A Grouping Genetic Algorithm for the Unrelated Parallel-Machine Scheduling Problem

Octavio Ramos-Figueroa
Universidad Veracruzana, Artificial Intelligence Research Institute
Xalapa, Veracruz, MEXICO
oivatco.rafo@gmail.com

Marcela Quiroz-Castellanos
Universidad Veracruzana, Artificial Intelligence Research Institute
Xalapa, Veracruz, MEXICO
maquiroz@uv.mx

ABSTRACT

A large number of NP-hard grouping problems have been addressed with the Grouping Genetic Algorithm (GGA). The unrelated parallel-machine scheduling with makespan minimization, also known as $R||C_{max}$, is an NP-hard combinatorial optimization problem that belongs to the NP-hard grouping problems family. This paper presents a GGA with variation operators specifically designed for the problem $R||C_{max}$. The performances of the proposed GGA is assessed by solving 1400 test instances of the problem $R||C_{max}$ taken from the specialized literature. The experimental results suggest that GGA is able to find high-quality solutions outperforming the best state-of-the-art GA.

CCS CONCEPTS

• **Applied computing** → **Operations research**; • **Computing methodologies** → **Discrete space search**; • **Mathematics of computing** → **Combinatorial optimization**.

KEYWORDS

Parallel machine scheduling problem, Grouping genetic algorithm, Representation scheme based on groups

ACM Reference Format:
Octavio Ramos-Figueroa and Marcela Quiroz-Castellanos. 2021. A Grouping Genetic Algorithm for the Unrelated Parallel-Machine Scheduling Problem. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459531

1 INTRODUCTION

This work tackles the well-known NP-hard grouping problem Parallel-machine scheduling problem with unrelated machines, jobs with no-preemptions, and the reduction of the maximum completion time as the optimization objective, stated as $R||C_{max}$ [6]. Given a set of jobs $N = \{j_1, ..., j_n\}$ and a set of parallel machines $M = \{i_1, ..., i_m\}$, the problem $R||C_{max}$ consist of identifying the most efficient sequential scheduling of the n jobs in the m machines. That is, the one that minimize the makespan (the highest processing time C_i of the m machines). In such a way that each machine i can process only one job j at a time, and each job j must be processed by a single machine i [6].

The specialized literature contains solution methods that tackle the problem $R||C_{max}$ with different approaches, covering deterministic methods [1], rounding methods or two-phase algorithms [4], branch and bound algorithms [6], and metaheuristic algorithms like local searches [2], the Particle Swarm Optimization (PSO) algorithm [5], and Genetic Algorithms (GA) [3], to mention some examples.

The state-of-the-art indicates that the GA is the most used metaheuristic to solve grouping problems, mainly the grouping genetic algorithm (GGA). Such demand is associated with its promising results and its adaptability to incorporate new ideas to handle the constraints and conditions with different characteristics [5]. The specialized literature includes some GAs that address the $R||C_{max}$, we presented the most recent genetic algorithm with the group-based representation (GGA) in 2020 [5]. In such work, we introduced an adaptation of a GGA that employs genetic operators designed with knowledge of the Bin Packing problem domain.

This work presents an Enhanced GGA with crossover and mutation operators that incorporate knowledge of the $R||C_{max}$ problem-domain to improve its performance when solving this problem.

2 THE PROPOSED APPROACH

In this work, we present an enhanced GGA to the GGA presented in [5] that incorporates genetic operators specially designed for the problem $R||C_{max}$. The proposed GGA uses the group-based representation scheme, the strategy to generate the initial population, the fitness function, assignment heuristic Min(), the selection and replacement mechanisms, as well as the reproduction technique of the original GGA. On the other hand, the enhanced GGA uses a crossover and a mutation operator introduced in this work, called Two Sorting Criteria crossover (TSCX) and Item Download mutation, respectively. The new operators work as follows.

2.1 Mutation operator

The Item Download mutation randomly selects one of the machines (w) with a processing time equal to the makespan ($C_i = C_{max}$) and one of the remaining machines (o). Subsequently, it randomly chooses a job (j_w and j_o) from each machine to release them and later reinsert them with the assignment heuristic Min(). Additionally, we proposed the rearrangement heuristic Assemble. This heuristic only is used if, after releasing and reinserting the jobs, the genetic material of the mutated solution has not been altered. Figure 1 contains a flow chart with the rearrangement heuristic procedure. For each job j in the selected machines w and o, Assemble traverses the m machines and first tries to apply the function Insertion(S, j_{sm},

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3459531

sm, i) that returns the solution derived from S by inserting the job j_{sm} (w or o) into the machine i. Later, it tries to apply the function Interchange(S, j_{sm}, sm, j_i, i) that returns the solution derived from S by exchanging job j_{sm} (j_w and j_o) from the selected machine sm (w or o) with the job j_i in machine i. Assemble performs the operations only if the makespan of the resulted solution will be equal or better than the initial solution. The rearrangement heuristic stops once one of the operations is performed.

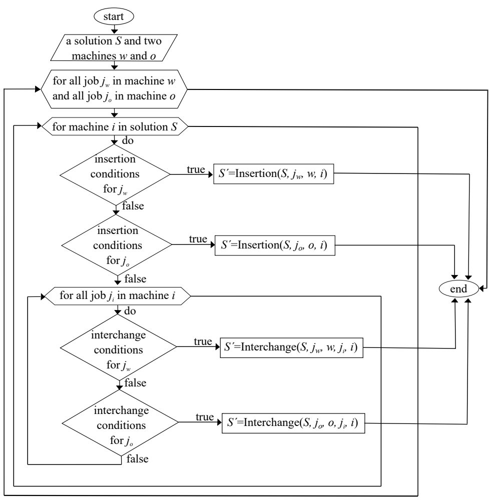

Figure 1: Flow chart of the rearrangement heuristic Assemble().

2.2 Crossover operator

TSCX uses two sorting criteria; first, it sorts the machines in parents from best to worst, considering the makespan (the first criterion). Later, TSCX compares the sorted machines of both parents in parallel, transmit the one with the higher number of jobs (the second criterion) first, and later the other one omitting the repeated machines and removing the repeated jobs. Finally, unscheduled jobs are permuted and scheduled with the assignment heuristic Min().

3 EXPERIMENTS AND RESULTS

In order to show the efficiency of the Enhanced GGA proposed in this work, we compare its performance versus the most recent genetic algorithms of the state-of-the-art by solving 1,400 test instances, introduced by Fanjul-Peyro in 2010 [2]. That is a GA with the extended permutation encoding introduced in [6] and a GGA presented in [5]. For a fair comparison with the GGA presented in [5], the Enhanced GGA iterates for 500 generations. Furthermore, the proposed metaheuristic uses a population size = 100, a crossover rate = 0.4, and a mutation rate = 0.8. In this way, the algorithms are compared based on the Relative Percentage Deviation (*RPD*). Given an instance i, the *RPD* is defined as $(C_{max}(i) - C^*_{max}(i))/C^*_{max}(i)$, where $C_{max}(i)$ depicts the C_{max} value found by the evaluated algorithm and $C^*_{max}(i)$ depicts the best C_{max} found using two hours of the commercial solver CPLEX (for the instances in which CPLEX cannot find the optimal solution). Thus, *RPD* indicates the deviation from the compared algorithms to CPLEX.

We analyze the algorithm performance considering the distribution of the processing times p_{ij} of the instances and the average of the 1400 instances. Table 1 shows the experimental results. In this table, the rows depict the instance sets, and the columns indicate the average RPD reached by each assessed algorithm. From Table 1 it can be observed that the Enhanced GGA presented in this work excelled in all the instance sets. Additionally, the Wilcoxon rank-sum test indicates that the original GGA and the Enhanced GGA are statistically significant with a 95%-confidence level with a p-Value = $5.85e^{-85}$.

Table 1: GA, GGA, and Enhanced GGA comparison using RPD.

Instance set	GA [6]	GGA [5]	Enhanced GGA
$U(1, 100)$	2.17	0.07	**0.05**
$U(10, 100)$	1.71	0.10	**0.04**
$U(100, 120)$	0.16	0.02	**0.01**
$U(100, 200)$	0.61	0.07	**0.03**
$U(1000, 1100)$	0.05	0.01	**0.01**
MacsCorr	1.81	0.08	**0.04**
JobsCorr	0.25	0.05	**0.05**
Average	0.97	0.06	**0.03**

CONCLUSIONS AND FUTURE WORK

In this paper, we proposed two variation operators that incorporate knowledge of the $R||C_{max}$ problem domain, the TSCX crossover operator, and the Item Download mutation operator. Later, we incorporated the proposed operators into a state-of-the-art GGA, and we calibrated the crossover and the mutation rates. Once configures, the Enhanced GGA performance was assed by solving 1400 test instances. The Enhanced GGA showed competitive results with significant differences and an improvement rate of above 50%. The experimental results suggest that a GGA with operators designed with knowledge of the problem domain can reach better results than a GGA with operators designed for other grouping problems. As future work, we will work in the design of new crossover and mutation operators that exploit the group-based representation scheme properties and the knowledge of the problem domain of other grouping problems.

REFERENCES

[1] Ernest Davis and Jeffrey M Jaffe. 1981. Algorithms for scheduling tasks on unrelated processors. *Journal of the ACM (JACM)* 28, 4 (1981), 721–736.
[2] Luis Fanjul-Peyro and Rubén Ruiz. 2010. Iterated greedy local search methods for unrelated parallel machine scheduling. *European Journal of Operational Research* 207, 1 (2010), 55–69.
[3] CA Glass, CN Potts, and P Shade. 1994. Unrelated parallel machine scheduling using local search. *Mathematical and Computer Modelling* 20, 2 (1994), 41–52.
[4] VS Kumar, Madhav V Marathe, Srinivasan Parthasarathy, and Aravind Srinivasan. 2009. A unified approach to scheduling on unrelated parallel machines. *Journal of the ACM (JACM)* 56, 5 (2009), 28.
[5] Octavio Ramos-Figueroa, Marcela Quiroz-Castellanos, Efrén Mezura-Montes, and Oliver Schütze. 2020. Metaheuristics to solve grouping problems: A review and a case study. *Swarm and Evolutionary Computation* 53 (2020), 100643.
[6] Veronique Sels, José Coelho, António Manuel Dias, and Mario Vanhoucke. 2015. Hybrid tabu search and a truncated branch-and-bound for the unrelated parallel machine scheduling problem. *Computers & Operations Research* 53 (2015), 107–117.