A Q-Learning Proposal for Tuning Genetic Algorithms in Flexible Job Shop Scheduling Problems

Christian Perez[1], Carlos March[2], Miguel A. Salido[2]

[1]Instituto Valenciano de Inteligencia Artificial, Spain
[2]Instituto de Automatica e Informatica Industrial, Universitat Politecnica de Valencia, Spain
cripeber@upv.es, cmarmoy@etsinf.upv.es, msalido@dsic.upv.es

Abstract

Genetic algorithms (GAs) belong to the category of evolutionary algorithms and are frequently utilized for resolving challenging combinatorial problems. However, they typically require customization to suit a particular problem type, and their performance is heavily influenced by numerous hyperparameters and reproduction operators. In this work, we propose a Reinforcement Learning approach for fine-tuning Genetic Algorithms in Flexible Job Shop Scheduling problems (FJSP), where the main parameters involved in the genetic algorithm operators are trained to allocate the most promising values. The approach returns an optimized schedule taking into account given constraints specific to the scenario, such as the relationship among release date, due date, and processing time, which machines must be selected out of a set of alternative machines, or which sequence-dependent setup time can be filtered. The approach takes input data in the form of FJSP instances by varying the numbers of jobs and machines and then uses the NSGA-II algorithm to generate solutions. These solutions are stored in a Solutions module and they are analyzed using a Principal components analysis (PCA) to identify clusters of similar instances and solutions. The Q-Learning module then generates hyperparameters for each iteration of the NSGA-II algorithm based on information from the previous modules. A toy example is presented to better understand the behavior of the proposal and the results obtained for optimizing further instances of the problem in a more efficient way.

Introduction

The industrial sector is one of the major energy consumers worldwide, accounting for more than half of the total global energy consumption (IEA, 2019). As a result, energy efficiency has become a major concern for the industry, as it not only helps reduce energy costs but also reduces greenhouse gas emissions and environmental impacts.

Job shop scheduling is an essential problem in the industry that deals with the allocation of resources and scheduling of tasks to optimize performance. Traditionally, the objective of job shop scheduling has been to minimize makespan or flowtime, which are metrics related to the time it takes to complete all tasks. However, with the increasing importance of energy efficiency, there has been a growing interest in incorporating energy consumption as an objective function in job shop scheduling.

Several metaheuristic techniques have been used for solving different versions of job shop scheduling problems (Mehdi et al., 2016)(Arora and Agarwal, 2016). Minimizing energy consumption in Job shop scheduling problems has gained increasing attention in recent years (Para et al., 2022)(Maiterth et al., 2018) covering various aspects of energy-efficient job shop scheduling, including problem formulations, objectives, and optimization techniques. Learning methods have also played a role, in the application of reinforcement learning (RL) to solve job shop scheduling problems. RL can be used to learn a policy that maps the current state of the scheduling problem to a scheduling decision. Thus the objective is to learn a policy that maximizes a reward function, which is typically defined as a function of the scheduling performance, such as the makespan or the energy consumption. Some works have been proposed using RL in Job Shop Scheduling problem, although most of them applied to specific scenarios (Palacio et al., 2022).

In this paper, we apply RL to learn the most appropriate values for tuning the main parameters of a Genetic Algorithm for solving energy-aware unrelated parallel machine scheduling problems with machine-dependent energy consumption and sequence-dependent setup time.

Description of the Scheduling Problem

The problem addressed in this study involves the scheduling of a collection of orders, where each order corresponds to a job j, across a set of independent parallel machines m. Each job is associated with two temporal characteristics: a release date Rd_j, which signifies the earliest time that the job can be processed, and a due date Dd_j, which marks the latest time by which the job must be completed. Failure to complete a job by its due date results in a penalty cost that reflects the priority of the job. Higher-priority jobs are subject to higher penalty costs for each unit of time beyond the expected completion time, relative to lower-priority jobs.

Each job must be executed on a single machine selected from a set of eligible machines that are specific to the job. Jobs that can be carried out on more than one machine are referred to as shared jobs, and the set of eligible machines for these jobs has more than one element. The energy consumption E_{mj} and processing time T_{mj} required for each job depend on the machine selected to execute it. As such, different machines may require varying amounts of energy and time to complete the same job. For instance, newer machines may be faster and more energy-efficient than older ones.

Upon completion of each job, a number of physical actions must be performed on the machine before it can process another job. These actions include cleaning and maintenance, among others, and therefore a setup time between jobs must be taken into account. The setup time is dependent on the particular job pair being executed sequentially, as well as the characteristics of the machine, resulting in a unique setup time for each job pair on each machine $Jps_m = [JxJ]$. Note that pairs of jobs that do not have setup time are re-scored with a zero.

The formal mathematical model is available in (Nicolò et al., 2021) and it can be represented as :

$$R_m|M_j, p_{jk}, E_{jk}, r_j, s_{ijk}| \sum w_j T_j, \sum E_{jk}, \sum S_{ijk}$$

where the problem has three measures to be minimized, which are expressed as objective functions: the total weighted tardiness of the jobs $TT(s)$, the total energy consumption $EN(s)$, and the total setup time $ST(s)$. The solution s^* can be obtained by minimizing a 3-dimensional objective function:

$$s^* = \arg\min_{s \in S} [TT(s), EN(s), ST(s)] \tag{1}$$

where S denotes the feasibility space for the problem solution space.

System Proposal

The proposed system is designed to solve instances of the Flexible Job Shop Scheduling Problem (FJSP) using a combination of machine learning and optimization techniques. The system takes input data in the form of FJSP instances by varying the numbers of jobs and machines, extracts features from that data (using functions: 2 3 4 5) to generate a simple representation of the problems and then uses the NSGA-II algorithm to generate and store solutions which are further analyzed.

$$\sum_{m \in M} \sum_{j \in J} \frac{Rd_j - Dd_j}{T_m j} \tag{2}$$

$$\sum_{j \in J} \max_{m \in M} E_{mj} \tag{3}$$

$$\sum_{j \in J} \max_{m \in M} T_{mj} \tag{4}$$

$$\sum_{m \in M} \sum_{p \in Jps_m} \max_{s \in p_j} ST_{mps} \tag{5}$$

The idea of representing the FJSP instances as D-arrays is similar to the well-known dataset iris. We use functions that allow the system to obtain values to discriminate the main differences among instances and to evaluate the quality of the solutions. To obtain the number of times a job can be within its time window (between the release date and the due date), it is used the equation 2. Equation 3 takes into account the total sum of the maximum energy cost of each job among all available machines 3, and the same operation for the maximum processing cost for each job on all available machines is obtained by using equation 4. In equation 5, it is obtained the added setup time of the worst possible sequence of jobs. Once we have de $\mathbb{R}^4$ representation of each instance we normalize them by using a min-max scaler.

The Q Learning module then generates hyperparameters for each iteration of the NSGA-II algorithm based on information from the previous modules. Overall, the system aims to provide an efficient and effective way of solving FJSP instances.

Algorithm 1 Q-metatuning

input: Q-learning matrix for each mataparameter Q_p, $\mathbb{R}^4$ representation of the problem instances *ins* and population *ppl*, Φ metaparameters, Iteration state t and the values of α and γ.
output: Best metaparameters in next state Φ^*
For p **in** $|\Phi|$:
    $rw \leftarrow$ **getRewardValue(**ins, ppl**)**
    $learn \leftarrow rw + \gamma \max_{a \in \Phi_p} \quad Q_p(t+1, a) - Q_p(t, \Phi_p)$
    $Q_p(t, \Phi_p) \leftarrow Q_p(t, \Phi_p) + \alpha \; learn$
end for
$\Phi^* \longleftarrow \arg\max_{x \in A} \quad Q_p(t+1, x) \quad | \quad \forall p \in \Phi$

Algorithm 1 outlines the training procedure for the QL module. Initially, the module receives three inputs: the parameters of the t-th iteration from the NSGA-II module, the $\mathbb{R}^4$ representation of the population (solutions), and the feautres of the problem instances. Subsequently, the module uses this input to re-estimate the Q-value for the state s-1. This is achieved by computing the reward function for each hyperparameter by applying the Bellman Equation. The final step of the training procedure involves the generation of a solution for the current state s, which serves as the hyperparameters for the NSGA-II module in the subsequent iteration.

Conclusion

This work proposes a Reinforcement Learning approach for fine-tuning Genetic Algorithms in general Flexible Job Shop Scheduling problems (FJSP), where the main parameters involved in the genetic algorithm operators are trained to allocate the most promising values. Preliminary results show that the proposal can be useful for solving new and general instances of FJSP.

Acknowledgements

The authors gratefully acknowledge the financial support of the European Social Fund (Investing In Your Future), the Spanish Ministry of Science (PID2021-125919NB-I00).

References

[IEA, 2019] (2019). Global energy co2 status report 2019. *License: CC BY 4.0*.

[Arora and Agarwal, 2016] Arora, D. and Agarwal, G. (2016). Meta-heuristic approaches for flowshop scheduling problems: a review. *International Journal of Advanced Operations Management*.

[Maiterth et al., 2018] Maiterth, M., Koenig, G., Pedretti, K., Jana, S., Bates, N., Borghesi, A., Montoya, D., Bartolini, A., and Puzovic, M. (2018). Energy and Power Aware Job Scheduling and Resource Management: Global Survey — Initial Analysis. In *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 685–693, Vancouver, BC. IEEE.

[Mehdi et al., 2016] Mehdi, G., Bouzouia, B., Achour, N., and Tebani, K. (2016). *Meta-heuristics Approaches for the Flexible Job Shop Scheduling Problem*, pages 285–314.

[Nicolò et al., 2021] Nicolò, G., Ferrer, S., Salido, M. A., Giret, A., and Barber, F. (2021). A Multi-Agent Framework to Solve Energy-Aware Unrelated Parallel Machine Scheduling Problems with Machine-Dependent Energy Consumption and Sequence-Dependent Setup Time. *Proceedings of the International Conference on Automated Planning and Scheduling*, 29:301–309.

[Palacio et al., 2022] Palacio, J. C., Jiménez, Y. M., Schietgat, L., Doninck, B. V., and Nowé, A. (2022). A q-learning algorithm for flexible job shop scheduling in a real-world manufacturing scenario. *Procedia CIRP*, 106:227–232. 9th CIRP Conference on Assembly Technology and Systems.

[Para et al., 2022] Para, J., Del Ser, J., and Nebro, A. J. (2022). Energy-aware multi-objective job shop scheduling optimization with metaheuristics in manufacturing industries: A critical survey, results, and perspectives. *Applied Sciences*, 12(3).