

Scheduling of Unrelated Parallel Machines with Secondary Resources and Interdependent Jobs

Hakan Gultekin

Department of Mechanical and Industrial Engineering
College of Engineering, Sultan Qaboos University
Muscat, Sultanate of Oman
hgultekin@squ.edu.om

Nasr Al-Hinai

Department of Mechanical and Industrial Engineering
College of Engineering, Sultan Qaboos University
Muscat, Sultanate of Oman
nhinai@squ.edu.om

Abstract

This paper studies a complex scheduling problem motivated by a plastic injection molding environment, where multiple product components must be produced and delivered simultaneously. The system consists of unrelated parallel machines that require secondary resources, such as molds, with compatibility restrictions between jobs, machines, and molds. Processing times vary across machines, and sequence- and machine-dependent setup times arise due to mold changes. In addition, some machines and molds may already be occupied at the start of the planning horizon. A key feature of the problem is the presence of interdependent jobs, representing components of the same final product. Although these components can be processed independently, their completion times jointly determine delivery performance. To capture this dependency, job tardiness is defined based on the latest completion time among related components, which introduces additional coordination challenges. The objective is to construct feasible schedules that minimize total weighted tardiness while respecting all operational constraints. To address this problem, we propose mixed-integer programming formulations and develop a tailored genetic algorithm to obtain high-quality solutions within reasonable computational times. A computational study based on randomly generated instances is conducted to evaluate the effectiveness of the proposed approaches. The results indicate that the heuristic consistently produces near-optimal solutions and significantly outperforms the exact model for medium- and large-scale instances within practical time limits.

Keywords

Unrelated parallel machines, Scheduling, Resource constraints, Mathematical programming, Heuristics

1. Introduction

This study considers the unrelated parallel machine scheduling problem with resource constraints. It is named as unrelated parallel machines since the processing time of a job differs according to the machine to which it is assigned. This problem arises in a real-life industrial setting in a plastic injection molding department. In this type of industry, a company may produce plastic containers for paints and some chemicals. The containers consist of a bucket, lid, and handle-all made of plastic. They differ in terms of material, size, label, and design. Depending on the customer orders and forecasts, the weekly and daily production requirements are exploded by the enterprise resource planning (ERP)

system. These production requirements include the product type, quantity, customer delivery date (due dates), weights that describe the customer priorities, and release time of the raw material.

The production system includes multiple plastic injection machines of different types, each with distinct capabilities in terms of size, compatible molds, and processing times. Jobs are processed using molds, which vary in design and weight. Not all molds are compatible with all machines or suitable for all jobs, and some jobs can be processed using alternative molds.

Due to the description of the plastic containers above, each of their components is defined as a separate "job". On the other hand, when the final product is to be delivered to the customer, these three components are delivered at the same time. Therefore, their delivery date and customer priority are identical. Since they are represented as separate jobs, each can be produced on different machines at different times in the final schedule. However, this may lead to large work in process (WIP) inventories due to some finished components waiting for other accompanying ones to be completed. Therefore, the jobs are assumed to be interdependent. The completion time of a job is not its own completion time but defined as the latest completion time of all the dependent components. Consequently, the tardiness of a job is calculated using this dependent completion time.

At the beginning of the scheduling horizon, some machines may already be processing jobs carried over from the previous period, with known completion times. Consequently, certain molds are initially installed on the machines. When switching from one job to another, sequence- and machine-dependent setup operations are required to change molds. These setup times depend on both the machine and the pair of molds involved.

The problem, therefore, involves determining job-machine-mold assignments and sequencing decisions on each machine while accounting for machine eligibility, mold compatibility, setup times, and job interdependencies.

1.1 Objectives

The objective of this research is to develop effective solution approaches for a complex unrelated parallel machine scheduling problem with resource constraints and interdependent jobs. The underlying optimization problem aims to determine a production schedule that minimizes the total weighted tardiness of jobs, where tardiness is defined based on the completion time of the latest component within each product set, reflecting the interdependent structure of jobs and customer priorities.

To address this problem, this study develops a mixed-integer programming (MIP) formulation and proposes an efficient heuristic algorithm. The performance of the proposed approaches is evaluated through a comprehensive computational study.

2. Literature Review

The considered problem is a more complex version of the classical unrelated parallel machine scheduling problem and therefore it is NP-hard (Framinan et al., 2014). Bektur and Saraç (2019) considered the unrelated parallel machine scheduling problem with sequence dependent setup times and a common server to perform the setup operations. They developed MIP formulation and heuristic algorithms. Villa et al. (2018), Yepes-Borrero et al. (2021), Vallada et al. (2019), and Afzalirad and Rezaeian (2016) also considered the unrelated parallel machine scheduling problem with resource constraints and makespan objective. They developed heuristic algorithms for the problem.

Other works have explored different aspects of the problem. For example, Fanjul-Peyro et al. (2017) addressed the problem with no setup time and with the makespan minimization objective. They developed mathematical programming formulations and heuristic algorithms. Fanjul-Peyro (2020) considered the problem with sequence and machine-dependent setup times and makespan objective. They developed an exact solution method for the problem. Ekici et al. (2019) considered the problem with machine eligibility constraints to minimize the total tardiness and earliness in an electronics manufacturer and developed several heuristic algorithms. Chen et al. (2022) considered a similar problem with machine eligibility and availability constraints with multiple objectives. They developed a mathematical programming formulation and a Tabu Search algorithm. Fang et al. (2022) developed a hybrid metaheuristic algorithm based on LA-ALNS and tabu search. Perez-Gonzalez et al. (2019) considered the problem

with the objective of minimizing the total tardiness. They developed mathematical programming formulations and heuristics with novel repair and improvement phases.

The current study extends the literature by incorporating interdependent jobs, realistic mold-machine compatibility constraints, initial system conditions, and sequence-dependent setups within a unified framework. In the following, a mixed-integer linear programming (MIP) formulation is developed for the problem.

3. Methods

3.1 Mathematical Programming Formulation

There are m unrelated parallel machines and p molds to produce n jobs. An artificial job numbered 0 will also be used that will precede the first job on each machine to correctly formulate the problem. All parameters and decision variables that are used in the model are listed below:

Problem Parameters

m = number of machines

n = number of jobs

p = number of molds

p_{ij} = processing time of job j on machine i

r_j = release time of job j

d_j = due date of job j

w_j = weight of job j

μ_i = the mold which is attached on machine i at the beginning of the scheduling period

ϵ_i = completion time of the prescheduled jobs on machine i

S_{irk} = setup time for switching from mold r to mold k on machine i

$f_{kj} = \begin{cases} 1, & \text{If job } j \text{ can be processed by mold } k \\ 0, & \text{Otherwise} \end{cases}$

$c_{ik} = \begin{cases} 1, & \text{If mold } k \text{ can be attached to machine } i \\ 0, & \text{Otherwise} \end{cases}$

$b_{jh} = \begin{cases} 1, & \text{If job } j \text{ and job } h \text{ are interdependent} \\ 0, & \text{Otherwise} \end{cases}$

Decision variables

$y_{ij} = \begin{cases} 1, & \text{If job } j \text{ is processed on machine } i \\ 0, & \text{Otherwise} \end{cases}$

$x_{ijh} = \begin{cases} 1, & \text{If job } j \text{ is followed by job } h \text{ on machine } i \\ 0, & \text{Otherwise} \end{cases}$

$z_{ikj} = \begin{cases} 1, & \text{If job } j \text{ is processed with mold } k \text{ on machine } i \\ 0, & \text{Otherwise} \end{cases}$

$K_{ijkkr} = \begin{cases} 1, & \text{If job } j \text{ is processed on mold } k \text{ before mold } r \text{ on machine } i \\ 0, & \text{Otherwise} \end{cases}$

$g_{jh} = \begin{cases} 1, & \text{If job } j \text{ is processed before job } h \\ 0, & \text{Otherwise} \end{cases}$

$u_i = \begin{cases} 1, & \text{If machine } i \text{ is used} \\ 0, & \text{Otherwise} \end{cases}$

B_j = Starting time of job j

T_j = Tardiness of job j

Mathematical Model

$$\text{Min} \quad \sum_{j=1}^n w_j T_j \quad (1)$$

$$\text{Subject To:} \quad T_j \geq b_{jh} \left(B_h + \sum_{i=1}^m p_{ih} y_{ih} \right) - d_j, \quad \forall j, h \quad (2)$$

$$\sum_{i=1}^m y_{ij} = 1, \quad \forall j > 1 \quad (3)$$

$$B_j \geq \epsilon_i y_{ij} + s_{i\mu_i k} z_{ikj}, \quad \forall i, j, k : j > 0 \quad (4)$$

$$y_{i0} = 1, \quad \forall i \quad (5)$$

$$x_{ihj} \leq y_{ij}, \quad \forall i, j, h \quad (6)$$

$$x_{ijh} \leq y_{ij}, \quad \forall i, j, h \quad (7)$$

$$y_{ij} = \sum_{h=1}^m x_{ihj}, \quad \forall i, j (j > 0) \quad (8)$$

$$y_{ij} = \sum_{h=1}^m x_{ijh}, \quad \forall i, j (j > 0) \quad (9)$$

$$B_h \geq B_j + \sum_{i=1}^m p_{ij} y_{ij} + \sum_{i=1}^m s_{irk} K_{ijk} - M \left(3 - \sum_{i=1}^m (x_{ijh} + z_{irj} + z_{ikh}) \right), \quad \forall j, h > 0, \forall r, k : r \neq k \quad (10)$$

$$B_h \geq B_j + \sum_{i=1}^m p_{ij} y_{ij} - M \left(3 - \sum_{i=1}^m (x_{ijh} + z_{irj} + z_{ikh}) \right), \quad \forall j, h > 0, \forall r, k : r = k \quad (11)$$

$$\sum_{i=1}^m \sum_{k=1}^p z_{ikj} = 1, \quad \forall j > 0, \quad (12)$$

$$\sum_{i=1}^m z_{ikj} \leq f_{kj}, \quad \forall j, k \quad (13)$$

$$B_j + p_{ij} y_{ij} + s_{irk} K_{ihkr} \leq B_h + M \left(3 - \sum_{\substack{t=1 \\ t \neq i}}^m z_{tkh} - z_{ikj} - g_{jh} \right), \quad \forall j, h, i, k : j > h > 0 \quad (14)$$

$$B_h + \sum_{\substack{t=1 \\ t \neq i}}^m p_{th} y_{th} + \sum_{\substack{t=1 \\ t \neq i}}^m s_{trk} K_{tjkr} \leq B_j + M \left(2 - \sum_{\substack{t=1 \\ t \neq i}}^m z_{tkh} - z_{ikj} + g_{jh} \right), \quad \forall j, h, i, k : j > h > 0 \quad (15)$$

$$B_j \geq r_j, \quad \forall j \quad (16)$$

$$z_{ikj} \leq c_{ik}, \quad \forall i, j, k \quad (17)$$

$$\sum_{k=1}^p z_{ikj} \leq M y_{ij}, \quad \forall i, j \quad (18)$$

$$z_{i\mu_i 0} = 1, \quad \forall i \quad (19)$$

$$\sum_{j=1}^n x_{i0j} = 1, \quad \forall i > 0 \quad (20)$$

$$\sum_{j=1}^n x_{ij0} = 1, \quad \forall i > 0 \quad (21)$$

$$\sum_{j=1}^n y_{ij} \leq M u_i, \quad \forall i \quad (22)$$

$$K_{ijk} \leq \sum_{i=1}^m z_{ikj}, \quad \forall j, k, r : j > 0 \quad (23)$$

$$\sum_{i=1}^m \sum_{r=1}^k K_{ijk} \leq 1, \quad \forall j, k : j > 0 \quad (24)$$

$$K_{ijk} \geq x_{ihj} + z_{irh} + z_{ikh} - 2, \quad \forall i, j, h, k, r : j > 0 \quad (25)$$

$$z_{ikj}, y_{ij}, x_{ijh}, g_{jh}, K_{ijk}, u_i \in \{0, 1\}, \quad \forall i, j, k, h \quad (26)$$

$$B_j, T_j \geq 0 \quad \forall j \quad (27)$$

In this model, Equation (1) is the objective function which minimizes the total weighted tardiness. Constraint (2) calculates the tardiness values for each job by subtracting the due date of the job from its completion time. Constraint (3) assigns each job to one machine. Constraint (4) identifies the time that a machine can start processing the first job of the day depending on the jobs that remain on the machine from the previous day and the necessary setup time. At each machine, an auxiliary job is assigned to the first and last positions. This auxiliary job is numbered as “0” and assigned to all machines by Constraint (5). If a job is not assigned to a machine, it cannot be the successor or predecessor of a job on this machine. Constraints (6) and (7) satisfies these two conditions, respectively. If a job other than job 0 is assigned to a machine, then this job must have a predecessor as satisfied by Constraint (8). Similarly, if a job other than job 0 is assigned to a machine, it must have a successor as satisfied by Constraint (9). If job h follows job j on a machine and if these two jobs use different molds, the starting time of job h must be later than the starting time of job j plus the necessary mold switchover time. This is satisfied by Constraint (10). Constraint (11) satisfies the same restriction for two jobs that are assigned to the same machine and that use the same mold. Note that, in this case there is no setup time. Constraint (12) satisfies that each job must be processed on a machine by using a mold. Constraint (13) satisfies the machine eligibility conditions. Each job can be assigned to a machine that can process that job. If two jobs that use the same mold are assigned to two different machines, then they cannot be processed at the same time. One of them must be completed earlier than the other. Starting time of the latter job must be later than the necessary mold removal and installation time after the completion time of the earlier job. Constraints (14) and (15) satisfy this restriction. A job cannot start before its release time as indicated by Constraint (16). A mold can only be used with the appropriate machines as satisfied by Constraint (17). If a job is being processed by a mold on a machine, then that is assigned to that machine by setting the value of the assignment variable to 1 as in Constraint (18). Constraint (19) defines the initially attached molds on the machines so that the necessary setup times can be calculated correctly. Constraints (20) and (21) indicates that the auxiliary job can be the predecessor and successor of exactly one job on each machine. Constraint (22) does not assign any operation to a machine which is not used. Constraint (23) satisfies the relation between K and z variables. Constraints (24) and (25) satisfies the mold sequence on the machines. Constraints (26) and (27) defines the binary variables and the sign restrictions.

3.2 Heuristic Algorithm

Due to the complexity of the problem, solving a medium or even a small size instance with the developed MIP formulation takes an extensive computational time. Therefore, we developed a Genetic Algorithm (GA) to solve this problem. In the developed GA, the solutions are encoded as a matrix with three rows and n columns where n is the number of jobs. Table 1 provides an example for an individual with 10 jobs:

Table 1. Individual Representation in GA

1	2	3	4	5	6	7	8	9	10
M1	M2	M2	M3	M1	M2	M4	M2	M3	M4
K2	K3	K2	K2	K3	K4	K4	K5	K1	K3

In this encoding, the first row represents the jobs, the second row represents the assigned machine, and the third row represents the assigned mold. This job-machine-mold assignment is named a work package. In the given example, Job 1 is assigned to machine 1 using mold 2 whereas job 9 is assigned to machine 3 using mold 1.

Steps of the Genetic Algorithm

1. Generate the initial population as described below in more detail. For each individual, schedule the work packages on the machines and calculate its fitness value.
2. Generate the next population:
 - 2.1. Select parents randomly and perform crossover. For each new individual, schedule the jobs and calculate the fitness values.
 - 2.2. Perform mutation with certain probability on the generated children and move the fittest children to the next generation.
3. Repeat Step 2 until the specified stopping criteria is met.

Generation of the Initial Population

1. For each job: Select randomly one of the molds that can be used with this job.
2. For each job-mold assignment: Select randomly one of the machines that can be used with this assignment.
3. Schedule the work packages on the machines and calculate the fitness value.
4. Repeat Step 1-3 until the required population size is reached.

Crossover

1. Calculate the total fitness value of the individual in the population, $\sum_j f_j$.
2. For each individual calculate the selection probability using the following:

$$p_i = \frac{f_i^{-1}}{\sum_j f_j^{-1}}$$

3. Rank the individuals in ascending order of the calculated probability value.
4. Using roulette wheel selection method, select a pair of individuals to be parents.
5. Generate ten children using the following procedures:
 - 5.1. Generate a random number between 1 and n to identify the cutoff point.
 - 5.2. Using the cutoff point, crossover the machine and mold rows to generate two children (Child 1 and 2).
 - 5.3. Using the cutoff point, do the crossover only for the mold row (excluding the machine row) and generate two children. For each mold, assign one of the machines randomly that appeared in the parents for that mold (Child 3 and 4).
 - 5.4. Repeat Step 5.3 by using all possible machines that appear in the parents for that mold and select one of the machines randomly (Child 5 and 6).
 - 5.5. Generate two equally spaced cutoff points and divide the genes into three parts. Generate two children (Child 7 and 8) by selecting the midpoint from one of the parents and two sides from the other parent.
 - 5.6. Generate two children (Child 9 and 10) by selecting one column from one parent and the consecutive column from the other parent interchangeably.
6. Considering the two parents and the newly generated children, select the two individuals with the best fitness values and move to the next generation.
7. Repeat Steps 4-6 until the population size in the next generation is satisfied.

4. Computational Study

In this section, we test the performances of the MIP formulation and the GA computationally. The purpose of these tests is to investigate the solution time of the MIP formulation and to show that the solutions developed by the heuristic algorithms are of a certain quality (low margin of error) and they derive these solutions within a reasonable time. Therefore, the tests were carried out to measure both the solution quality and the solution time.

The MIP formulation is implemented in Java programming language and solved by CPLEX 22.1 solver using Java libraries. The GA is also coded in Java language.

We generated test instances to represent realistic industrial settings of the problem under consideration. These instances vary in the number of machines, jobs, and molds, and incorporate diverse processing and setup times. The test problems are generated randomly such that they are guaranteed to have feasible solutions. The initial busy time of the machines and molds, the processing time, and mold switchover times are randomly generated. We generated different problems with the following problem sizes:

- 8 jobs, 2 machines, 5 molds
- 8 jobs, 2 machines, 10 molds
- 15 jobs, 5 machines, 10 molds
- 15 jobs, 7 machines, 25 molds
- 20 jobs, 2 machines, 10 molds
- 20 jobs, 3 machines, 5 molds

Performing the computational study enables us to comment about the average and worst case performance of the heuristic algorithm. While measuring performance, the criteria of "solution quality" (i.e. how much deviation from optimal) and "solution time" are taken into account. Solution time measures the elapsed computational time (seconds, minutes, etc.) the algorithm spends to derive a solution. The amount of deviation of the obtained solution from the optimal solution is expressed as a percentage value. For this purpose, if the objective function value of the heuristic

algorithm is represented by H and the optimal objective function value is represented by O , the percent deviation value, Gap , is calculated as follows:

$$Gap = \frac{H-O}{O} \times 100 \quad (28)$$

The MIP formulation is solved with 1-hour time limit and the best solution is recorded.

5. Results and Discussion

From each problem size, five random problem instances are generated. In total, 30 problem instances are generated. Table 2 summarizes the results of the computational study.

Table 2. Run Results for the MIP Formulation and the Heuristic Algorithm

Job/Machine/Mold	MIP			GA	
	Best Integer	Relative Gap (%)	CPU Time (s)	Gap (%)	CPU Time (s)
8/2/5	252	0.0	4.5	0.0	485.8
8/2/5	1,286	0.0	73.5	0.0	483.0
8/2/5	229	0.0	1.5	0.0	482.1
8/2/5	857	0.0	10.0	0.0	482.8
8/2/5	935	0.0	144.5	0.0	482.8
8/2/10	471	88.5	TL	0.0	482.7
8/2/10	693	60.8	TL	0.0	482.3
8/2/10	549	0.0	67.2	0.0	482.6
8/2/10	1,765	38.0	TL*	0.0	483.4
8/2/10	91	100.0	TL	0.0	482.4
15/5/10	344	82.3	TL	0.0	906.5
15/5/10	481	90.9	TL	-12.6	910.6
15/5/10	427	84.5	TL	0.0	912.0
15/5/10	500	57.9	TL	0.0	905.9
15/5/10	655	81.4	TL	2.1	907.9
15/7/25	289	81.7	TL	-11.9	905.4
15/7/25	1,043	97.4	TL	-17.6	905.1
15/7/25	1,645	80.6	TL	-11.7	907.2
15/7/25	1,113	58.6	TL	0.2	909.5
15/7/25	1,307	89.9	TL	-35.2	907.5
20/2/10	1,766	94.1	TL	-4.1	1,209.4
20/2/10	1,326	82.7	TL	-0.5	1,209.0
20/2/10	3,107	92.7	TL	-7.8	1,211.9
20/2/10	481	95.2	TL	-7.5	1,212.5
20/2/10	663	73.6	TL	-11.6	1,211.2
20/3/5	1,956	96.3	TL	0.0	1,208.9
20/3/5	414	100.0	TL	-46.4	1,238.0
20/3/5	1,749	97.6	TL	0.0	1,216.7
20/3/5	114	100.0	TL	0.0	1,231.6
20/3/5	3,847	91.2	TL	-7.6	1,206.9

As it can be seen, the MIP formulation stopped by time limit (TL) in most of the instances. Only in six out of the 30 instances it could find the optimal solution. Moreover, for the instances that it stopped due to time limit, the relative gap (the percent deviation from the lower bound) is very large. On the other hand, the GA found the same or better

solutions than the best known MIP solution in all but two instances. Since most of the MIP solutions are not optimal due to time limits, negative gaps indicate improvement over the best known MIP solutions. The gaps in the remaining two instances are 2.1% and 0.2%. Furthermore, the GA was able to find up to 46.4% better results than the MIP formulation. The run time of the heuristic for the largest instances is around 1200 seconds, which is a reasonable time for the considered complex problem.

6. Conclusion

This work addressed a new variant of unrelated parallel machine scheduling problem. The problem considered how to schedule interdependent jobs with sequence and machine-dependent setup times and resource constraints. The solution methodologies are proposed to solve this problem namely, MIP formulation and GA. Computational tests revealed the superiority of the developed GA in achieving very good solution quality within a reasonable solution time compared to the MIP solution, which usually reaches the computational time limit due to the complex nature of this problem. There is ongoing work to expand the test cases used for this problem as well as introduce new heuristic methods such as Ant Colony and Simulated Annealing and compare their overall performances.

Acknowledgements

Authors would like to acknowledge Sultan Qaboos University for providing necessary financial support to conduct this research work through grant number: IG/ENG/MIED/20/02.

References

- Afzalirad, M., Rezaeian, J., Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions. *Computers & Industrial Engineering*, 98. 2016. <https://doi.org/10.1016/j.cie.2016.05.020>
- Bektur, G., Saraç, T., A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server. *Computers & Operations Research* 103, 46–63. 2019. <https://doi.org/10.1016/j.cor.2018.10.010>
- Chen, C., Fathi, M., Khakifirooz, M., Wu, K., Hybrid tabu search algorithm for unrelated parallel machine scheduling in semiconductor fabs with setup times, job release, and expired times. *Computers & Industrial Engineering*, 165. 2022. <https://doi.org/10.1016/j.cie.2021.107915>
- Ekici, A., Elyasi, M., Özener, O.Ö., Sarıkaya, M.B., An application of unrelated parallel machine scheduling with sequence-dependent setups at Vestel Electronics. *Computers & Operations Research* 111. 2019. <https://doi.org/10.1016/j.cor.2019.06.007>
- Fang, W., Zhu, H., Mei, YHybrid meta-heuristics for the unrelated parallel machine scheduling problem with setup times. *Knowledge Based Systems* 241, 108193. ., 2022. <https://doi.org/10.1016/J.KNOSYS.2022.108193>
- Fanjul-Peyro, L., Models and an exact method for the Unrelated Parallel Machine scheduling problem with setups and resources. *Expert Systems with Applications*: X 5, 100022. 2020.<https://doi.org/10.1016/J.ESWAX.2020.100022>
- Fanjul-Peyro, L., Perea, F., Ruiz, R., Models and matheuristics for the unrelated parallel machine scheduling problem with additional resources. *European Journal of Operational Research* 260, 93. 2017.<https://doi.org/10.1016/j.ejor.2017.01.002>
- Framinan, J.M., Leisten, R., García, R.R., Manufacturing scheduling systems: An integrated view on models, methods and tools, *Manufacturing Scheduling Systems: An Integrated View on Models, Methods and Tools*. 2014.<https://doi.org/10.1007/978-1-4471-6272-8>
- Perez-Gonzalez, P., Fernandez-Viagas, V., Zamora García, M., Framinan, J.M., Constructive heuristics for the unrelated parallel machines scheduling problem with machine eligibility and setup times. *Computers & Industrial Engineering* 131, 131–145. 2019. <https://doi.org/10.1016/j.cie.2019.03.034>
- Vallada, E., Villa, F., Fanjul-Peyro, L., Enriched metaheuristics for the resource constrained unrelated parallel machine scheduling problem. *Computers & Operations Research* 111. 2019. <https://doi.org/10.1016/j.cor.2019.07.016>
- Villa, F., Vallada, E., Fanjul-Peyro, L., Heuristic algorithms for the unrelated parallel machine scheduling problem with one scarce additional resource. *Expert Systems with Application* 93. 2018. <https://doi.org/10.1016/j.eswa.2017.09.054>
- Yepes-Borrero, J.C., Perea, F., Ruiz, R., Villa, F., Bi-objective parallel machine scheduling with additional resources during setups. *European Journal of Operational Research*, 292. 2021.<https://doi.org/10.1016/j.ejor.2020.10.052>

Biographies

Hakan Gultekin is an Associate Professor at the Department of Mechanical and Industrial Engineering, Sultan Qaboos University, Muscat, Oman. He earned his Ph.D. in Industrial Engineering from Bilkent University in Türkiye. His research interests include scheduling, optimization modeling, and exact and heuristic algorithm development, especially for problems in modern manufacturing systems, energy systems, transportation and logistics, and wireless sensor networks.

Nasr Al-Hinai is an Associate Professor and Head of the Department of Mechanical and Industrial Engineering at Sultan Qaboos University, Oman. He earned his Ph.D. in Industrial Engineering from the University of Manitoba, Canada. His research interests encompass sustainable manufacturing, supply chain optimization, circular economy applications, and advanced materials engineering. Dr. Al-Hinai has received several awards, including Best Teacher and Best Paper awards, in recognition of his academic excellence and scholarly contributions. He is actively involved in organizing international conferences and has served in multiple leadership and chairing roles within the academic and professional community.