

Instance Configuration for Sustainable Job Shop Scheduling

Christian Pérez^{1,2,*}, Carlos March^{1,†} and Miguel Salido¹

¹Universitat Politècnica de València, Instituto de Automática e Informática Industrial, Camí de Vera S/N, Valencia, Spain

²Universitat Politècnica de València, valgrAI - Valencian Graduate School and Research Network of Artificial Intelligence, Camí de Vera S/N, Valencia, Spain

Abstract

The Job Shop Scheduling Problem (JSP) is a pivotal challenge in operations research and is essential for evaluating the effectiveness and performance of scheduling algorithms. Scheduling problems are a crucial domain in combinatorial optimization, where resources (machines) are allocated to job tasks to minimize the completion time (makespan) alongside other objectives like energy consumption. This research delves into the intricacies of JSP, focusing on optimizing performance metrics and minimizing energy consumption while considering various constraints such as deadlines and release dates. Recognizing the multi-dimensional nature of benchmarking in JSP, this study underscores the significance of reference libraries and datasets like JSPLIB in enriching algorithm evaluation. The research highlights the importance of problem instance characteristics, including job and machine numbers, processing times, and machine availability, emphasizing the complexities introduced by energy consumption considerations.

An innovative instance configurator is proposed, equipped with parameters such as the number of jobs, machines, tasks, and speeds, alongside distributions for processing times and energy consumption. The generated instances encompass various configurations, reflecting real-world scenarios and operational constraints. These instances facilitate comprehensive benchmarking and evaluation of scheduling algorithms, particularly in contexts of energy efficiency. A comprehensive set of 500 test instances has been generated and made publicly available, promoting further research and benchmarking in JSP. These instances enable robust analyses and foster collaboration in developing advanced, energy-efficient scheduling solutions by providing diverse scenarios.

Keywords

Job Shop Scheduling Problem, Instance Generation, Benchmarking, Energy consumption, Speed scaling

1. Introduction

The Job Shop Scheduling Problem (JSP) stands as a cornerstone in the realm of operations research and optimization, representing a fundamental challenge pivotal for evaluating algorithmic effectiveness and performance. In essence, JSP revolves around the intricate task of allocating jobs to machines within a manufacturing environment to optimize a plethora of performance metrics, ranging from makespan and flow time to tardiness, resource utilization, and energy consumption [1]. The process of benchmarking in JSP is multi-dimensional, necessitating the definition and evaluation of metrics such as makespan, energy consumption, and tardiness to assess scheduling efficiency and resource utilization [2]. Reference libraries and datasets like JSPLIB play an indispensable role in these benchmarking endeavours, furnishing researchers with a rich array of instances sourced

from seminal works and experimental studies, thereby enriching the evaluation of algorithms [3].

A profound understanding of problem instance characteristics significantly shapes benchmarking efforts in JSP. Factors such as the number of jobs and machines, variability in processing times, machine availability, and precedence relationships exert notable influences on algorithm performance [4]. Furthermore, incorporating energy consumption considerations, contingent upon machine speed and operational attributes, introduces an added layer of complexity to these instances [5]. The delicate balance between energy consumption and scheduling decisions emerges as paramount in achieving energy efficiency goals without compromising production targets [6].

In recent years, the spotlight on addressing energy efficiency within the realm of JSP has intensified, driven by its profound environmental and economic implications [7]. Strategies involving integrating speed-adjustable machines and vehicles have been explored as avenues to optimize energy consumption while upholding productivity levels [8]. Concurrently, developing advanced algorithms and optimization techniques tailored to tackle energy-related challenges has seen significant advancements, considering factors such as machine speed, idle time, and energy requirements [9]. Real-world implementations of these energy-efficient strategies have yielded tangible benefits, manifesting in substantial cost savings

ConfWS'24: 26th International Workshop on Configuration, Sep 2–3, 2024, Girona, Spain

* Corresponding author.

† These authors contributed equally.

✉ cripeber@upv.es (C. Pérez); cmarmoy@upv.es (C. March); msalido@upv.es (M. Salido)

🌐 <https://gps.blogs.upv.es/> (C. Pérez); <https://gps.blogs.upv.es/> (C. March); <https://gps.blogs.upv.es/> (M. Salido)

📄 0000-0002-9121-7939 (C. Pérez); 0009-0009-7525-9133

(C. March); 0000-0002-4835-4057 (M. Salido)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



and positive environmental impacts [10].

In conclusion, the imperative of energy efficiency in JSP research has become increasingly pronounced, paralleling the traditional focus on performance metrics [11]. Benchmarking is a linchpin in evaluating the efficacy of energy-efficient scheduling strategies, furnishing invaluable insights into their ramifications on production efficiency and energy consumption [12]. Manufacturers can embark toward more sustainable and economically viable operations by harnessing advanced optimization techniques and leveraging real-world implementations.

2. Instance Configuration

Test sets play a vital role in JSP research by providing a standardized platform for comparing algorithmic approaches. Their diversity enables researchers to evaluate various algorithms, from heuristics to exact methods, identifying strengths, weaknesses, and potential limitations across different contexts [13]. However, as industrial systems evolve, the complexity of real-world problems increases, necessitating the development or expansion of test sets to simulate real-world scenarios better.

To address this need, the proposed instance configurator operates with the following parameters:

- $J = \{0, \dots, n\}$: the set of jobs.
- $M = \{0, \dots, m\}$: the set of machines.
- $S = \{0, \dots, s\}$: the set of speeds, indexed by s in S .
- T_j : the set of tasks in job j , indexed by $t_{jt} \in T_j, \forall j \in J, \forall t \in M$.
- D_{jt} : the due date of task job $t_{jt} \forall j \in J, \forall t \in M$.
- R_{jt} : the release date of task job $t_{jt} \forall j \in J, \forall t \in M$.
- P_{jts} : the processing time of task job $t_{jt}, \forall j \in J, \forall t \in M, \forall s \in S$.
- E_{jts} : the energy consumption for processing task job $t_{jt}, \forall j \in J, \forall t \in M, \forall s \in S$.

The process of configuring instances, managed by Algorithm 1, is initiated by receiving several input parameters: the number of instances to generate Q , the number of machines M , the count of jobs J , the number of tasks t , the types of release and due dates $rrdd$, random seeds $seeds$, and the distribution $dist$.

Subsequently, the algorithm executes a series of steps to generate instances systematically. The random seed is initialized to ensure reproducibility, and an empty list G is created to store the generated instances (Line 2). The

Algorithm 1 Instance Configurator

input: Quantity of instances Q , Number of machines M , Number of jobs J , Number of tasks T , Release and due date type $rrdd$, Random seed $seed$, Distribution $dist$

output: Generated instances G

```

1: SetSeed(seed)
2:  $G \leftarrow [ ]$ 
3: for  $q$  from 1 to  $Q$  do
4:    $O \leftarrow \text{GenerateJobsOperations}(T, J, M)$ 
5:    $P \leftarrow \text{GenerateProcessingTimes}(J, M, S)$ 
6:    $E \leftarrow \text{GenerateEnergy}(J, M, S)$ 
7:    $R, D \leftarrow \text{GenerateRDDDate}(J, M, dist)$ 
8:    $G \leftarrow G \cup \text{JSP}(O, P, E, R, D)$ 
9: end for
10: return  $G$ 

```

algorithm generates jobs and tasks within a loop iterating through each instance q from 0 to $Q - 1$ (Line 4).

Next, each job operation's processing times and energy consumption are generated.

$$f(x) = \lfloor e^{-\frac{x}{100}} \times 100 \rfloor \quad (1)$$

$$g(x) = 4.0704 \times \frac{\log(2)}{\log(1 + (x \times 2.5093)^3)} \quad (2)$$

The functions responsible for generating the processing times (line 5) and energy consumption (line 6) for the combination of jobs, machines, and speeds are managed by functions $f(x)$ 1 and $g(x)$ 2, as illustrated in Figure 1 as studied in [14]. These functions balance the variables to establish a correlation between processing time and energy consumption. In these equations, x represents the processing time for Equation $f(x)$ 1 and the energy consumption for Equation $g(x)$ 2. Speeds are generated by obtaining the energy consumption percentage for each speed using Equation $f(x)$ 1, which models an inverse relationship between processing time and energy consumption. This involves dividing the interval $[0.5, 3]$ into $|S| - 1$ equal parts, where the boundaries of these new intervals correspond to the energy consumption percentages for each speed. Subsequently, Equation $g(x)$ 2 is utilized to determine the fraction of time corresponding to each speed.

Release and due dates are computed using functions based on the chosen distribution (Line 7). Each distribution offers distinct characteristics suited for modeling various real-world scenarios.

The exponential distribution, defined by its probability density function $f(x; \lambda) = \lambda e^{-\lambda x}$, is ideal for modeling the time until an event occurs, such as machine failures or job arrivals, assuming a constant hazard rate $\lambda > 0$. Its mean is $\frac{1}{\lambda}$. The Gaussian distribution, with mean μ and standard deviation σ , has a probability density function

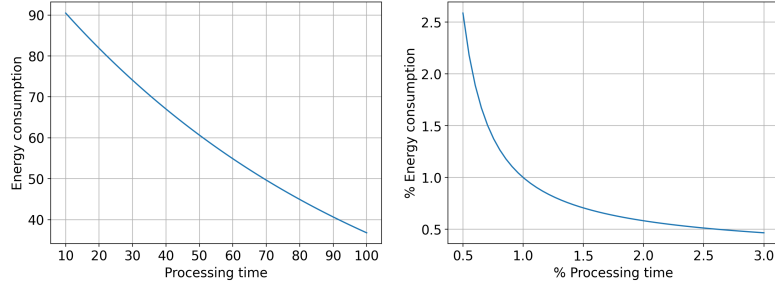


Figure 1: Distribution of processing times and the relationship between time and energy

$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. This distribution represents naturally occurring variations in processing times or delays. The uniform distribution generates values evenly within a specified range, defined by the probability density function $f(x; a, b) = \frac{1}{b-a}$, where a and b are the lower and upper bounds, respectively. It provides a straightforward way to explore a range of scenarios without bias towards any particular value. These distributions were selected due to their unique properties and common use in modeling different real-world data types, enhancing the diversity and comprehensiveness of the generated instances [15].

Additionally, a random start is chosen for each job within a specified range for the release and due date intervals. The time interval between release and due dates is determined based on the median processing time. A random value within the corresponding interval is generated depending on the chosen distribution. This comprehensive approach ensures the creation of instances encompassing a wide range of scenarios, facilitating robust analyses and evaluations.

These steps culminate in constructing a JSP instance (Line 8), incorporating the generated data. Finally, the algorithm returns the list G containing the generated instances. This systematic approach ensures the creation of instances that cover diverse scenarios, which is crucial for comprehensive analyses and evaluations.

3. Generated Problems

A comprehensive set of random instances has been generated following the procedure described in Algorithm 1. These instances exhibit diverse characteristics: the number of jobs ranges from thirty to two hundred fifty, and the number of machines ranges from three to twenty. Normal, exponential, and uniform distributions were utilized in the generation process.

Each instance was extended by relaxing release and due date restrictions. Leveraging three different speed scales, variations of each problem were created, maintain-

ing identical data but with different operational speeds. Specifically, two additional instances were derived from each original: one incorporating the first, third, and fifth-speed scaling and another utilizing only the third-speed scaling.

In addition to varying job and machine counts, the instances encompass different operational complexities and constraints. For instance, some involve jobs with precedence constraints, necessitating certain jobs to be finished before others begin. This complexity challenges algorithms to find optimal solutions efficiently. The chosen distributions—normal, exponential, and uniform—offer a spectrum of scenarios, ranging from predictable and evenly spread job times to highly variable and unpredictable duration. This diversity ensures that the generated instances serve as robust benchmarks to evaluate the performance of scheduling algorithms under varied conditions.

Moreover, a collection of 500 test instances has been generated and made publicly accessible through [16]. These instances incorporate mixed distributions and speed scalings, providing researchers with a comprehensive dataset to evaluate the efficacy of scheduling algorithms. The research group aims to foster collaboration and innovation in planning and scheduling research by facilitating access to these instances. Researchers can use these standardized problems to compare methods and contribute to advancing scheduling solutions.

In summary, the generated instances cover a wide range of job and machine configurations, distribution types, and speed variations, making them suitable for diverse scheduling and planning research applications. Their availability for public use enhances their utility, promoting collaboration and enabling continuous improvement and benchmarking in the field.

4. Acknowledgments

The authors gratefully acknowledge the financial support of the European Social Fund (Investing In Your Fu-

ture), the Spanish Ministry of Science (project PID2021-125919NB-I00), and valgrAI - Valencian Graduate School and Research Network of Artificial Intelligence and the Generalitat Valenciana, and co-funded by the European Union.

References

- [1] H. Xiong, S. Shi, D. Ren, J. Hu, A survey of job shop scheduling problem: The types and models, *Computers & Operations Research* 142 (2022) 105731. doi:<https://doi.org/10.1016/j.cor.2022.105731>.
- [2] D. B. M. M. Fontes, S. M. Homayouni, J. F. Gonçalves, A hybrid particle swarm optimization and simulated annealing algorithm for the job shop scheduling problem with transport resources, *European Journal of Operational Research* 306 (2023) 1140–1157. doi:<https://doi.org/10.1016/j.ejor.2022.09.006>.
- [3] D. M. Torres, F. Barber, M. A. Salido, Psplib-energy: a extension of psplib library to assess the energy optimization in the rcpsp, *Inteligencia Artificial* 17 (2014) 48–61. doi:[10.4114/intartif.vol17iss54pp48-61](https://doi.org/10.4114/intartif.vol17iss54pp48-61).
- [4] M. M. S. El-Kholany, M. Gebser, K. Schekotihin, Problem decomposition and multi-shot asp solving for job-shop scheduling, 2022.
- [5] C. Perez, M. A. Salido, D. Gurrea, A metaheuristic search technique for solving the warehouse stock management problem and the routing problem in a real company, in: *Artificial Intelligence XXXVII*, Springer International Publishing, Cham, 2020, pp. 187–201.
- [6] M. Dai, D. Tang, A. Giret, M. A. Salido, Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints, *Robotics and Computer-Integrated Manufacturing* 59 (2019) 143–157. doi:<https://doi.org/10.1016/j.rcim.2019.04.006>.
- [7] C. Pérez, L. Climent, G. Nicoló, A. Arbelaez, M. A. Salido, A hybrid metaheuristic with learning for a real supply chain scheduling problem, *Engineering Applications of Artificial Intelligence* 126 (2023) 107188. doi:<https://doi.org/10.1016/j.engappai.2023.107188>.
- [8] M. A. Salido, J. Escamilla, A. Giret, F. Barber, A genetic algorithm for energy-efficiency in job-shop scheduling, *Int J Adv Manuf Technol* 85 (2016) 1303–1314. doi:<https://doi.org/10.1007/s00170-015-7987-0>.
- [9] K. Jyothi, R. B. Dubey, Minimizing non-processing energy consumption/total weighted tardiness & earliness, and makespan into typical production scheduling model-the job shop scheduling problem, *Journal of Intelligent & Fuzzy Systems* 45 (2023) 6959–6981. doi:[10.3233/JIFS-222362](https://doi.org/10.3233/JIFS-222362).
- [10] A. Ham, M.-J. Park, K. M. Kim, Energy-aware flexible job shop scheduling using mixed integer programming and constraint programming, *Mathematical Problems in Engineering* 2021 (2021) 1–12. doi:[10.1155/2021/8035806](https://doi.org/10.1155/2021/8035806).
- [11] J. Kotary, F. Fioretto, P. V. Hentenryck, Fast approximations for job shop scheduling: A lagrangian dual deep learning method, *AAAI* 36 (2022) 7239–7246. doi:[10.1609/aaai.v36i7.20685](https://doi.org/10.1609/aaai.v36i7.20685).
- [12] L. Toma, M. Zajac, U. Störl, Solving distributed flexible job shop scheduling problems in the wool textile industry with quantum annealing, 2024. arXiv:2403.06699.
- [13] J. Hoorn, The current state of bounds on benchmark instances of the job-shop scheduling problem, *Journal of Scheduling* 21 (2017) 127–128. doi:[10.1007/s10951-017-0547-8](https://doi.org/10.1007/s10951-017-0547-8).
- [14] D. Morillo Torres, F. Barber Sanchís, M. A. Salido Gregorio, Psplib-energy: Una extension de la libreria psplib para la evaluacion de la optimizacion energetica en el rcpsp, *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial* 17 (2014) 35–48.
- [15] L. Gui, X. Li, Q. Zhang, L. Gao, Domain knowledge used in meta-heuristic algorithms for the job-shop scheduling problem: Review and analysis, *Tsinghua Science and Technology* null (2024) null. doi:[10.26599/tst.2023.9010140](https://doi.org/10.26599/tst.2023.9010140).
- [16] Planning, S. R. Group, Energy Instances for Sustainable Job Shop Scheduling, 2024. doi:[10.5281/zenodo.11640130](https://doi.org/10.5281/zenodo.11640130).