parameter. The parameter $\alpha \in [0,1] \subset$ is the only parameter of GRASP. Two versions were analyzed, GRASP-LS-SRRN and GRASP-LS-SRDRR, depending on the neighborhood used in the local search.

The metaheuristic SA, in each iteration, disturbs the current solution (Shake), accepting the new solution with a probability that depends on its quality and a "temperature" that decreases over time. The probability of accepting a worse solution E', given the current E, is $e^{-\Delta E/t}$, where $\Delta E = (E' - E)/E$ and t is the temperature. There are two versions of the Shake routine, one based on the SRRN neighborhood and the other on the SRDRR neighborhood. Shake in SA-SRDRR introduces more randomness by reconstructing partial solutions with the random version of GCH. Both versions were analyzed. GVNS (Hansen et al. [18]) is generalized version of VNS that iterates between different neighborhoods to explore the search space more broadly. The proposed strategy uses different neighborhoods in both the "Shake" (perturbation) phase between the neighborhoods and in the local search.

5 Numerical Experiments and Results

The numerical experiments aimed to evaluate the performance of the proposed methods—constructive heuristics, local search strategies, and metaheuristics—for the FJS scheduling problem with nonlinear routes, learning effect, and energy consumption minimization. The tests were performed in C++ using an Intel i9-12900K processor. Two sets of instances were used: 50 large and 60 small, totaling 150 and 180 instances, respectively, when including different learning rates. Energy consumption data was added based on discrete uniform distributions. More details can be found in [1].

The local search strategies, LS-SRRN and LS-SRDRR, were compared. Although LS-SRRN iterations were computationally faster, LS-SRDRR proved superior in improving the initial solution generated by the greedy constructive heuristic (GCH), with an average improvement of 9.22% compared to 4.93% for LS-SRRN. This indicates that the more "destructive" neighborhood of LS-SRDRR was more effective in escaping local optima.

The metaheuristics GRASP-LS-SRRN, GRASP-LS-SRDRR, SA-SRRN, SA-SRDRR, and GVNS were tested after parameter calibration with the irace package. GVNS was the most effective in large instances, finding the best solution in 64 of the 150 instances. A statistical analysis (Wilcoxon test) revealed that GVNS and GRASP-LS-SRRN are statistically equivalent, and both significantly outperform the other methods. GRASP-LS-SRDRR was the third best. For longer CPU time limits, GRASP-LS-SRRN, being "cheaper" per iteration, allowed for greater diversification and, on average, led to better solutions in some comparisons. To evaluate the quality of the solutions, the two best-performing metaheuristics (GVNS and GRASP-LS-SRRN) were compared with optimal solutions found by the CPLEX solver. CPLEX was able to prove optimality in only 117 of the 180 small instances within the 1-hour limit. In these 117 instances with a known optimal solution, GRASP-LS-SRRN found solutions with an average deviation of only 0.24% from the optimal solution, while GVNS had an average deviation of 1.84%. Considering all 180 instances (including those without optimality guarantees), GRASP-LS-SRRN had an average deviation of -0.89%, highlighting its effectiveness for smaller instances. In summary, GVNS was the most effective method for large instances, while GRASP-LS-SRRN stood out in small instances, achieving solutions very close to the optimal ones.

6 Conclusions

This work addressed the Flexible Job Shop Scheduling Problem, incorporating nonlinear routes and a position-based learning effect, with the objective of minimizing energy consumption by presenting and evaluating various heuristic approaches. Numerical experiments demonstrated that, for large instances, GVNS was the most effective method, using a combination of the two neighborhoods. For small instances with known optimal solutions, GRASP-LS-SRRN stood out, finding solutions that, on average, were only 0.24% away from the optimal solution. It is important to note that for some small instances where exact solver failed to prove optimization within the time limit, GRASP-LS-SRRN found solutions with a lower objective value than the best feasible solution reported by CPLEX, demonstrating the effectiveness of the heuristic even in challenging cases. To ensure reproducibility and facilitate future comparisons, the instances, code, and solutions found are publicly available. As future work, the authors intend to explore the minimization of total energy cost (TEC) considering costs that vary over time.

Acknowledgements. This work was founded by the Brazilian agencies FAPESP (grants 2013/07375-0, 2022/05803-3, 2022/16743-1, and 2023/08706-1) and CNPq (grants 311536/2020-4 and 302073/2022-1).

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] E. G. Birgin, J. A. Riveaux, and D. P. Ronconi. Energy-aware flexible job shop scheduling problem with nonlinear routes and position-based learning effect. *International Transactions in Operational Research*, 2025.

[2] L. Meng, C. Zhang, X. Shao, and Y. Ren. MILP models for energy-aware flexible job shop scheduling problem. *Journal of Cleaner Production*, vol. 210, pp. 710–723, 2019.

[3] H. Zhang, Z. Deng, Y. Fu, L. Lv, and C. Yan. A process parameters optimization method of multi-pass dry milling for high efficiency, low energy and low carbon emissions. *Journal of Cleaner Production*, vol. 148, pp. 174–184, 2017a.

[4] L. Zhang, Q. Tang, Z. Wu, and F. Wang. Mathematical modeling and evolutionary generation of rule sets for energy-efficient flexible job shops. *Energy*, vol. 138, pp. 210–227, 2017b.

[5] Y. Zhang, Q. Liu, Y. Zhou, and B. Ying. Integrated optimization of cutting parameters and scheduling for reducing carbon emissions. *Journal of Cleaner Production*, vol. 149, pp. 886–895, 2017c.

[6] A. Ham, M.-J. Park, and K. M. Kim. Energy-aware flexible job shop scheduling using mixed integer programming and constraint programming. *Mathematical Problems in Engineering*, vol. 2021, pp. 1–12, 2021.

[7] K. A. G. Araujo, E. G. Birgin, and D. P. Ronconi. Models, constructive heuristics, and benchmark instances for the flexible job shop scheduling problem with nonlinear routes and position-based learning effect. Technical Report MCDO02022024, University of São Paulo, São Paulo, SP, Brazil, 2024.
[8] X. Wu and Y. Sun. A green scheduling algorithm for flexible job shop with energy-saving measures. *Journal of Cleaner Production*, vol. 172, pp. 3249–3264, 2018.
[9] Z. Li and Y. Chen. Minimizing the makespan and carbon emissions in the green flexible job shop scheduling problem with learning effects. *Scientific Reports*, vol. 13, pp. 6369, 2023.
[10] H. Li, H. Zhu, and T. Jiang. Modified migrating birds optimization for energy-aware flexible job shop scheduling problem. *Algorithms*, vol. 13, n. 2, pp. 44, 2020.
[11] Y. Lu, J. Lu, and T. Jiang. Energy-conscious scheduling problem in a flexible job shop using a discrete water wave optimization algorithm. *IEEE Access*, vol. 7, pp. 101561–101574, 2019.
[12] D. Lei, Y. Zheng, and X. Guo. A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption. *International Journal of Production Research*, vol. 55, n. 11, pp. 3126–3140, 2016.
[13] X. Gong, T. De Pessemier, L. Martens, and W. Joseph. Energy- and labor-aware flexible job shop scheduling under dynamic electricity pricing: A many-objective optimization investigation. *Journal of Cleaner Production*, vol. 209, pp. 1078–1094, 2019.
[14] X. Wu, X. Shen, and C. Li. The flexible job-shop scheduling problem considering deterioration effect and energy consumption simultaneously. *Computers and Industrial Engineering*, vol. 135, pp. 1004–1024, 2019.
[15] E. G. Birgin, P. Feofiloff, C. G. Fernandes, de E. L. Melo, M. T. I. Oshiro, and D. P. Ronconi. A MILP model for an extended version of the flexible job shop problem. *Optimization Letters*, vol. 8, n. 4, pp. 1417–1431, 2014.
[16] E. G. Birgin, J. E. Ferreira, and D. P. Ronconi. List scheduling and beam search methods for the flexible job shop scheduling problem with sequencing flexibility. *European Journal of Operational Research*, vol. 247, n. 2, pp. 421–440, 2015.
[17] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, vol. 6, n. 2, pp. 109–133, 1995.
[18] P. Hansen, N. Mladenović, J. Brimberg, and J. A. M. Pérez. *Variable Neighborhood Search*, pp. 57–97. Springer International Publishing, 2018.