Scheduling Allocation Model for Unrelated Parallel Machine Considering Multi-task Simultaneous Supervision Dual Resources-Constraints to Minimize Makespan

Akbar Rugova[1*], Sukoyo[2], Muhammad Akbar[2]
[1,2)] Faculty of Industrial Technology, Master Program in Industrial Engineering and Management Departement, Bandung Institute of Technology
Jl. Ganesha 10, Lb. Siliwangi, Coblong, Bandung, 40116, Indonesia
Email: 23422050@mahasiswa.itb.ac.id*, sukoyo@itb.ac.id, muhammad@itb.ac.id

ABSTRAK

Untuk memaksimalkan produksinya, banyak perusahaan menginvestasikan dananya ke dalam mesin semi-otomatis untuk membantu mereka mencapai potensi maksimalnya. Seringkali, lantai pabrik masih belum optimal karena sistem penjadwalan tidak sesuai dengan mesin yang baru dibeli. Multi-task simultaneous supervision dual-resources constrained (MTSSDRC) menjadi metode dapat menyesuaikan alokasi penjadwalan antara operator dan mesin semi-otomatis baru. Meskipun demikian, masih terdapat celah pada bidang MTSSDRC yang melibatkan Unrelated Parallel Machine Scheduling Problem (UPMSP) yang menunjukkan bahwa waktu pemesinan tidak berkorelasi dengan pekerjaan yang dilakukan atau mesin yang dikerjakan. Penelitian ini mengembangkan model MTSSDRC versi baru yang sejalan dengan UPMSP yang diabaikan oleh penelitian sebelumnya. Menggunakan Gurobi untuk menguji model baru dalam berbagai kasus untuk memastikan bahwa model tersebut dapat diterapkan pada lantai produksi nyata yang menghadapi masalah tersebut. Model ini berhasil mencapai solusi optimal global dan menghasilkan alokasi penjadwalan yang optimal dalam beberapa subkasus yang sangat kecil hingga kecil dan diperlukan kemajuan lebih lanjut dalam metodologi solusi untuk subkasus yang lebih kompleks. Penelitian ini juga membawa MTSSDRC lebih dekat dengan mitranya di kehidupan nyata sebagai lantai pabrik yang ada di banyak industri manufaktur yang memiliki kemampuan pemesinan berbeda-beda.

Kata kunci: *MTSSDRC, UPMSP, alokasi penjadwalan*

ABSTRACT

To maximize their production, many companies invest their funds into a semi-automatic machine to help them reach their fullest potential. The shop floor is still not optimized because the scheduling system does not fit the newly bought machines. Multi-task simultaneous supervision dual resource constraints (MTSSDRC) have been the go-to method to adjust the scheduling allocation between operator and the new semi-automatic machines. Nonetheless, there is still an absence in MTSSDRC field that involves an Unrelated Parallel Machine Scheduling Problem (UPMSP) that indicates that machining time does not correlate to the job it did or the machine it worked on. This research developed a new MTSSDRC model that is in line with UPMSP which is overlooked by previous studies. The new model was tested using Gurobi in various cases to ensure that the model is able to be applied to the shop floor that encounters such problems. The model successfully reaches global optimal solution and produces an optimal scheduling allocation in every very-small subcases and some in small subcases. Further advancement in solution methodology is needed for the more complex subcases. This research brought MTSSDRC a more complex designation as a shop floor in many manufacturing industries have varying machining capabilities.

Keywords: *MTSSDRC, UPMSP, scheduling allocation*

1. Introduction

Resources are summed up as something finite that controls the performance and outcome of a shop floor. More often than not, operators and machines become the main resources that concern manufacturing companies because of their availability (Berti et al., 2021). But that doesn't stop researchers from finding the use of scheduling theory in another type of resource, such as a machine (Logendran et al., 2007), a worker (Andrade-Pineda et al., 2020), a masker (Chen et al., 2023), etc. It is important to allocate the right resources at the right time to maximize efficiency (Martins et al., 2020). Early in its development, scheduling research only considered one limited resource, selecting

*Email corresponding author: 23422050@mahasiswa.itb.ac.id, Tel.: +62-82269305090;

the most important one, such as those studied by Yang & Geunes (2007). However, in the field, it is not enough to include only one resource because the production floor is increasingly complex. This gave rise to the topic of the problem of Dual Resource Constraints (Eimaraghy, 2000).

Dual resource constrained (DRC) is a problem that limits the capacity scheduling model to two types of resources. These resources are usually workers and machines (Cunha et al., 2019). DRC is one of the constraints that is often used because basically, DRC focuses on scheduling which becomes more dependent on workers, machines, and unique attributes of resources to provide an optimal schedule (Treleven, 1989). DRC systems have more complexity than considering 1 resource and present many challenges that must be considered to achieve optimal scheduling (Zhu et al., 2023). Research related to dual resource constrained (DRC) was pioneered by Nelson (Nelson, 1968) has developed rapidly and has considered various machine characteristics using various cases such as cloud-computing (Aron & Abraham, 2022), the photolithography industry (Chen et al., 2023) etc. DRC theory continues to develop over time by considering various characteristics of machines and workers to obtain a model that suits the work conditions in each field. Because of its crucial function in describing conditions on the production floor, much research has continued to develop DRC until now.

Akbar & Irohara (2018a) solve the DRC problem by paying attention to sequence-dependent setup and unloading, moving, and simultaneous supervision to describe the state of a semi-automatic machine. That is because the development of technological advances in industry has given rise to semi-automatic machines, workers only need to do work before and after machining such as setup and unloading (Karabegović et al., 2016). The particular problem is called Multi-task Simultaneous Supervision Dual Resources Constrained (MTSSDRC). Although MTSSDRC research has developed rapidly since its emergence by Zhu et al. (2023) who researched MTSSDRC in a flexible job shop environment on independent robots, there is still a lot of ground that needs to be explored in MTSSDRC, especially in unrelated parallel machine environments. In the previous studies, the machine configuration that is being used as a reference is all a single-stage machine. Using a single-stage machine configuration helped simplify the model and make it easier to build around since this is fairly a new field of study in the scope of dual-resource-constrained (DRC) scheduling.

In manufacturing industries, decision-making is a critical thing to do the manufacturing and service industry (Pinedo, 1995). Several studies old and new such as those conducted by Ho & Lau (1992) and Martins et al. (2023) have shown that optimizing existing scheduling can minimize the costs incurred by manufacturing companies. Research regarding scheduling has its foundation on a single parallel machine. Single-machine scheduling is the core of research related to scheduling because this model can be used as a building block in completing more complex scheduling models (Koulamas & Kyparisis, 2023). At the beginning of its development, scheduling research involving more than one machine was still limited (Koulamas, 1994). Multiple machine scheduling theory is the study of constructing a machine schedule for a series of jobs to ensure their execution takes place at a reasonable time. Multiple-machine scheduling can be done in a series or parallel system. Even though they have their advantages, identical and different series machines have disadvantages, namely reducing the flexibility of the system (Cheng & Sin, 1990). Meanwhile, parallel machines provide the spotlight for scheduling n jobs on m machines (Van Den Akker et al., 1999). Each job is executed on a specific machine throughout the runtime. This scheduling aims to find a schedule that optimizes certain performance measurements according to the researcher's perspective (Mokotoff et al., 2001).

Unrelated parallel machines have been the focus of much scheduling research. Until 2022, 42% of scheduling research dabbled in this field of study. They are often used to represent real-world cases of what happened in many manufacturing industries (Berthier et al., 2022). Unrelated parallel machines are a situation where a series of parallel machines do not have identical production capacities and do not have an identifiable correlation at the level of comparison with other machines (Pinedo, 1995). Unrelated parallel machines have significant practical relevance and have existing applications in various fields of work (Lin & Huang, 2021). Unrelated conditions on parallel machines can be caused by the use of various machines causing varying workloads and machine fatigue conditions as well. One example of research on unrelated parallel machines is that carried out by Pfund et al. (2004) who surveyed unrelated parallel machine algorithms for single and multiple objectives. Lenstra & Vakhania (2023) examined the complexity of scheduling unrelated parallel machines with limited preemption, and others. Research on unrelated parallel machines has developed into various points of view, one of which is sustainable, carried out by Sanati et al. (2023) by considering sequence-dependent setup and electricity usage time. Research on unrelated parallel machines has touched the realm of health, such as that carried out by Bazargan-Lari et al. (2022) who aimed at optimizing production scheduling in manufacturing during the COVID-19 pandemic with attention to physical distancing. So much of this research shows the relevance of existing parallel unrelated machines.

MTSSDDRC can be found in manufacturing industries that implement parallel machine environments using semi-automatic machines. One of the establishments that did so is Company X which did business in the aeronautic manufacturing industry. They use semi-automatic machines to do various things, including the milling process for airplane components. Because the number of operators is less than the number of machines, operators need to have the ability to be flexible and agile to be able to supervise more than one machine at a time. Operators have to move from machine to machine so that they can perform certain tasks, in this case setting up the machines and unloading them after the machining process is over. Based on the observed situation, it is very clear that MTSSDRC is a present situation and if explored deeper, it can benefit industries such as Company X that faced the same situation.

With all the consideration above, we learn that there's a certain gap in the scheduling field of study that hasn't been explored yet, particularly the MTSSDRC scheduling problem. To fulfill the needed advancement of the newly proposed scheduling, this study aims to develop the new MTSSDRC model that fits the Unrelated Parallel Machine (UPM) environment. The difference between widely used identical parallel machines (IPM) and unrelated parallel machines (UPM) in the MTSSDRC problem is in the pool of data that can be accessed by the model. Machining time in IPM only depends on the job that is being worked on, meanwhile machining time in UPM needs to consider not only the job but also where the job is worked on and, in this context what machine did which job. Because of the different nature of the problem, the current model proposed by Akbar & Irohara (2018a, 2020b, 2020a) can't simply use the model to access the UPM pool of data. By modifying the machining time input of the current model, we can adjust the input needed for the model to be able to process the data correctly. Following those previous studies, this research also pays attention to the sequence-dependent setup and unloading, by that we mean Sequence-dependent setup and unloading in this research refers to the setup and unloading time timeline which changes if the job allocation also changes. Because of the allocation variation that can happen in UPM scheduling problems, the setup and unloading time is very dependent on that scheduling allocation. In all the tweaks and iterations of the newly proposed idea, there has been an absence of the incorporation of such a machine environment. The dual resource-constrained (DRC) scheduling problems that consider unrelated parallel machine environments are too few and far between, let alone the specific MTSSDRC.

2. Literature Review

Unrelated Parallel Machine Scheduling Problem (UPMSP)

The concept of Unrelated Parallel Machine Scheduling Problem (UPMSP) has been around for ages. Research regarding unrelated parallel machines will gradually benefit manufacturing industries. Not only that, the UPMSP model in a scheduling problem can cover all kinds of varying input data even if the real-life system is not always going to be unrelated. In short, it can input host data from identical, uniform, and unrelated parallel machines. UPMSP considers all job's machining times to be different from any other machines in the same parallel system. UPMSP replicates the condition in manufacturing fields where most of the machines are a complex system that has various types and operation speeds (Arroyo & Leung, 2017). Kumar et al. (2009) proposed a unified way to solve the UPMSP in the form of a single rounding algorithm on UPMSP. Recently, there has been a surge of research with UPMSP and DRC collaboration in mind. Avgerinos et al. (2023) did research about UPMSP on weighted tardiness minimization with resource-constrained setups. Many take different aspects of a shop floor to tweak and modify the base UPMSP. Pulansari & M. (2021) did research regarding UPMSP with the concern of setup duration that is dependent on one another and solved it using Ant Colony Optimization in an effort to minimize makespan and tardiness which in turn lowers the company's waste of resources. The previous study addressed UPMSP which involved the delivery times of a job from a machine that is dependent on one another and also eligibility concerns to minimize the total weighted tardiness (Maecker et al., 2023). Lei & Yang (2022) did a UPMSP study with the collaboration of preventive maintenance and sequence-dependent setup duration times which arrangement is dependent on each one of them. The result of the study is a multi-sub-colony artificial bee colony. The aim is to minimize the total tardiness and makespan. Fleszar & Hindi (2018) took UPMSP involving a renewable resource constraint. The result of the research is a newly proposed and efficient MILP model and a two-stage heuristic to solve the two-machine variant of the problem. However, it does not involve a multiple resource as a constraint which means that it is not considered a DRC system

One of the most recent iterations of UPMSP research in taking the DRC as a constraint is done by Chen et al. (2023) et al who did so in a Photolithography industry where a mask is a limited resource that has to be allocated efficiently. This is unique among the many UPMSP research because there's not one of them that explicitly mentioned a DRC problem in a UPMSP environment. Currently, no research satisfies the current predicament faced by many manufacturing industries that implement semi-automatic machines. With most of the workers in an unrelated parallel machine environment being static, only assigned to one machine to supervise at a time, this hinders the potential of a company that already invests in semi-automatic machines to reach its maximum productivity.

Multi-task Simultaneous Supervision Dual Resource Constraints (MTSSDRC)

Dual Resource-constrained (DRC) Scheduling is not a new field to be explored concerning scheduling problems. However, the advancement of machine technologies in manufacturing industries requires DRC to fit into the semi-automatic machine environment (Akbar & Irohara, 2020a). Costa et al. did a single-task simultaneous supervision DRC (STSSDRC) that allocated a variable that included the assigned operator, machine, current job, and preceding jobs on certain machines (Costa et al., 2013). Despite that, this only involves one single task which is a setup and different from the problem faced in this research. Previous research proposed using working modes to estimate processing times with multipliers which are defined into four stages for each operation: setup, control, machining, and unloading which the operator did (Baptiste et al., 2017).

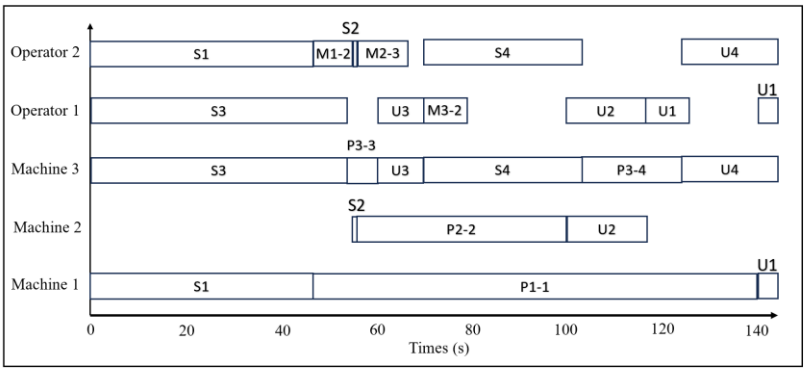
MTSSDRC evolved from the basis of a DRC system that considers simultaneous supervision and the ability of the operator to move from machine to machine. As highlighted, this is the first attempt any research tried to involve Dual Resource-Constrained (DRC) Scheduling with the ability of an operator to supervise one or more machines simultaneously. Because the machine is static in place, an operator needs the ability to move from machine to machine (Aron & Abraham, 2022). It also can be considered sustainable scheduling (Akbar & Irohara, 2018b). The MTSSDRC model was also altered to solve a double objective purpose using NSGA-II. After that, Akbar & Irohara brought the MTSSDRC model even further by testing it with various decoding schemes and metaheuristic algorithms. Despite all of that, none of the research that has been done considers the model in a different machine environment. This niche of the DRC field is increasingly developing, with more and more research that stems from it. Recently, research developed a DRC system with a robot as the main agent who carries out the production process to help solve the MTSSDRC problem in a flexible job shop. In handling a new subject or field, researchers usually start with the assumption that the machine environment is identical which makes it more simplistic and easy to understand compared to a UPM model, hence why the identical parallel machine model is more abundant than UPM. So, it is only natural for MTSSDRC to reach the UPMSP knowledge sphere. Table 1 summarizes the related studies in UPMSP and MTSSDRC.

Table 1. Related Studies

Literature	Machine Environment	DRC	Constraints				Objective Function
			Type of Constraint	Simultaneous Supervision	Task	Moving	
Costa et al. (2013)	Identical PM	√	Worker s & Machine ($w < m$)	√	Setup	-	Makespan
Baptiste et al. (2017)	Identical PM	-	Workers & Machine ($w < m$)	√	Loading, Setup, Controlling, Unloading	-	Makespan
Fleszar & Hindi (2018)	Unrelated PM	-	Machine	-	Setup and Unloading	-	Makespan
Akbar & Irohara (2020b)	Identical PM	√	Workers & Machine ($w < m$)	√	Setup and Unloading	√	Makespan & Produktifitas
Akbar & Irohara (2020a)	Identical PM	√	Workers & Machine ($w < m$)	√	Setup and Unloading	√	Makespan
Pulansari & M. (2021)	Unrelated PM	-	Machines	-	Setup	-	Makespan and Tardiness
Lei & Yang (2022)	Unrelated PM	-	Machine with PM	-	Setup and Unloading	-	Makespan
Chen et al. (2023)	Unrelated PM	√	Machine & Masker	-	Setup and Unloading	-	Total weighted completion time
Maecker et al. (2023)	Unrelated PM	-	Machine with eligibility	-	Setup	-	Total Weighted Tardiness
This Research	Unrelated PM	√	Workers & Machine ($w < m$)	√	Setup and Unloading	√	Makespan

3. Problem Statement

The MTSSDRC scheduling problem from the previous studies (Akbar & Irohara, 2018a, 2020b, 2020a) appoints a set of job J s to a set job of a bundle of operators W operating a set I of m semi-automatic machines, where $w < m$. For modeling purposes, a set J' that represents a dummy job j_0. Each job needs to complete setup, machining, and unloading, to be considered finished. Operators play a role in both setting up (a_s) and unloading (a_u) machines. To model this, we define a set of tasks as A. Every task (b) within a job (l) required a specific operating time (o_{bl}). A key aspect of the MTSSDRC model is that the operators need time (m_{hl}) to move between machines after a task is complete, this moving time is a significant factor. After a machine is set up, it will process job (l) for the duration of the machining process (p_{il}) time units. Finally, the linear programming model uses a large number (B) to manage the binary variables.



Sl = setup time for job l Ul = unloading time for job l
Pi-l = machining time in machine i for job l Mh-i = moving time from machine h to machine i

Figure 1. Illustration of MTSSDRC problem in UPMSP environment

Figure 1 shows the illustration of the MTSSDRC problem if it operates in a UPMP environment. Overall, it is quite similar to the previous studies, the number of operators is less than the number of machines, the existence of moving time because operators are allowed to wander around the shop floor to multi-task (setup and unloading), and supervise multiple machines and two resources that must be managed (machines and operators). The difference is that there is new notation in the machining time Pi-l in Figure 1 that needed to be addressed. In the UPM, it is important to know the identity of the machining process, where it takes place and what job do the machine works on. This is the notation that differs the this research to the previous studies where people only need the notation of Pl to identify machining times.

When brought to the UPMSP problem, the difference that it makes to the identical parallel machine is the pool of data UPMSP provided with than an identical one. Because of the larger pool of data, the will be more variation in mix-and-match because of the sheer variation of the outcome. The presented mathematical model is as follows.

Indices
$J = (j_1, j_2, \dots j_n)$ of n jobs
$K = (k_1, k_2, \dots k_w)$ of w operators
$I = (i_1, i_2, \dots i_m)$ of m semi-automatic machines
$J' = (j_0, j_1, \dots j_n)$ of n+1 jobs
$A = \{a_s, a_u\}$ of task setup a_s and unloading a_u that the operator does

Parameters
p_{il} = machining time of job l on machine i
o_{bl} = operation time for the task b of job l
m_{hl} = moving time of operator from the machine h to i
B = a big number

Decision variables
$x_{khajibl}$ = a variable that consists of binary number; 1 to indicate worker k is assigned to do activity b of job l on machine i after finishing activity a of job j on machine h
q_{fl} = a variable that consists of binary number, 1 to indicate setup activity of job l performs before unloading activity of job f in the same machine

o_{bl}^{c} = the completion time of activity b of job l
p_{bl}^{c} = the completion time of machining process for task b of job l
m_{bl}^{c} = the completion time of operator moving to perform task b of job l
c_{max} = the schedule's makespan

Model

$$minimize\ c_{max} \tag{1}$$

$$\sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'}\sum_{i\in I} x_{khajibl} = 1 \qquad \forall b \in A; l \in A \tag{2}$$

$$\sum_{k\in K}\sum_{h\in I}\sum_{i\in I}\sum_{b\in A}\sum_{l\in J} x_{khajibl} \le 1 \qquad \forall a \in A; j \in J \tag{3}$$

$$\sum_{h\in I}\sum_{i\in I}\sum_{b\in A}\sum_{l\in J} x_{kha_u j_0 ibl} \le 1 \qquad \forall k \in K \tag{4}$$

$$\sum_{k\in K}\sum_{h\in I}\sum_{i\in I}\sum_{b\in A}\sum_{l\in J} x_{kha_s j_0 ibl} = 0 \tag{5}$$

$$\sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'} x_{khajia_s l} - \sum_{q\in K}\sum_{g\in I}\sum_{c\in A}\sum_{f\in J'} x_{qgcfia_u l} = 0 \qquad \forall i \in I; l \in J \tag{6}$$

$$\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'} x_{khajibl} \ge \sum_{g\in I}\sum_{c\in A}\sum_{f\in J} x_{kiblgcf} \qquad \forall k \in K; i \in I; b \in A; l \in J \tag{7}$$

$$o_{bl}^{c} - m_{bl}^{c} \ge o_{bl} \qquad \forall b \in A; l \in J \tag{8}$$

$$m_{hl}^{c} - o_{al}^{c} \ge \sum_{h\in I}\sum_{i\in I}\sum_{k\in K} m_{hi} \times x_{khajibl} - B \times \left(1 - \sum_{h\in I}\sum_{i\in I}\sum_{k\in K} x_{khajibl}\right) \qquad \forall b \in A; \in J; a \in A; j \in J \tag{9}$$

$$o_{a_u l}^{c} - p_l^{c} \ge o_{a_u l} \qquad \forall l \in J \tag{10}$$

$$p_l^{c} - o_{a_s l}^{c} = \sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'}\sum_{i\in I} p_{il} \times x_{khajia_s l} \qquad \forall l \in J \tag{11}$$

$$\begin{cases} o_{a_s l}^{c} - o_{a_u f}^{c} \ge \sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'} o_{a_s l} \times x_{khjia_s f} \\ \qquad - B \times \left(2 - \sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'} (x_{khjia_s l} + x_{khjia_s f}) + q_{fl}\right) \\ o_{a_s l}^{c} - o_{a_u l}^{c} \ge \sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'} o_{a_s l} \times x_{khjia_s f} \\ \qquad - B \times \left(2 - \sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'} (x_{khjia_s l} + x_{khjia_s f}) + 1 - q_{fl}\right) \end{cases} \qquad \forall i \in I; f \in J; l > f \tag{12}$$

$$o_{a_u j_0}^{c} = 0 \tag{13}$$

$$c_{max} \ge o_{a_u l}^{c} \qquad \forall l \in J \tag{14}$$

$$x_{khajibl} \in \{0; 1\} \qquad \forall k \in K; h \in I; a \in A; j \in J'; i \in I; b \in A; l \in J \tag{15}$$

$$q_{fl} \in \{0; 1\} \qquad \forall f \in J; l > f \tag{16}$$

Eq. (1) aims to minimize makespan to complete all tasks. Eq. (2) ensures that each task is assigned by a single operator. Eq. (2) guarantees that each operator is assigned to do any job on one machine even after done only one processing the previous task of job. Eq. (3) ensures that the machine and operator follow a specific order within a job. Eq. (4) specifies that every worker begins by unloading a dummy job (j_0.). Eq. (5) prohibits assigning a setup task to the dummy job (j_0.). Eq. (6) restricts setup and unloading tasks for each job to a single machine. Eq. (6) restricts the setup task to one machine, the same goes for the unloading task. Eq. (7) precedence constraint assigned to every operator; task c of job f is contingent upon the prior completion of task b of job l, which itself necessitates the preceding execution of task a of job j. Eq. (8) states that operation time o_{bl} is equal to the minimum time lag between task b of job l completion time and the moving completion time of its preceding task Otherwise, Eq. (9) set a distinction between moving completion time of task b of job l and the corresponding precedence task a of job j on the minimum to be equal to the operator's moving time while performing each task. Eq. (10) arrange that the unloading time o_{aul} to be greater or equal than the minimum time lag between unloading completion time and machining task of job l. Meanwhile, Eq. (11) the machining task for all jobs launches straight away after the setup task. The twofold constraints on Eq. (12) defines the precedence constraint between jobs on machines: if a machine commences setup task of job l and unloading activity of job f, then the start of setup task of job l must be done after the unloading task of job f, also works the other way around. Eq. (13) defines the dummy job (j_0.) value as zero. Eq. (14) defines the maximum completion time across all jobs as the makespan. Last, Eq. (15) and Eq. (16) introduce binary variables for further decision-making within the model.

MTSSDRC in identical parallel machine environment does not have to elaborate job in machining time. Meanwhile, unrelated parallel machine environment allows the possibility of a job to have a completely random machining time depends on the machine that processed it. It is important for the input data of machining time to have two indices instead of one. That said, it needed to add a decision varable of x_{kha_sjibl} to select the chosen alocation and prevent the unwanted data to be included.

4. Result and Discussion

To represent a real case that happened on the shop floor, the presented new model of MTSSDRC on the UPM environment was tested in various cases to test the capabilities of the model. These cases are divided into 4 categories, very small, small medium, and large cases. In these cases, there are subcases that are segmented into 3 notations (*job* × *machine* × *operator*) to identify the complexity of each case. Not only that but to test the robustness of the model. This test is important to show the versatility of the model to process the uncertainty of real shop floor data. Input data used in the running of the model is randomly generated with the guide of the previous studies (Akbar & Irohara, 2018a, 2020b, 2020a). The computational result can be seen in Table 2.

Very-small subcase in this research starts with 4 jobs followed by 3 machines and 2 workers. This is shown to be in line with the constraint of the mathematical model where to replicate the situation in a semi-automatic manufacturing shop floor the number of operators must be less than the number of machines that exist and the operator has to do several different jobs simultaneously. Based on the Table it can be shown that in a small subcase, the overall Gurobi gap is 0%. This means the solution obtained is practically an optimal global solution. By definition, the Gurobi gap also known as MIPGap is a parameter that controls the minimal quality of the returned solution. It is an upper bound on the actual MIP gap of the final solution. It can happen that Gurobi runs until finding a solution that is significantly better than the MIP gap because it is not always possible to terminate with the given MIP gap requirement. In this research, the small subcases all have a Gurobi Gap value of 0%, which means that the solution provided by Gurobi Software most likely is the global optimal solution. Samples of the Gantt Chart in small subcases can be seen in Figure 3.

The same things can't be said for the small subcase. Out of the ten subcases, there are several that encountered an imaginary wall where a globally optimal solution can't be acquired. In the previous study done by Akbar & Irohara, a medium-sized subcase starts in the range of 9 jobs meanwhile in this paper small subcase starts with 6 jobs and a large subcase with 9 jobs. This adjustment is in response to the lack of transition of results that can be viewed between one subcase to another. In this case, a small subcase that all the subcases can be solved followed by a medium subcase that has several subcases that can't be completely solved with Gurobi and hit the time limit of 900 sec.

The large subcase that represented 100 jobs, 15 or more machines, and more than 7 operators showed that the Gurobi run-in Intel i5-12500H processor can't even start the calculation in the range of 15 minutes/900 sec. Because of that, there's not a single result that can be shown in any way shape, or form. In manufacturing industries, it is very common to have a large number of machines that have the same capabilities that have been operated in a parallel environment. This turns out to be a hurdle that this research cannot climb over because the nature of the problem is that have a considerable amount of input data in a compromise computation prowess. This was also confirmed by Akbar & Irohara who did an MTSSDRC in the identical parallel machine, that the Gurobi solver alone cannot produce a solution in a large subcase. Because of that, there is a need to use more than Gurobi to calculate such complex cases of scheduling.

In Figure 2, there is a massive difference in processing time when comparing one of the optimal solutions to the non-optimal solution acquired during simulation. In a very small subcase, it can be observed that there is an increase in the run time for a subcase with 2 operators than the same subcase with 3 operators. It can be seen that simplex iteration that a subcase with 2 operators has more than the same subcase with 3 operators. This difference in simplex iteration points out that a subcase with 2 operators has to mix and match more to find the optimal solution than the same subcase with 3 operators. This can be shown in Figure 2.

Table 2. Computational result

Case *(job × machine × operator)*	Gurobi Gap					Result					Run Time				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Very Small															
4 x 3 x 2	0%	0%	0%	0%	0%	136	111	144	121	129	0.37	0.61	0.93	0.81	0.18
4 x 4 x 2	0%	0%	0%	0%	0%	114	109	138	109	109	0.27	1.24	0.84	2.26	0.37
4 x 4 x 3	0%	0%	0%	0%	0%	102	82	138	84	74	0.25	0.47	0.25	0.55	0.54
5 x 3 x 2	0%	0%	0%	0%	0%	175	154	181	167	175	1.65	2	3.39	2.21	3.17
5 x 4 x 2	0%	0%	0%	0%	0%	153	151	160	150	128	2	15.67	6.56	19.45	2.1
5 x 4 x 3	0%	0%	0%	0%	0%	134	111	144	114	151	1.23	2.17	3.95	1.84	0.93
Small															
6 x 3 x 2	0%	0%	0%	0%	0%	230	213	226	212	220	19.56	20.08	8.27	19.74	9.47
6 x 4 x 2	0%	14%	0%	22%	0%	209	207	210	207	207	170.66	900	67.87	900	713.48
6 x 4 x 3	0%	0%	0%	0%	0%	170	147	160	145	146	15.14	3.73	11.41	9.25	11.26
6 x 5 x 2	33%	37%	20%	21%	36%	205	207	204	204	203	900	900	900	900	900
6 x 5 x 3	0%	0%	0%	0%	0%	140	142	142	140	139	16.93	26.9	13,98	137.21	34.53
7 x 3 x 2	0%	0%	0%	0%	0%	243	242	255	239	239	52.91	147	43,33	768.64	768.64
7 x 4 x 2	0%	38%	24%	39%	31%	236	237	236	237	235	900	900	900	900	900
7 x 4 x 3	0%	0%	0%	0%	31%	184	156	181	162	159	53.45	65.17	18.75	51.87	900
7 x 5 x 2	47%	46%	45%	39%	46%	238	236	235	237	235	900	900	900	900	900
7 x 5 x 3	0%	8%	0%	17%	14%	156	158	158	157	158	633.27	900	138.52	900	900
Medium															
9 x 4 x 2	53%	50%	46%	56%	55%	294	296	297	295	300	900	900	900	900	900
9 x 4 x 3	0%	4%	34%	16%	28%	216	206	220	200	201	543.19	900	900	900	900
9 x 5 x 2	56%	55%	59%	55%	60%	289	286	291	291	293	900	900	900	900	900
9 x 5 x 3	35%	28%	35%	33%	39%	194	196	201	194	196	900	900	900	900	900
10 x 4 x 2	56%	56%	57%	61%	61%	333	334	333	331	341	900	900	900	900	900
10 x 4 x 3	47%	36%	39%	44%	16%	299	229	231	233	228	900	900	900	900	900
10 x 5 x 2	60%	43%	64%	61%	64%	335	221	330	332	333	900	900	900	900	900
10 x 5 x 3	40%	61%	46%	42%	45%	222	332	224	223	235	900	900	900	900	900
Large (cannot be generated)															
100 x 15 x 7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
100 x 15 x 9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
100 x 20 x 7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
100 x 20 x 9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

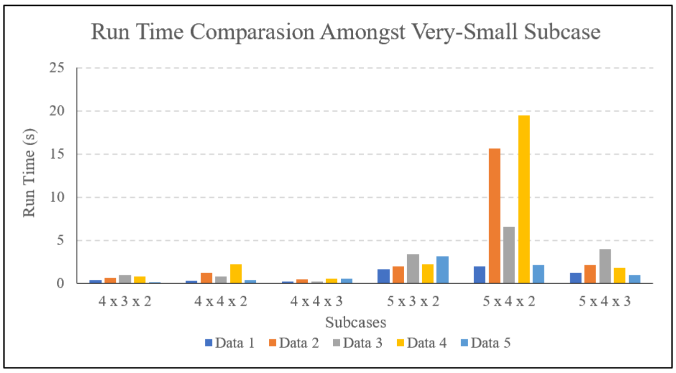


Figure 2. Comparison of run time amongst the very-small subcases

There seems to be no indication of prioritization whether a job with shorter job completion or longer job completion time. Also, it is seen that it is almost impossible to assign an operator to just one specific activity because all of the operators have to be able to perform setup and unloading. So homogenous type of operator is not desirable.

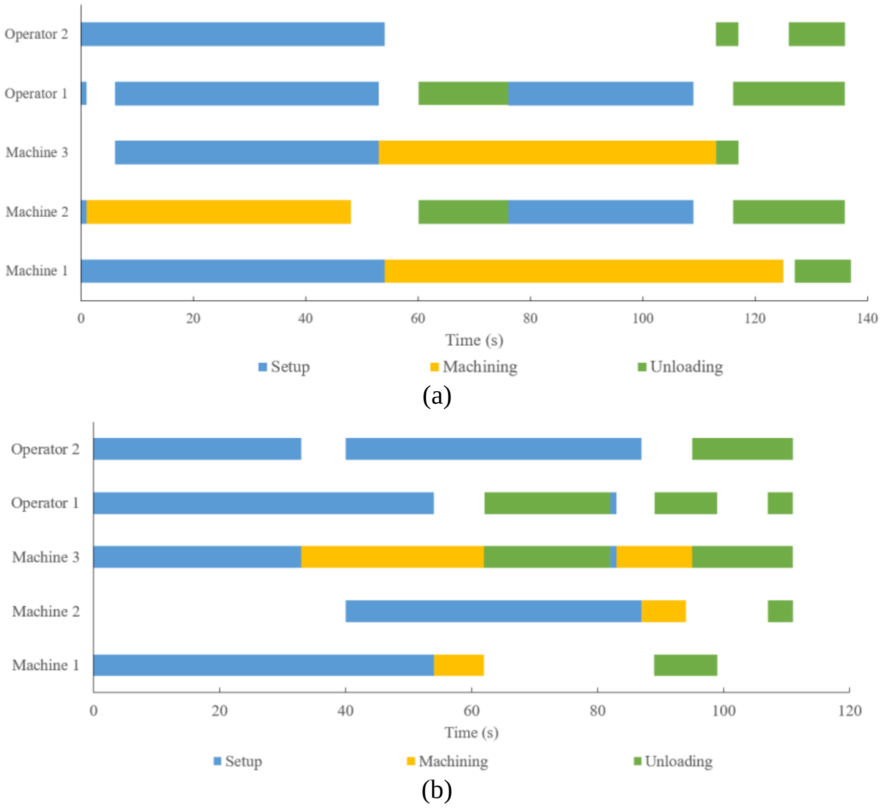
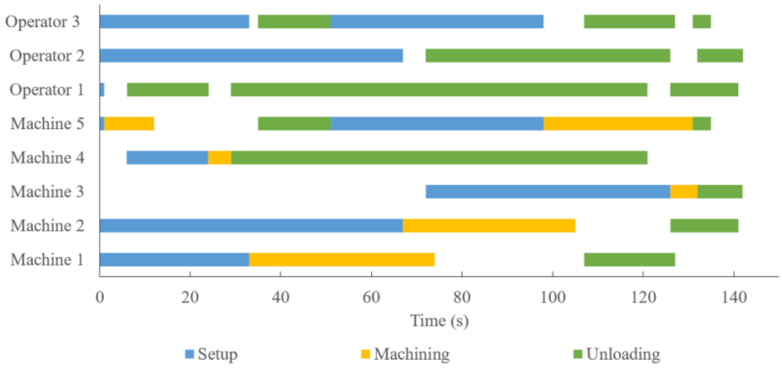
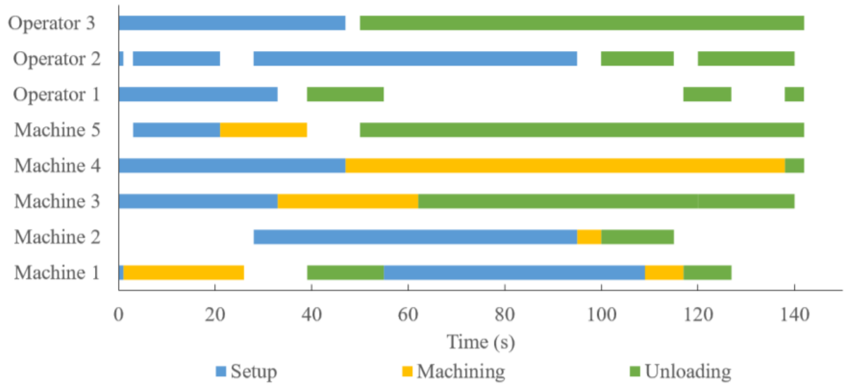


Figure 3. Samples of gantt chart for very-small subcase; (a) 4 × 3 × 2 iteration 1, (b) 4 × 3 × 2 iteration 2

The same can be said with the medium-sized subcase. With a larger pool of data to be analyzed and calculated and a varying machining time, the solution can be seen in the Gantt chart in Figure 4. This Gantt chart also represents the validity of the solution generated by Gurobi. Without interception of different jobs and activities, this chart shows that the solution acquired by the model in fact can become a working schedule for a shop floor with the same criteria as the subcase and constraints mentioned.



(a)



(b)

Figure 4. Samples of gantt chart for small subcase; (a) 6 × 5 × 3 iteration 1, (b) 6 × 5 × 3 iteration 2

Without a change in the mathematical model, it is impossible to input and process UPMS data. Because the machining time is a function of machine number and job number, it is required to adjust the model to accommodate such data. Adjusted model can also customized a schedule based on a machine with different capabilities and efficiency in case a manufacturing industry prefers to still use an older machine to save funds and maximize their productivity. Because the machine on the shop floor of a manufacturing industry is a complex system with many moving parts and various conditions that can alter the productivity of the machine, this model can generate a schedule based on that unpredictable condition which in turn makes life easier for the operator to design a optimal schedule.

Table 3. Comparison of subcase's result with and without time limit

Subcase	6x5x2 – With Time Limit	6x5x2 – Without Time Limit	10x4x3 – With Time Limit	10x4x3 – Without Time Limit
Simplex Iteration	489826	14235644	5277651	179718338
Gurobi Gap	32%	0%	47%	0%
Makespan	205	204	279	274
Run Time	900	36160	900	90171

The solution of some of the subcases tested in this paper did not reach the global optimal, as evidenced by the Gurobi Gap value that is above 0%. Some of the medium and large subcases were unable to search for an optimal solution within the time limit set in the Gurobi model of 900 seconds. Even so, the solution still can be obtained even though noted that it was not the optimal solution. The researcher compared one of the subcase's solutions to themselves if it was run without a time limit, as shown in Table 3. The result shows that the difference in run time between the subcase with a time limit and the one without is significant. Even so, the gap in the solution did not appear to be a massive difference. This shows that it is important to set the limit of the run time of the Gurobi model as it decreases the time needed to run. However, the time limit can alter the solution of the model. Some of the research surrounding the scheduling problem also faces the same problem and many of them choose to solve it with various methods such as metaheuristic which is nowadays one of the most popular solution methods used. Perhaps this is also the future that this research will be heading, using and possibly comparing various metaheuristic models to find the optimal solution in the shortest time possible.

Conclusion

It is important to adjust a model that specifically handles unrelated parallel machine environments as this is a form of effort to develop an already existing MTSSDRC model. This model not only can calculate the previous rendition of the model based on an identical parallel machine and uniform parallel machine but is also able to consider the variability and randomness of an unrelated parallel machine problem. This research also brought MTSSDRC closer to its real-life counterpart as a shop floor that exists in many manufacturing industries. Because the nature of the machine is a complex system, more often than not the machining processing time when a machine operates, the processing time can be different from job to job. Not only that but processing time from a different machine to the same set of jobs can be completely different. The randomness of input data makes unrelated parallel machine scheduling problems so unique and overall a robust model compared to identical or uniform parallel machines.

There's still a research gap that can be explored by future researchers. Because the analytical approach of the model in this research uses Gurobi, it cannot produce an optimal solution without a reasonable time. In the manufacturing industry, decision-making needs to be done fast and this limitation can hinder the production process. In the future, this research can be expanded by using various solution methods and tools. This research can also take in the different operator skills, various objective functions that some of the manufacturers prioritize more, etc.

References

1. Akbar, M., & Irohara, T. (2018a). Dual resource constrained scheduling considering operator working modes and moving in identical parallel machines using a permutation-based genetic algorithm. *IFIP Advances in Information and Communication Technology*, *535*, 464–472. https://doi.org/10.1007/978-3-319-99704-9_57
2. Akbar, M., & Irohara, T. (2018b). Scheduling for sustainable manufacturing: A review. In *Journal of Cleaner Production* (Vol. 205, pp. 866–883). Elsevier Ltd. https://doi.org/10.1016/j.jclepro.2018.09.100
3. Akbar, M., & Irohara, T. (2020a). Metaheuristics for the multi-task simultaneous supervision dual resource-constrained scheduling problem. *Engineering Applications of Artificial Intelligence*, *96*. https://doi.org/10.1016/j.engappai.2020.104004
4. Akbar, M., & Irohara, T. (2020b). NSGA-II variants for solving a social-conscious dual resource-constrained scheduling problem. *Expert Systems with Applications*, *162*. https://doi.org/10.1016/j.eswa.2020.113754
5. Andrade-Pineda, J. L., Canca, D., Gonzalez-R, P. L., & Calle, M. (2020). Scheduling a dual-resource flexible job shop with makespan and due date-related criteria. *Annals of Operations Research*, *291*(1–2), 5–35. https://doi.org/10.1007/s10479-019-03196-0
6. Aron, R., & Abraham, A. (2022). Resource scheduling methods for cloud computing environment: The role of meta-heuristics and artificial intelligence. In *Engineering Applications of Artificial Intelligence* (Vol. 116). Elsevier Ltd. https://doi.org/10.1016/j.engappai.2022.105345
7. Arroyo, J. E. C., & Leung, J. Y. T. (2017). Scheduling unrelated parallel batch processing machines with non-identical job sizes and unequal ready times. *Computers and Operations Research*, *78*, 117–128. https://doi.org/10.1016/j.cor.2016.08.015
8. Avgerinos, I., Mourtos, I., Vatikiotis, S., & Zois, G. (2023). *Weighted tardiness minimization for unrelated machines with sequence-dependent and resource-constrained setups*. http://arxiv.org/abs/2307.06671
9. Baptiste, P., Rebaine, D., & Zouba, M. (2017). FPTAS for the two identical parallel machine problem with a single operator under the free changing mode. *European Journal of Operational Research*, *256*(1), 55–61. https://doi.org/10.1016/j.ejor.2016.05.062
10. Bazargan-Lari, M. R., Taghipour, S., Zaretalab, A., & Sharifi, M. (2022). Production scheduling optimization for parallel machines subject to physical distancing due to COVID-19 pandemic. *Operations Management Research*, *15*(1–2), 503–527. https://doi.org/10.1007/s12063-021-00233-9
11. Berthier, A., Yalaoui, A., Chehade, H., Yalaoui, F., Amodeo, L., & Bouillot, C. (2022). Unrelated parallel machines scheduling with dependent setup times in textile industry. *Computers and Industrial Engineering*, *174*. https://doi.org/10.1016/j.cie.2022.108736
12. Berti, N., Finco, S., Battaïa, O., & Delorme, X. (2021). Ageing workforce effects in Dual-Resource Constrained job-shop scheduling. *International Journal of Production Economics*, *237*. https://doi.org/10.1016/j.ijpe.2021.108151
13. Chen, H., Guo, P., Jimenez, J., Dong, Z. S., & Cheng, W. (2023). Unrelated Parallel Machine Photolithography Scheduling Problem with Dual Resource Constraints. *IEEE Transactions on Semiconductor Manufacturing*, *36*(1), 100–112. https://doi.org/10.1109/TSM.2022.3232108
14. Cheng, T. C. E., & Sin, C. C. S. (1990). A state-of-the-art review of parallel-machine scheduling research. In *European Journal of Operational Research* (Vol. 47).
15. Costa, A., Cappadonna, F. A., & Fichera, S. (2013). A hybrid genetic algorithm for job sequencing and worker allocation in parallel unrelated machines with sequence-dependent setup times. *International Journal of Advanced Manufacturing Technology*, *69*(9–12), 2799–2817. https://doi.org/10.1007/s00170-013-5221-5
16. Cunha, M. M., Viegas, J. L., Martins, M. S. E., Coito, T., Costigliola, A., Figueiredo, J., Sousa, J. M. C., & Vieira, S. M. (2019). Dual resource constrained scheduling for quality control laboratories. *IFAC-PapersOnLine*, *52*(13), 1421–1426. https://doi.org/10.1016/j.ifacol.2019.11.398
17. Eimaraghy, H. (2000). Scheduling of Manufacturing Systems Under Dual-Resource Constraints Using Genetic Algorithms. In ~ *Journal of Manufacturing Systems* (Vol. 19, Issue 3).
18. Fleszar, K., & Hindi, K. S. (2018). Algorithms for the unrelated parallel machine scheduling problem with a resource constraint. *European Journal of Operational Research*, *271*(3), 839–848. https://doi.org/10.1016/j.ejor.2018.05.056

19. Ho, C.-J., & Lau, H.-S. (1992). Minimizing Total Cost in Scheduling Outpatient Appointments. *Management Science*, *38*(12), 1750–1764. https://doi.org/10.1287/mnsc.38.12.1750
20. Karabegović, I., Husak, E., Karabegović, E., & Mahmić, M. (2016). *The Role of Industrial Robots in the Development of Automotive Industry in China HOW THE CORE TECHNOLOGIES OF INDUSTRY 4.0 ARE CHANGING THE AUTOMOTIVE INDUSTRY IN THE WORLD, WITH A FOCUS ON CHINA*. https://www.researchgate.net/publication/312376111
21. Koulamas, C. (1994). Total tardiness problem: review and extensions. *Operations Research*, *42*(6), 1025–1041. https://doi.org/10.1287/opre.42.6.1025
22. Koulamas, C., & Kyparisis, G. J. (2023). A classification of dynamic programming formulations for offline deterministic single-machine scheduling problems. In *European Journal of Operational Research* (Vol. 305, Issue 3, pp. 999–1017). Elsevier B.V. https://doi.org/10.1016/j.ejor.2022.03.043
23. Kumar, V. S. A., Marathe, M. V., Parthasarathy, S., & Srinivasan, A. (2009). A unified approach to scheduling on unrelated parallel machines. *Journal of the ACM*, *56*(5). https://doi.org/10.1145/1552285.1552289
24. Lei, D., & Yang, H. (2022). Scheduling unrelated parallel machines with preventive maintenance and setup time: Multi-sub-colony artificial bee colony. *Applied Soft Computing*, *125*. https://doi.org/10.1016/j.asoc.2022.109154
25. Lenstra, J. K., & Vakhania, N. (2023). On the complexity of scheduling unrelated parallel machines with limited preemptions. *Operations Research Letters*, *51*(2), 187–189. https://doi.org/10.1016/j.orl.2023.02.004
26. Lin, D. Y., & Huang, T. Y. (2021). A hybrid metaheuristic for the unrelated parallel machine scheduling problem. *Mathematics*, *9*(7). https://doi.org/10.3390/math9070768
27. Logendran, R., McDonell, B., & Smucker, B. (2007). Scheduling unrelated parallel machines with sequence-dependent setups. *Computers and Operations Research*, *34*(11), 3420–3438. https://doi.org/10.1016/j.cor.2006.02.006
28. Maecker, S., Shen, L., & Mönch, L. (2023). Unrelated parallel machine scheduling with eligibility constraints and delivery times to minimize total weighted tardiness. *Computers and Operations Research*, *149*. https://doi.org/10.1016/j.cor.2022.105999
29. Martins, M. S. E., Viegas, J. L., Coito, T., Firme, B., Costigliola, A., Figueiredo, J., Vieira, S. M., & Sousa, J. M. C. (2023). Minimizing total completion time in large-sized pharmaceutical quality control scheduling. *Journal of Heuristics*, *29*(1), 177–206. https://doi.org/10.1007/s10732-023-09509-8
30. Martins, M. S. E., Viegas, J. L., Coito, T., Firme, B. M., Sousa, J. M. C., Figueredo, J., & Vieira, S. M. (2020). Reinforcement learning for dual-resource constrained scheduling. *IFAC-PapersOnLine*, *53*, 10810–10815. https://doi.org/10.1016/j.ifacol.2020.12.2866
31. Mokotoff, E., Jimeno, L., & Guti~rrez, A. I. (2001). *List Scheduling Algorithms to Minimize the Makespan on Identical Parallel Machines* (Vol. 9, Issue 2).
32. Nelson, R. T. (1968). Dual-Resource Constrained Series Service Systems. *Operations Research*, *16*(2), 324–341. https://doi.org/10.1287/opre.16.2.324
33. Pfund, M., Fowler, J. W., & Gupta, J. N. D. (2004). A survey of algorithms for single and multi-objective unrelated parallel-machine deterministic scheduling problems. *Journal of the Chinese Institute of Industrial Engineers*, *21*(3), 230–241. https://doi.org/10.1080/10170660409509404
34. Pinedo. (1995). *Scheduling: Theory, Algorithms, and System*.
35. Pulansari, F., & M., T. D. R. (2021). The Unrelated Parallel Machine Scheduling with a Dependent Time Setup using Ant Colony Optimization Algorithm. *Jurnal Teknik Industri*, *23*(1), 65–74. https://doi.org/10.9744/jti.23.1.65-74
36. Sanati, H., Moslehi, G., & Reisi-Nafchi, M. (2023). Unrelated parallel machine energy-efficient scheduling considering sequence-dependent setup times and time-of-use electricity tariffs. *EURO Journal on Computational Optimization*, *11*. https://doi.org/10.1016/j.ejco.2022.100052
37. Treleven, M. (1989). A review of the dual resource constrained system research. *IIE Transactions (Institute of Industrial Engineers)*, *21*(3), 279–287. https://doi.org/10.1080/07408178908966233
38. Van Den Akker, M., Hoogeveen, J. A., & Van Velde, S. L. D. E. (1999). Parallel machine scheduling by column generation ü. *Operations Research*, *47*(6), 862–872. https://doi.org/10.1287/opre.47.6.862
39. Yang, B., & Geunes, J. (2007). A single resource scheduling problem with job-selection flexibility, tardiness costs and controllable processing times. *Computers and Industrial Engineering*, *53*(3), 420–432. https://doi.org/10.1016/j.cie.2007.02.005
40. Zhu, X., Xu, J., Ge, J., Wang, Y., & Xie, Z. (2023). Multi-Task Multi-Agent Reinforcement Learning for Real-Time Scheduling of a Dual-Resource Flexible Job Shop with Robots. *Processes*, *11*(1). https://doi.org/10.3390/pr11010267