Author's copy of
García Gómez, P., González-Rodríguez, I., Vela, C.R. (2022). Reducing Energy Consumption in Fuzzy Flexible Job Shops Using Memetic Search. In: Ferrández Vicente, J.M., Álvarez-Sánchez, J.R., de la Paz López, F., Adeli, H. (eds) Bio-inspired Systems and Applications: from Robotics to Ambient Intelligence. IWINAC 2022. Lecture Notes in Computer Science, vol 13259. Springer, Cham.
The final authenticated version is available online at: https://doi.org/10.1007/978-3-031-06527-9_14

Reducing energy consumption in fuzzy flexible job shops using memetic search

Pablo García Gómez[1][0000−0001−9084−1337], Inés González-Rodríguez[1][0000−0003−3266−009X], and Camino R. Vela[2][0000−0001−9271−2360]

[1] Universidad de Cantabria, Santander, Spain
[2] Universidad de Oviedo, Oviedo, Spain

Abstract. The flexible job shop is a problem that has attracted much research attention both because of its importance in manufacturing processes and its computational complexity. However, industry is a highly complex environment that is constantly changing, and models and solving methods need to evolve and become richer to stay relevant. A source of complexity is the uncertainty in some parameters, in this work it is incorporated by modeling processing time using triangular fuzzy numbers. We also introduce the objective of reducing energy consumption, motivated by the fight against global warming. To solve the problem, we propose a memetic algorithm, a hybrid method that combines global search with local search. We have put a special focus on the neighborhood functions used to guide the local search since they are key for correct intensification. To assess the performance of the proposed method, we present an experimental analysis that compares the memetic algorithm to a powerful constraint programming solver, and we analyze how the proposed neighborhood functions contribute to increasing the search power of our method.

Keywords: flexible job shop scheduling · energy consumption · fuzzy numbers · memetic algorithm · neighborhood.

1 Introduction

Scheduling has always been a critical problem for most industrial processes, however, the objective has been evolving since the earliest days of industrialization to the present. Historically, the objective was minimizing the maximum completion time, usually known as the makespan, and while it is still relevant it has increasingly been replaced by other production-related measures such as tardiness, as a response to the relocation of factories to different countries. Nowadays the trend is changing again due to newer laws and regulations that seek a reduction in emissions as a countermeasure to global warming. The objective function we propose is framed in this new kind of measures.

As the base problem, we use the flexible job shop scheduling problem (FJSP) that tries to model in a generic way problems that consist in planning the execution of a series of operations on a finite set of resources. In order to increase real-life fidelity, we add uncertainty in the processing time of operations, modeling it using triangular fuzzy numbers. This should help to obtain more robust solutions as they consider the whole range of possible events [15].

Even simpler variants of the job shop problem are NP-Hard problems [9] and therefore intractable with exact algorithms. Hence we will make use of metaheuristics in order to find good solutions in a reasonable time. We propose a memetic algorithm that combines an evolutionary method with a tabu search that employs some new neighborhood functions with an emphasis on reducing energy consumption.

Although energy consumption in scheduling problems is a relatively new topic, several papers already can be found in the literature. They can be organized into three different non-exclusive high-level approaches.

The first one consists in scheduling the operations in such a way that the sum of the energy when the resources are processing and where they are kept idle is as low as possible. In [11] a genetic algorithm based on NSGA-II is proposed to solve the job shop minimizing the total electricity consumption and total weighted tardiness. This work is improved in [7] introducing additional components to help the search. In [12] the model is further extended with the addition of crane transportation between machines.

The second approach consists in allowing to turn on and off the resources more than once. This way, if a resource is going to be idle for a long time it can be turned off and the extra energy used in the starting-up process may be compensated with the savings during the off period. This approach is taken in [10], where the model and method from [11] are extended to allow for switching on and off the resources. This is also the energy model in [2], where new job arrivals are considered and a backtracking search algorithm is proposed.

The third approach consists in slowing down resources so that they consume less energy at the cost of taking longer to process the operations. In [18] a multi-objective genetic algorithm incorporated with two problem-specific local improvement strategies is used to reduce total weighted tardiness together with energy consumption in a job shop problem. In [8] the authors consider a job shop with flexibility and they use a shuffled frog-leaping algorithm to reduce total energy consumption and workload.

The choice of one framework over the others depends on the problem under consideration. In [17] all of them are combined in a flexible job shop and a non-dominated sorted genetic algorithm is used to optimize the makespan, the energy consumption and the number of times resources are turned on/off simultaneously.

To this point, all references are for the deterministic case, considering both energy and uncertainty literature is even more scarce. In [16] the authors propose an evolutionary algorithm to reduce the non-processing energy and the total weighted tardiness, and in [1] a memetic algorithm is proposed to minimize the non-processing energy and the makespan.

The main contributions of this paper are the definition of a model to reduce the energy consumption in the fuzzy flexible job shop (FFJSP) and dealing with the issues introduced with uncertainty, together with the definition of the neighborhood structures to improve solutions in a local search procedure.

The rest of the paper is organized as follows. Section 2 formally defines the problem. Section 3 describes the proposed algorithm. In Section 4 we report and analyze the experimental results to evaluate the potential of our proposal. Finally, Section 5 presents some conclusions.

2 Problem Formulation

The job shop scheduling problem consists in scheduling a set $\mathcal{O}$ of operations (also called tasks) in a set $\mathcal{R}$ of m resources (also called machines) subject to a set of constraints. Operations are organized in a set $\mathcal{J}$ of n jobs, so operations within a job must be sequentially scheduled. Given an operation $o \in \mathcal{O}$, the job to which it belongs is denoted by $\chi_o \in \mathcal{J}$ and the position in which it has to be executed relative to this job is denoted by η_o. The total number of operations in a job j is n_j. There are also capacity constraints, by which each operation requires the uninterrupted and exclusive use of one of the resources for its whole processing time. An operation $o \in \mathcal{O}$ may be executed in any resource from a given set $\mathcal{R}_o \subseteq \mathcal{R}$ and its processing time p_{or} depends on the resource $r \in \mathcal{R}_o$ where it is executed.

We model processing times as triangular fuzzy numbers (TFNs), a particular type of fuzzy numbers [4], with an interval $[a^1, a^3]$ of possible values and a modal value a^2. A TFN can be represented as $a = (a^1, a^2, a^3)$ and its membership function is given by the following expression:

$$\mu_a(x) = \begin{cases} \frac{x-a^1}{a^2-a^1} & a^1 \leq x \leq a^2 \\ \frac{x-a^3}{a^2-a^3} & a^2 \leq x \leq a^3 \\ 0 & x < a^1 \vee a^3 < x \end{cases}$$

For our problem, we need two arithmetic operations, the sum and the maximum. Both operations can be obtained using the extension principle but, unfortunately, the set of TFNs is not closed under the maximum and for this reason we rely on an approximation. This way, the sum of two TFNs a and b is defined as $a+b = (a^1+b^1, a^2+b^2, a^3+b^3)$ and the maximum is approximated as $\max(a, b) \approx (\max(a^1, b^1), \max(a^2, b^2), \max(a^3, b^3))$.

As there exists no natural relation of total order for TFNs, we have to turn to some ranking method. Here, we will use a ranking based on the expected value of TFNs, so $a \leq_E b$ iff $E[a] \leq E[b]$, where $E[a] = \frac{a^1+2a^2+a^3}{4}$.

A solution $(\boldsymbol{\tau}, \mathbf{s})$ to the problem (also called schedule) consists of both a resource assignment $\boldsymbol{\tau}$ and starting time assignment $\mathbf{s}$ to all operations. A solution is said to be feasible if all constraints hold. More formally, let τ_o be the resource assigned to operation $o \in \mathcal{O}$ in this solution and s_o and $c_o = s_o + p_{o\tau_o}$ be its starting and completion times respectively. Then, precedence constraints hold if $\forall i\, c_a^i \leq s_b^i$ when $\eta_a < \eta_b, \chi_a = \chi_b$ and capacity constraints hold if $\forall i\, c_a^i \leq s_b^i \vee \forall i\, c_b^i \leq s_a^i$ when $\tau_a = \tau_b$, for any operation $a, b \in \mathcal{O}$. Notice that the starting time assignment induces a global operation processing order $\boldsymbol{\sigma}$ and a resource operation processing order $\boldsymbol{\delta}_r$ for every $r \in \mathcal{R}$. The position of operation o in $\boldsymbol{\sigma}$ is denoted by σ_o and the position in which operation o is executed in resource τ_o is denoted by δ_o.

For a feasible solution, the makespan is defined as $C_{max} = \max_{j \in \mathcal{J}} C_j$ where $C_j = c_o$ such that $\chi_o = j, \eta_o = n_j$, denotes the completion time of job j.

To this base problem, we add the concept of energy consumption. We consider two types of energy: active energy and passive energy. Passive energy PE_r is intrinsic to each resource $r \in \mathcal{R}$ and can only be reduced by turning them off. It is the product of the passive power consumption of the resource PP_r and the time it is on. We consider that all resources are turned on at the same time (at instant 0) and turned off when all operations are completed, i.e., resources are on for the entire duration of the makespan. Thus, $PE_r = PP_r C_{max}$. Active energy AE_{or} depends on the power AP_{or} required to execute an operation o in a resource r. This way, $AE_{or} = AP_{or} p_{or}$. Our model is an additive model so the total energy consumption of a resource E_r is the result of summing up the passive energy and the active energy $E_r = PE_r + \sum_{o \in \mathcal{O}, \tau_o = r} AE_{or}$. Given these definitions, the total energy consumption is $E = \sum_{r \in \mathcal{R}} E_r$.

This energy function is equivalent to the first approach we explained in the introduction but it is formulated in a different way. Usually, the states of the resources are disjoint and they can contribute to either the processing energy (the energy consumed when they are executing) or the non-processing energy (the energy consumed when they are idle). However, in the fuzzy job shop this means calculating the idle times of resources, i.e., the time span between the end of an operation and the start of the next one in the same resource, which due to the fact that fuzzy subtraction assumes that ill-known numbers are not interactive, some undesired uncertainty may be introduced. For this reason, inspired by energy models used to reduce energy consumption in data centers, we make use of overlapping states, i.e., an additive model where the two involved energies are aggregated.

3 Memetic Algorithm

To solve the problem we use a memetic algorithm, a hybrid algorithm that combines an evolutionary algorithm with local search, taking advantage of the synergies between both methods.

The evolutionary algorithm is composed of a population of solutions that, in each iteration, is replaced by a new one obtained by combining its individuals. To do so, individuals are randomly matched, giving everyone an equal chance to reproduce, and each pair is combined by means of a crossover operator that generates two offspring. The new population is generated by a tournament such that the best two from each pair of parents and their two offspring are chosen. To encode the solutions, we use the tuple of sequences $(\boldsymbol{\tau}, \boldsymbol{\sigma})$ and to decode them, we assign to each operation the earliest starting time such that the order defined by $\boldsymbol{\sigma}$ is not altered. To ensure enough diversity the initial population is generated randomly. A key component here is the crossover operator. We use the extension of the Generalized Order Crossover (GOX) proposed in [5]. Because we use a tournament, this operator is applied unconditionally. We do not make use of any mutation operators because it is incorporated in the local search explained below.

The other component of the memetic algorithm is the local search. In our proposal, offspring generated at each iteration of the evolutionary algorithm are improved using tabu search before the tournament is applied. Tabu search is a local search algorithm that keeps a memory structure, called tabu list, where it stores a trace of the recently visited search space. In particular, to avoid undoing recently made moves, we store in the tabu list the inverse of the moves performed to obtain the neighbors. Our tabu list has a dynamic size, similar to the one introduced in [3], so the size of the list can vary between a lower and an upper bound. When the selected neighbor is worse (resp. better) than the current solution and the upper (resp. lower) bound has not been reached, the list's size increases (resp. decreases) in one unit. If the selected neighbor is the best solution found so far, the list is cleared; this is similar to restarting the search from this solution. We also incorporate an aspiration criterion, so a tabu move can be executed if it improves the best solution found up to this moment. In the rare situation that all neighbors are tabu, we choose the best one, clear the tabu list and slightly change its bounds by picking a random number within a given range.

The neighborhood function is the key component of any local search. In order to describe it we have to introduce some notation. Given a solution $\phi = (\boldsymbol{\tau}, \boldsymbol{\sigma})$, for an operation o, let JP_o (resp. JS_o) denote its predecessor (resp. successor) in its job, $RP_o(\phi)$ (resp. $RS_o(\phi)$) its predecessor (resp. successor) in its resource and $p_o(\phi)$ its processing time in the resource it is assigned to. Its head $h_o(\phi)$ is its earliest starting time, $h_o(\phi) = \max\{h_{JP_o}(\phi)+p_{JP_o}(\phi), h_{RP_o(\phi)}(\phi)+p_{RP_o(\phi)}(\phi)\}$ with $h_o(\phi) = (0, 0, 0)$ if o has no predecessors, and its tail $q_o(\phi)$ is the time left after o has been processed until all other operations are completed, $q_o(\phi) = \max\{q_{JS_o}(\phi)+p_{JS_o}(\phi), q_{RS_o(\phi)}(\phi)+p_{RS_o(\phi)}(\phi)\}$ with $q_o(\phi) = (0, 0, 0)$ if o has no successors. An operation o is said to be makespan-critical in a solution ϕ if there exists a component i of the fuzzy makespan such that $C^i_{max} = (h_o(\phi) + p_o(\phi) + q_o(\phi))^i$. A makespan-critical block for a component i is a maximal sequence B of operations all requiring the same resource, such that no pair of consecutive operations belong to the same job and where every operation meets $C^i_{max} = (h_o(\phi) + p_o(\phi) + q_o(\phi))^i$.

As our energy function is the addition of two different components, the passive energy and the active energy, we can as well differentiate between two types of neighborhoods.

To reduce passive energy consumption we have to reduce the time the resources are active and to do so we have to reduce the makespan. A neighbor can only improve in terms of makespan, and thus passive energy consumption, if there is some change in makespan-critical operations. Also, if a neighbor is obtained by exchanging the position of two consecutive operations, it can be proved that it can only improve in terms of makespan if the operations lie at the extreme of a makespan-critical block.

This motivates the definition of a neighborhood function for passive energy that is the union of two smaller ones, one that acts on the assigned resources (N_{MCORR}) based on [6] and one that acts on the order of the operations (N_{MCET}) based on [13].

Definition 1. Makespan-critical operation resource reassignment neighborhood (N_{MCORR}). *For a feasible solution $\phi = (\boldsymbol{\tau}, \boldsymbol{\sigma})$, let $\boldsymbol{\tau}_{(o,r)}$ denote the assignment that results from reassigning operation o to resource r. Then:*

$$N_{MCORR}(\phi) = \{(\boldsymbol{\tau}_{(o,r)}, \boldsymbol{\sigma}) : o \text{ is makespan-critical}, r \in \mathcal{R}_o, r \neq \tau_o\}$$

Definition 2. Makespan-critical end transpose neighborhood (N_{MCET}). *For a feasible solution $\phi = (\boldsymbol{\tau}, \boldsymbol{\sigma})$, let $\boldsymbol{\sigma}_{(u,v)}$ denote the operation processing order that results from inverting the positions of operations u and v in the same resource. Then:*

$$N_{MCET}(\phi) = \{(\boldsymbol{\tau}, \boldsymbol{\sigma}_{(u,v)}) : u, v \text{ are at the extreme of a makespan-critical block}\}$$

As for active energy consumption, it can only be reduced by moving operations to a more power-efficient resource.

Definition 3. Operation power-efficient resource reassignment neighborhood (N_{OPERR}). *For a feasible solution $\phi = (\boldsymbol{\tau}, \boldsymbol{\sigma})$, let $\boldsymbol{\tau}_{(o,r)}$ denote the assignment that results from reassigning operation o to resource r. Then:*

$$N_{OPERR}(\phi) = \{(\boldsymbol{\tau}_{(o,r)}, \boldsymbol{\sigma}) : r \in \mathcal{R}_o, r \neq \tau_o, AE_{or} < AE_{o\tau_o}\}$$

The complete neighborhood is the union of the neighborhoods for reducing passive energy N_{MCET} and N_{MCORR} with the neighborhood for reducing active energy N_{OPERR}. It is worth mentioning that, although we have designed N_{MCORR} and N_{OPERR} to reduce one type of energy, they can also alter the other because they move operations between resources. Moreover, there can exist some overlapping between them, i.e., there may be repeated neighbors, that should be removed not to increase execution time. On the other hand, N_{MCET} can only alter passive energy. Later we will make an analysis of how each perturbation affects the search.

As neighbor evaluation is the most time-consuming part of the local search, we make use of a filtering mechanism to discard uninteresting solutions. This mechanism consists in evaluating the neighbors following the order defined by a lower bound, and stopping as soon as this lower bound is bigger than the exact value of any of the already evaluated solutions. We integrated the lower bound described in [5] in our objective function.

4 Experimental results

To compare the performance of the proposed algorithm we have used the instances proposed in [14] for a fuzzy flexible job shop. As it is usual in benchmark instances, we have completed the missing power consumption values with random values [2]. For passive power we have taken a value in $[80, 120]$ and for active power we have taken $[1.5, 2.5)$ times the passive power.

The experimental analysis is organized as follows. Since the problem definition is new to this work, there are no previous methods or results available for direct comparisons. In consequence, we first present a comparison between our metaheuristic algorithm and those obtained by CP Optimizer, a commercial software developed by IBM known for its outstanding performance in scheduling problems. Then we analyze the effect of the different defined neighborhoods to assess the benefit of combining all of them in the local search.

All results have been obtained in a Linux machine with two Intel Xeon Gold 6132 processors without Hyper-Threading (14 cores/14 threads) and 128GB RAM using a parallelized implementation of the algorithm in Rust. The source code together with detailed results and benchmark instances can be found at `https://pablogarciagomez.com/research`

In CP Optimizer we have set a limit of 12 hours of real computing time per instance. This limit has been reached in all cases so the solver has returned a lower bound and the best solution found. In the memetic algorithm, after doing a preliminary experimental study, the parameters of the algorithm were set as follows: population size, 100; stopping criterion for the memetic algorithm, 20 iterations without improvement or an average quality of the population equal to the best solution; stopping criterion for the tabu search, 400 iterations without improvement; tabu list lower bound between $[20, 30]$ and tabu list upper bound between $[50, 60]$.

In Table 1 we can see the relative difference between the best and average expected value of the energy obtained by our metaheuristic and the lower bound obtained with CP Optimizer. We also include the relative difference between the best energy found by CP Optimizer and its lower bound. Moreover, we report the time spent by the memetic algorithm in each instance. Our algorithm surpasses CP Optimizer in 11 of the 12 instances and it not only obtains better solutions, but it does so in much less time. In addition, in order to assess that the improvement of our algorithm over CP Optimizer is statistically significant, we have performed Wilcoxon signed-rank test.

To evaluate the performance of the different neighborhoods, we compare the results of executing the memetic algorithm using any possible combination of N_{MCET}, N_{MCORR} and N_{OPERR} as neighborhood in the tabu search. Comparing different neighborhoods can be a tricky task because they can have different characteristics and requisites. In our case, we have opted for only changing neighborhoods and keeping other parameters constant. As we use a dynamic stopping criterion for the tabu search, if one of the neighborhoods happens to be slower to converge, it will be able to use more time while it keeps into the allowed range. To ensure this comparison was fair, after a first execution with the default parameters, we make another one, substantially increasing the stopping criterion. As, by doing this, execution time did not increase significantly, we can

Table 1. Comparison with CP Optimizer

instance	cp optimizer		memetic algorithm			
	lower bound	best(%)	mean(%)	best(%)	real time(s)	cpu time(s)
07a	4827850	9.28	9.08	8.62	39.73	811.93
08a	3771185	16.05	15.21	14.72	58.00	1193.80
09a	3902771	20.01	17.97	17.25	131.60	2689.13
10a	4626061	9.73	9.54	9.04	32.60	670.83
11a	4059000	17.02	15.68	15.10	59.33	1227.27
12a	3572763	20.58	18.72	18.05	107.47	2210.07
13a	6303917	9.27	9.71	9.31	62.90	1324.37
14a	5517284	19.33	17.85	17.28	125.17	2640.53
15a	4668937	23.74	22.62	21.91	254.53	5352.83
16a	5933312	9.84	9.76	9.46	64.47	1363.60
17a	4916945	20.08	18.12	17.57	117.17	2474.97
18a	4731626	23.09	22.02	21.31	219.33	4623.97

conclude that the original dynamic stopping criterion was not halting the search prematurely.

In Figure 1 we can see the relative difference between the lower bound obtained with CP-Optimizer and the result obtained with the different combinations of neighborhoods. We can conclude that N_{MCORR} is the best standalone neighborhood and this can be explained by looking at the properties of the movements performed by each one. Using N_{MCET} alone only optimizes passive energy because it does not change resource assignment. On the other hand, although N_{MCORR} was designed to reduce passive energy consumption, as it moves operations from one resource to another and the tabu search takes the best not forbidden neighbor, at a side effect it also reduces active energy, but only for a subset of all operations. In the case of N_{OPERR}, it can also reduce the passive energy of a solution, but when used alone it is very limited, as it only finds the most efficient resource for operations. By looking at the graph, it is also clear that the complete neighborhood gives the best results. Although in some instances it is surpassed by other combinations, the Wilcoxon signed-rank test confirms its better performance.

5 Conclusions

In this work, we have considered energy consumption minimization in a flexible job shop scheduling problem. The proposed model is based on existing literature for the flexible job shop and the energy function has been built with the objective of fitting into existing proposals both in terms of local search neighborhoods and evolutionary algorithm operators. We have incorporated uncertainty in the problem by considering the processing time of the operations as triangular fuzzy numbers. This way we are solving a problem more similar to real-life situations and improving its applicability. As for the solving method, we have proposed a memetic algorithm, but our focus has been on the neighborhood functions

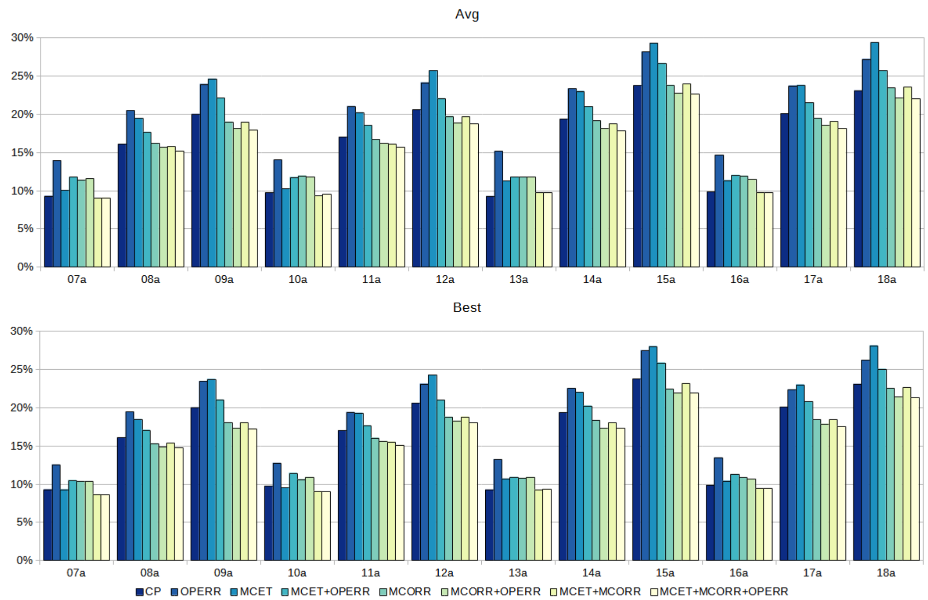

Fig. 1. Comparison between neighborhood functions

because they determine the intensification power of the local search, the key factor to find the best solutions in the explored search space.

We have carried out an experimental analysis, not only to compare our proposal with a commercial solver in order to check its good performance but also to check how the combination of the different neighborhood functions allows achieving better results.

References

1. Afsar, S., Palacios, J.J., Puente, J., Vela, C.R., González-Rodríguez, I.: Multi-objective enhanced memetic algorithm for green job shop scheduling with uncertain times. Swarm and Evolutionary Computation **68**, 101016 (2022). https://doi.org/10.1016/j.swevo.2021.101016
2. Caldeira, R.H., Gnanavelbabu, A., Vaidyanathan, T.: An effective backtracking search algorithm for multi-objective flexible job shop scheduling considering new job arrivals and energy consumption. Computers & Industrial Engineering **149**, 106863 (2020). https://doi.org/10.1016/j.cie.2020.106863
3. Dell'Amico, M., Trubian, M.: Applying tabu search to the job-shop scheduling problem. Annals of Operations Research **41**(3), 231–252 (1993). https://doi.org/10.1007/bf02023076
4. Dubois, D., Prade, H.: Fuzzy numbers: An overview. In: Readings in Fuzzy Sets for Intelligent Systems, pp. 112–148. Elsevier (1993). https://doi.org/10.1016/b978-1-4832-1450-4.50015-8

5. García Gómez, P., Rodríguez Vela, C., González-Rodríguez, I.: A memetic algorithm to minimize the total weighted tardiness in the fuzzy flexible job shop. In: Proceedings of the 19th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2020/2021, Málaga, Spain, September 22-24, 2021 (2021)
6. González, M.A., Vela, C.R., Varela, R.: An efficient memetic algorithm for the flexible job shop with setup times. In: Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling, ICAPS 2013, Rome, Italy, June 10-14, 2013. AAAI (2013)
7. González, M.Á., Oddi, A., Rasconi, R.: Multi-objective optimization in a job shop with energy costs through hybrid evolutionary techniques. In: Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling, ICAPS 2017, Pittsburgh, Pennsylvania, USA, June 18-23, 2017. pp. 140–148. AAAI Press (2017)
8. Lei, D., Zheng, Y., Guo, X.: A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption. International Journal of Production Research **55**(11), 3126–3140 (2017). https://doi.org/10.1080/00207543.2016.1262082
9. Lenstra, J., Kan, A.R., Brucker, P.: Complexity of machine scheduling problems. In: Studies in Integer Programming, pp. 343–362. Elsevier (1977). https://doi.org/10.1016/s0167-5060(08)70743-x
10. Liu, Y., Dong, H., Lohse, N., Petrovic, S.: A multi-objective genetic algorithm for optimisation of energy consumption and shop floor production performance. International Journal of Production Economics **179**, 259–272 (2016). https://doi.org/10.1016/j.ijpe.2016.06.019
11. Liu, Y., Dong, H., Lohse, N., Petrovic, S., Gindy, N.: An investigation into minimising total energy consumption and total weighted tardiness in job shops. Journal of Cleaner Production **65**, 87–96 (2014). https://doi.org/10.1016/j.jclepro.2013.07.060
12. Liu, Z., Guo, S., Wang, L.: Integrated green scheduling optimization of flexible job shop and crane transportation considering comprehensive energy consumption. Journal of Cleaner Production **211**, 765–786 (2019). https://doi.org/10.1016/j.jclepro.2018.11.231
13. Nowicki, E., Smutnicki, C.: A fast taboo search algorithm for the job shop problem. Management Science **42**(6), 797–813 (1996). https://doi.org/10.1287/mnsc.42.6.797
14. Palacios, J.J., González, M.A., Vela, C.R., González-Rodríguez, I., Puente, J.: Genetic tabu search for the fuzzy flexible job shop problem. Computers & Operations Research **54**, 74–89 (2015). https://doi.org/10.1016/j.cor.2014.08.023
15. Palacios, J.J., Rodríguez, I.G., Vela, C.R., Puente, J.: Coevolutionary makespan optimisation through different ranking methods for the fuzzy flexible job shop. Fuzzy Sets Syst. **278**, 81–97 (2015). https://doi.org/10.1016/j.fss.2014.12.003
16. Rodríguez, I.G., Puente, J., Palacios, J.J., Vela, C.R.: Multi-objective evolutionary algorithm for solving energy-aware fuzzy job shop problems. Soft Comput. **24**(21), 16291–16302 (2020). https://doi.org/10.1007/s00500-020-04940-6
17. Wu, X., Sun, Y.: A green scheduling algorithm for flexible job shop with energy-saving measures. Journal of Cleaner Production **172**, 3249–3264 (2018). https://doi.org/10.1016/j.jclepro.2017.10.342
18. Zhang, R., Chiong, R.: Solving the energy-efficient job shop scheduling problem: a multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption. Journal of Cleaner Production **112**, 3361–3375 (2016). https://doi.org/j.jclepro.2015.09.097