Jurnal Teknik Industri, Vol. 28, No. 2, December 2026
ISSN 1411-2485 print / ISSN 2087-7439 online
DOI: https://doi.org/10.9744/jti.28.2.164-175

A Mixed-Integer Programming Model for Skill-Aware Multi-Task Simultaneous Supervision Dual Resource-Constrained Scheduling on Unrelated Parallel Machines

Adi Candra[1*], Anas Ma'ruf[2], Muhammad Akbar[2]
[1)] Faculty of Industrial Technology, Doctoral Program in Industrial Engineering Department
Bandung Institute of Technology, Jl. Ganesha 10, Bandung 40132, Indonesia
[2)] Faculty of Industrial Technology, Industrial Engineering Department
Bandung Institute of Technology, Jl. Ganesha 10, Bandung 40132, Indonesia
Email: 33425011@mahasiswa.itb.ac.id*, maruf@itb.ac.id, muhammad@itb.ac.id
*Corresponding author

Abstract: In semi-automated manufacturing systems, operators are required to perform manual tasks while simultaneously supervising multiple unrelated parallel machines. Consequently, effective scheduling requires the integrated allocation of machines and operators as dual-capacitated resources. This study introduces a multi-task simultaneous supervision dual resource-constrained scheduling model for unrelated parallel machine environments. The proposed model incorporates machine heterogeneity, operator skill heterogeneity, and operator movement time between machines, with the objective of minimizing makespan. The scheduling problem is formulated as a mixed-integer linear programming model and solved using Gurobi Optimizer on synthetically generated benchmark instances. All six small instances were solved to optimality, while the medium instances produced feasible solutions within the specified time limit. Under the same average workforce skill, the heterogeneous scenario reduced makespan in all small instances by an average of 8.95%. The average reduction in true machine idle time was 22.29%, although the effect was not consistent across all instances. The results demonstrate that operator skill heterogeneity influences machine assignment, manual-operation sequencing, and workload distribution. The model generally assigns faster operators to critical setup and unloading activities, but this allocation may create workload imbalance. Therefore, practical implementation should consider the trade-off between production efficiency and equitable operator workload distribution.

Keywords: MTSSDRC, scheduling, unrelated parallel machines, operator skill, makespan.

Introduction

Dual resource-constrained scheduling refers to a scheduling problem in which machines and operators are positioned as two limited resources required simultaneously to execute production operations [1], [2]. The model must determine the assignment of jobs to machines as well as the assignment of available operators who satisfy the requirements for performing the operation [2], [3]. These decisions are interrelated, as a machine cannot begin a manual activity if the required operator is currently engaged in an activity on another machine. Therefore, machine and operator assignments need to be determined in an integrated manner, as demonstrated by Gong *et al*. through a memetic algorithm based on a genetic algorithm with a three-layer chromosome that simultaneously encodes operation sequencing, machine assignment, and worker assignment [4]. The characteristics of semi-automatic manufacturing systems distinguish the DRC problem from scheduling settings that require an operator to remain at the machine throughout the entire processing time. In such systems, operators are generally required only for specific manual activities, such as setup, unloading, inspection, and movement between machines, whereas the machining process can proceed without continuous operator attendance [5]. This condition allows a single operator to handle multiple machines, provided that the manual activities do not overlap with one another. The concept of simultaneous supervision has previously been represented through common-server models, in which a single operator performs both the loading and unloading operations of parallel machines while machining proceeds without operator attendance [6], as well as through operator working modes that regulate the allocation of operator capacity across multiple machines [7]. Akbar and Irohara formulated this condition as multi-task simultaneous supervision dual resource-constrained (MTSSDRC) scheduling [8]. The MTSSDRC problem must simultaneously coordinate job allocation

Submitted: August 13, 2026; Revised: Oct 08, 2026; Accepted: Oct 22, 2026

to machines, job sequencing, operator assignment, sequencing of manual activities, and operator travel time [8]. The interdependence among these decisions causes the computational time of exact models to increase sharply as the number of jobs grows, thereby necessitating metaheuristic approaches to solve larger-sized instances.

The development of MTSSDRC has focused not only on solution methods, but also on objective functions that represent production performance and workforce utilization. Akbar and Irohara developed an NSGA-II variant to minimize makespan and operator workload imbalance [9]. Subsequent research developed an MTSSDRC model with the objective of minimizing total tardiness and improving labor productivity, and compared several algorithms within the NSGA family [10]. The results of that study indicate that scheduling decisions in semi-automatic systems need to consider the interaction between time-based performance and workforce characteristics. The development of multi-objective DRCFJSP has likewise shown that makespan needs to be considered together with other resource performance indicators [11]. In dynamic environments, coordinating decisions related to jobs, machines, and additional resources likewise underpins real-time scheduling [12]. Most early MTSSDRC models were developed for identical parallel machines, assuming that the processing time of a job is the same across all machines. This assumption was subsequently relaxed by Larasati et al., who developed an MTSSDRC model for unrelated parallel machines, an environment in which the processing time of a job depends on the selected machine, with the objectives of minimizing total tardiness and balancing operator workload [13]. This study highlights the importance of accounting for differences in machine processing capability, as job allocation decisions can shift according to the job–machine pairings available. Incorporating additional resource constraints, release dates, and sequence-dependent setup times into the unrelated parallel machine environment broadens the scope of allocation and sequencing decisions, making metaheuristic approaches necessary to achieve favorable makespan outcomes [14].

In addition to differences in machine capability, the completion time of manual activities is also influenced by operator skill level. Rizka *et al.* developed an MTSSDRC model for identical parallel machines with heterogeneous operator skill levels, aiming to minimize the makespan and workload smoothness index [15]. In the flexible job shop environment, Wang *et al.* also demonstrated that differences in operator skills can affect setup time, operator selection, and scheduling outcomes [16]. Research on unrelated parallel machines with worker constraints and learning effects has also shown that worker and machine characteristics can interact in determining scheduling outcomes [17]. Simulation results from Costa *et al.* further confirm that workforce multifunctionality and heterogeneous labor efficiency affect the performance of DRC systems [18].

Vázquez-Serrano *et al.* developed an agent scheduling model for unrelated parallel machines considering sequence-dependent setup times and agent-machine assignments. The model integrates agent assignment, job allocation, and job sequencing to minimize makespan. However, agents are associated only with the execution of setup, and thus the model does not yet represent the MTSSDRC structure in which an operator handles both setup and unloading activities, including movement between machines [19]. The unrelated parallel machines model assumes homogeneous (undifferentiated) operator capability [13], whereas the model incorporating differences in operator skill level still relies on identical parallel machines [15]. Therefore, the research gap should not be construed as the claim that these two characteristics have never been considered at all, but rather as the absence of a makespan-oriented MTSSDRC formulation for unrelated parallel machines that explicitly links operator skill to setup and unloading durations and evaluates its impact through a comparison of differentiated and undifferentiated skill scenarios. The integration of these two characteristics is important because job completion time is shaped by an interrelated sequence of manual and machining activities. Setup activities initiate the processing cycle, while unloading closes the cycle and determines when the machine becomes available for the next job. If differences in operator skill are ignored, the model may produce inaccurate estimates of manual activity duration and operator assignment. Conversely, if differences in processing time across machines are ignored, the model may assign jobs to machines that are less efficient. The integration of these elements enables production managers to more accurately determine the combination of jobs, machines, and operators, as well as to identify whether delays in completion are attributable to manual activities, machine capabilities, or their interaction.

This study develops a mathematical MTSSDRC model for unrelated parallel machines that incorporates operator-skill heterogeneity and aims to minimize makespan. Accordingly, this study investigates whether explicitly representing operator-skill heterogeneity, while maintaining the same average workforce skill level, can reduce makespan compared with a homogeneous-skill representation. The contribution of this study is threefold. First, it extends the MTSSDRC formulation for unrelated parallel machines by explicitly incorporating operator-dependent setup and unloading durations under a makespan objective. Second, it presents

Table 1. Mapping of MTSSDRC scheduling problem characteristics

Author	Year	Machine environment	Opt. Skill Het-	Activities	Objective function	Primary focus / distinction
Costa *et al.*	2013	UPM	Yes	Setup	Makespan	Skill-dependent setup; no complete MTSSDRC structure
Akbar and Irohara	2020	IPM	No	Setup, unloading, and moving	Makespan	Identical machines; homogeneous skills
Akbar and Irohara	2020	IPM	No	Setup, unloading, and moving	Makespan and workload balance	Workload balance; homogeneous skills
Akbar and Irohara	2024	IPM	No	Setup, unloading, and moving	Total tardiness and labor productivity	Tardiness and productivity; homogeneous skills
Larasati *et al.*	2024	UPM	No	Setup, unloading, and moving	Total tardiness and workload balance	Tardiness and workload balance; homogeneous skills
Rizka *et al.*	2024	IPM	Yes	Setup, unloading, and moving	Makespan and WSI	Skill het-; identical machines
This Research	2026	UPM	Yes	Setup, unloading, and moving	Makespan	Joint unrelated-machine and skill-aware MTSSDRC formulation

an integrated MILP formulation that jointly determines job allocation, machine selection, operator assignment, activity sequencing, and operator movement. Third, it provides a controlled computational comparison of homogeneous- and heterogeneous-skill scenarios with the same average workforce skill level and evaluates their effects on makespan, machine idle time, and operator workload distribution across the tested instances.

Methods

This study uses a quantitative approach based on mathematical modeling to develop an MTSSDRC model for an unrelated parallel machine environment. The model was developed by considering the differences in job processing times across machines and differences in operator skill levels in performing manual activities. The objective function of the model is to minimize makespan, defined as the completion time of all jobs in the system. The research stages include identifying system characteristics, formulating assumptions, defining notation and parameters, developing the mathematical formulation, generating test instances, implementing the model, verification, internal validation, and result analysis. The model structure is developed based on the fundamental characteristics of MTSSDRC, namely that an operator is required only when performing manual activities, whereas the machining process proceeds automatically, allowing the operator to move and attend to other machines [6], [8]. The system considered in this study comprises a set of real jobs, job $J' = \{1, \dots, n\}$, a set of unrelated parallel machines $I = \{1, \dots, m\}$, and a set of operators $K = \{1, \dots, w\}$. A dummy job, indexed by j_0, is introduced to represent the beginning of an operator-activity sequence. Accordingly, the complete job set is defined as $(J' = J \cup \{j_0 = 0\}, J = \{1, , , n\})$. Each job must be allocated to exactly one machine and processed through the sequence of setup–machining–unloading activities. Loading is treated as part of the setup activity. Setup and unloading activities require one operator, whereas machining proceeds automatically without continuous operator attendance. The interaction among jobs, machines, and operators, as well as the sequence of activities in the developed MTSSDRC system, is shown in Figure 1.

The machine environment examined in this study is an unrelated parallel machine environment. Accordingly, the processing time of job l on machine i, denoted as p_{il}, may differ for each machine–job pair. This difference indicates that machines vary not only in general speed but may also exhibit different levels of suitability for each job. Therefore, machine selection and job sequencing decisions must be determined in an integrated manner [13]. Differences in operator skill affect only the duration of manual activities, namely setup and unloading. Machining time is not affected by operator skill, as this process proceeds automatically and its duration is determined by the machine–job pair. This approach follows the concept that variation in operator capability can lead to differences in the execution time of manual activities [15], [20]. The duration of manual activity b of job l when performed by operator k is defined as follows:

$$O_{kbl} = O_{bl}^{std} \eta_k, \forall k \in K, b \in A, l \in J \quad (1)$$

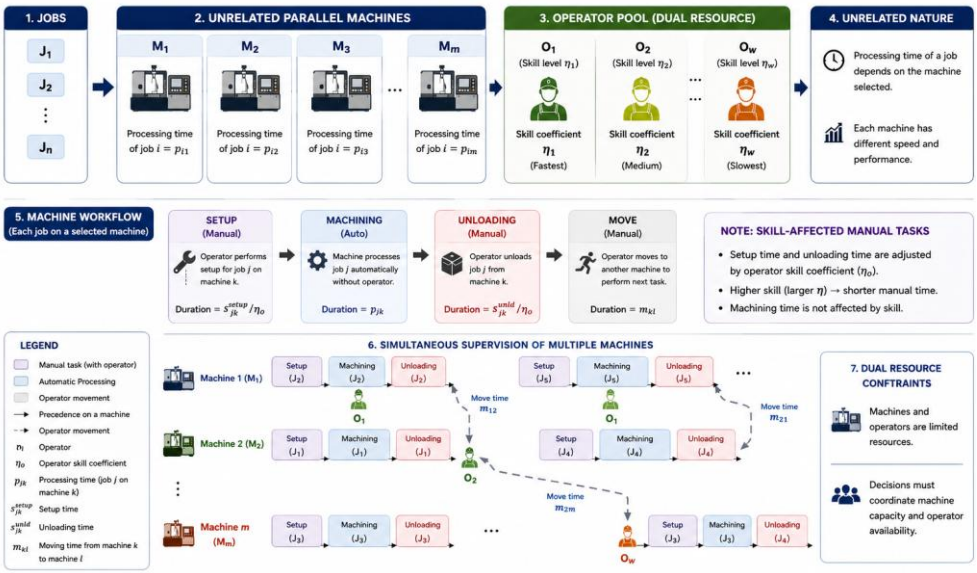


Figure 1. MTSSDRC system for unrelated parallel machines with skill heterogeneity

where:
O_{kbl}: actual duration of manual activity b of job l when performed by operator k.
O_{bl}^{std}: standard duration of manual activity b of job l.
η_k : skill coefficient of operator k.

The value of $\eta_k < 1$ indicates that the operator can complete the activity faster than the standard time, $\eta_k = 1$ represents a standard-skilled operator, and $\eta_k > 1$ indicates that the operator requires more time than the standard duration. The model was developed using a mixed-integer linear programming (MILP) formulation. This formulation is suitable because the MTSSDRC problem introduced by Akbar and Irohara [8] involves interrelated assignment and sequencing decisions together with time-related variables. Similar mathematical formulations have also been applied in subsequent MTSSDRC studies involving unrelated parallel machines and operator-skill differences [13], [15]. The decision variables include job allocation to machines, job sequencing on each machine, operator assignment to setup and unloading activities, the sequence of manual activities performed by each operator, as well as the start and completion times of each activity. Additional variables are used to represent operator movement between machines and to determine the system makespan value. The objective function of the model is formulated as follows:

$$\min C_{\max}$$

where $C_{\max}$ is the completion time of the last job. The relationship between makespan and the completion time of each job is formulated as follows:

$$C_{\max} \geq C_l, \ \forall l \in J'$$

C_l is the completion time of job l. This objective function directs the model to determine the combination of job allocation, machine selection, operator assignment, and activity sequencing that results in the lowest overall completion time. The model constraints are formulated to ensure that each job is allocated to exactly one machine, each manual activity is assigned to exactly one operator, and the setup–machining–unloading sequence is satisfied. The machine non-overlapping constraint ensures that two jobs are not processed simultaneously on the same machine. Meanwhile, the operator non-overlapping constraint ensures that an operator does not perform two manual activities simultaneously. The model explicitly considers operator movement time between machines. If operator k, after completing activity a of job j on machine h, subsequently performs activity b of job l on machine i the latter activity can begin only after the preceding activity has been completed and the movement time V_{hi} has elapsed. When $h = i$, the movement time is set to 0. This constraint prevents the generation of solutions that are mathematically feasible but operationally infeasible on the shop floor. The test instances were generated based on three problem-size dimensions: the number of jobs n, machines m, and operators w. Each test instance is denoted by $n \times m \times w$. The values for setup time, unloading time, processing time, operator moving time, and operator skill coefficient are generated within predetermined ranges. The parameter ranges used to generate the instances are reported below to support reproducibility.

The experiments considered two operator-skill scenarios. The first scenario assumed uniform skill levels, with $\eta_k = 1$ for all operators. The second scenario assumed heterogeneous skill levels, η_k with different values assigned to individual operators. These values were generated while maintaining an average skill coefficient of 1. This approach aims to ensure that differences in results between scenarios primarily represent the effect of skill variation or heterogeneity, rather than a change in average workforce capacity. Each pair of scenarios used the same fixed base data. Accordingly, the number of jobs, number of machines, processing time, standard time of manual activities, and moving time remained the same across both scenarios. The only difference was in the distribution of the η_k values. This configuration ensures fair comparisons of makespan values and scheduling decisions [21]. The MILP model was implemented in Python and solved using Gurobi Optimizer version 13.0.2. All computational experiments were conducted on a computer equipped with an Intel Core i7-3770S processor operating at 3.10 GHz and 16 GB of RAM. To ensure consistent computational settings, the target MIP gap was set to 0%, the Gurobi random seed was fixed at 0, and the maximum number of threads was set to eight for all test instances. The small instances were solved without a time limit, whereas a time limit of 600 seconds was imposed on the medium instances. For each run, the makespan, best bound, MIP gap, computational time, number of simplex iterations, number of branch-and-bound nodes, and solution status were recorded.

Model verification examined job and operator assignments, activity sequences, machine and operator non-overlap, movement-time compliance, and makespan consistency. Gantt charts were used to inspect the resulting schedules, while parameter-reduction scenarios were applied to assess model consistency. When all operators have a skill coefficient of $\eta_k = 1$, the constraint structure reduces to the MTSSDRC formulation for unrelated parallel machines with homogeneous operators [13]. Setting the processing time for each job uniformly across all machines reduces the model to the MTSSDRC case on identical parallel machines with operator skill-level differences, consistent with Rizka *et al.* [15]. Both conditions were used to assess the consistency of the developed model against established structures in prior literature. The results were analyzed by comparing uniform-skill and heterogeneous (varied-skill) scenarios across key parameters: makespan, job allocation, machine selection, operator assignment, service sequence, machine idle time, and operator idle time. Furthermore, changes in computation time and branch-and-bound node counts were evaluated to illustrate the computational impact of increased problem scale and the addition of operator skill variation.

Mathematical Models

The mathematical model in this study is based on the MTSSDRC concept introduced by Akbar and Irohara [8]. This study extends the work of Akbar and Irohara by adopting unrelated parallel machines, following Larasati *et al.* [13]. In their work, processing times are independent of machine assignment due to the use of identical machines. Here, processing times are explicitly machine dependent. Consequently, the mathematical model is extended to accommodate additional parameters, decision variables, and constraints. The model also incorporates operator skill heterogeneity, which affects setup and unloading times. The combined treatment of machine and operator heterogeneity provides a more realistic depiction of manufacturing systems. We formulate the problem as a MILP model aimed at minimizing makespan. The complete formulation is described as follows:

Sets

f, l	$\in$	J	:	$\{1,2,\dots,n\}$ real jobs
j	$\in$	J'	:	$J \cup \{j_0 = 0\}$ real jobs and dummy job
g, h, i	$\in$	I	:	$\{1,2,\dots,m\}$ machines
k, q	$\in$	K	:	$\{1,2,\dots,w\}$ workers
a, b, c	$\in$	A	:	$\{a_s, a_u\},\ a_s = setup, a_u = unloading$

Parameters

P_{il}	:	processing time of job l on machine i (Unrelated Parallel Machine)
V_{hi}	:	moving time from machine $h\ to\ i$
O_{bl}^{std}	:	standard time duration for completing task activity b of job l
M	:	a big number
η_k	:	skill coefficient of operator k

Decision variables

$X_{khajibl} \in \{0,1\}$	:	binary variable equal to 1 if operator k, after performing activity a of job$j$on machine h, subsequently performs activity b of job $l$on machine i ; 0 otherwise.

The indices k, h, a, j, i, b, and l represent the operator, preceding machine, preceding activity, preceding job, subsequent machine, subsequent activity, and subsequent job, respectively.

$Q_{fl} \in \{0,1\}$: binary variable equal to 1 if job l is scheduled before job f on the same machine; 0 otherwise.

$O_{bl}^{c} \geq 0$: completion time of manual activity b of job l

$P_{l}^{c} \geq 0$: machining completion time of job l

$V_{bl}^{c} \geq 0$: completion time of operator movement preceding activity b of job l

Notation		Notation	
J	real jobs	O_{bl}^{std}	standard duration of manual activity b of job l
J'	real jobs and dummy job	η_k	skill coefficient of operator k
I	set of machines	O_{kbl}	actual duration of manual activity b of job l when performed by operator k
K	set of operators	M	a sufficiently large positive constant used in the Big-M constraints
A	set of manual activities, $\{a_s, a_u\}$	$X_{khajibl}$	binary variable equal to 1 if operator k, after performing activity a of job j on machine h, subsequently performs activity b of job $l$on machine i; 0 otherwise
f, j, l	indices of jobs	Q_{fl}	binary variable equal to 1 if job $l$is scheduled before job $f$on the same machine; 0 otherwise
g, h, i	indices of machines	O_{bl}^{c}	completion time of manual activity $b$of job l
k, q	indices of operators	P_{l}^{c}	completion time of the machining operation of job l
a, b, c	indices of manual activities	V_{bl}^{c}	completion time of the operator movement preceding activity b of job l
a_s	setup activity	C_l	completion time of job l
a_u	unloading activity	$C_{\max}$	makespan, or the maximum completion time of all jobs
n	number of jobs	j_0	dummy job used to represent the initial node of an operator activity sequence
m	number of machines	O_{u0}^{c}	completion time of the unloading activity of the dummy job, fixed at zero
w	number of operators	$n \times m \times w$	instance configuration consisting of n jobs, m machines, and w operators
P_{il}	machining time of job l on machine i	$\%Gap$	relative difference between the best feasible solution and the best bound produced by the solver
V_{hi}	operator moving time from machine $h$to machine i	$U[\alpha, \beta]$	uniform distribution with lower bound $\alpha$and upper bound β
IPM	identical Parallel Machine		skill coefficient of operator k
UPM	unrelated Parallel Machine		

Objective function

$$\text{Minimize } C_{max} \tag{2}$$

Constraints

$$\sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'}\sum_{i\in I} X_{khajibl} = 1 \quad \forall b \in A; l \in J \tag{3}$$

$$\sum_{k\in K}\sum_{h\in I}\sum_{i\in I}\sum_{b\in A}\sum_{l\in J} X_{khajibl} \leq 1 \quad \forall a \in A; j \in J \tag{4}$$

$$\sum_{h\in I}\sum_{i\in I}\sum_{b\in A}\sum_{l\in J} X_{kha_u j_0 ibl} \leq 1 \quad \forall k \in K \tag{5}$$

$$\sum_{k\in K}\sum_{h\in I}\sum_{i\in I}\sum_{b\in A}\sum_{l\in J} X_{kha_s j_0 ibl} = 0 \tag{6}$$

$$\sum_{\boldsymbol{k\in K}}\sum_{\boldsymbol{h\in I}}\sum_{\boldsymbol{a\in A}}\sum_{\boldsymbol{j\in J'}} \boldsymbol{X_{khaja_s l}} - \sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'} X_{khaja_u l} = 0 \quad \forall i \in I; l \in J \tag{7}$$

$$\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'} X_{khajibl} \geq \sum_{g\in I}\sum_{c\in A}\sum_{f\in J} X_{kiblgcf} \quad \forall k \in K; i \in I; b \in A; l \in J \tag{8}$$

$$O_{bl}^{c} - V_{bl}^{c} \geq \sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'}\sum_{i\in I}\left(O_{bl}^{std} \cdot \eta_k\right) \cdot X_{khajibl} \quad \forall b \in A;\ l \in J \tag{9}$$

$$V_{bl}^{c} - O_{aj}^{c} \geq \sum_{h\in I}\sum_{i\in I}\sum_{k\in K} V_{hi} \cdot X_{khajibl} - M \times \left(1 - \sum_{h\in I}\sum_{i\in I}\sum_{k\in K} X_{khajibl}\right) \quad \forall b \in A; l \in J; a \in A; j \in J' \tag{10}$$

$$O_{a_u l}^{c} - P_{l}^{c} \geq \sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'}\sum_{i\in I}\left(O_{a_u l}^{std} \cdot \eta_k\right) \cdot X_{khaja_u l} \quad \forall l \in J \tag{11}$$

$$P_{l}^{c} - O_{a_s l}^{c} = \sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'}\sum_{i\in I} P_{il} \cdot X_{khaja_s l} \quad \forall l \in J \tag{12}$$

$$O^c_{a_s l} - O^c_{a_u f} \geq \sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'}\left(O^{std}_{a_s l}\cdot \eta_k\right)\times X_{khajia_s l} \quad (13)$$
$$-M\times\left(2-\sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'}\left(X_{khajia_s l}+X_{khajia_s f}\right)+Q_{fl}\right)$$
$$O^c_{a_s f} - O^c_{a_u l} \geq \sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'}\left(O^{std}_{a_s f}\cdot \eta_k\right)\times X_{khajia_s f}$$
$$-M\times\left(2-\sum_{k\in K}\sum_{h\in I}\sum_{a\in A}\sum_{j\in J'}\left(X_{khajia_s l}+X_{khajia_s f}\right)+1-Q_{fl}\right)$$
$$\forall i = 1,2,\dots,m;\ f = 1,2,\dots,n;\ l = f+1, f+2,\dots,n$$
$$O^c_{a_u j_0} = 0 \quad (14)$$
$$C_{max} \geq O^c_{a_u l} \quad \forall l \in J \quad (15)$$
$$X_{khajibl} \in \{0;1\},\ \forall k \in K; h,i \in I; a,b \in A;\ j \in J';\ l \in J \quad (16)$$
$$Q_{fl} \in \{0;1\},\qquad \forall f,l \in J, f < l$$

The objective function of the proposed mathematical model is makespan minimization, as specified in Equation (2). Equation (3) stipulates that every task of each job must be assigned a single machine/operator and a single preceding activity, where these assignments are represented by binary variables $X_{khajibl}$. Equation (4) ensures that each activity can be followed by at most one other activity performed by the same operator. Equation (5) allows each operator sequence to originate at most once from the dummy unloading activity a_u of dummy job j_0, whereas Equation (6) prohibits the dummy job from being represented as a setup activity. Therefore, an operator sequence may originate only from the designated dummy unloading activity used as the starting node. Equation (7) ensures that the setup activity a_s and *unloading* a_u for job l are performed on the same machine. Equation (8) is used to maintain consistency in the sequence of operator activities. This constraint ensures that if an operator completes an activity, the next activity must follow a feasible sequence using the decision variable $\boldsymbol{X_{khajibl}}$. Equation (9) ensures that the time between movement completion and manual activity completion is at least the corresponding skill-adjusted activity duration. Equation (10) ensures that sufficient movement time is provided between consecutive activities performed by an operator, according to the corresponding machine-to-machine movement time. Equation (11) ensures that unloading of job l is completed only after machining completion and the required unloading duration. Equation (12) ensures that the machining process of the job on the machine is carried out immediately after the setup activity a_s completed. Equation (13) controls job sequencing on each machine using operator-dependent setup times and the binary sequencing variable Q_{fl} thereby preventing overlapping jobs on the same machine. Equation (14) ensures that the completion time of the dummy job's unloading activity O_{j0} is 0. Equation (15) defines the makespan value C_{max} as the maximum completion time among all job activities. Equation (16) defines the domain of the binary variables $\boldsymbol{X_{khajibl}}$ and Q_{fl}.

Following the methodology of Akbar and Irohara, experimental data were synthetically generated using a uniform distribution [8] and Costa *et al.* [20]. Machining time (P_{il}) generated from a distribution$U[1,99]$, setup time ($O_{a_s l}$) from $U[1,79]$, unloading time ($O_{a_u l}$) from $U[1,20]$, and operator moving time (V_{hi}) from $U[3,10]$. The setup and unloading time ranges were selected as $U[1,79]$ and $U[1,20]$ respectively. The selected durations assume that setup activities generally require more time than unloading because they involve more complex preparation tasks. All generated parameter values were stored as fixed data arrays and used consistently across computational runs and skill scenarios. Therefore, no random seed was required for instance generation, while the Gurobi seed was fixed at 0 to support consistent solver behavior. The operator skill coefficient η_k is defined as a scaling factor for the standard duration of manual activities. The actual duration of manual activity $b$for job $l$when performed by operator k is calculated using $O_{kbl} = O^{std}_{bl}.\eta_k$. A coefficient of $\eta_k < 1$ represents an above-average operator who completes the activity faster than the standard duration, $\eta_k = 1$ represents a standard operator, and $\eta_k > 1$ represents a below-average operator who requires more time to complete the activity. The heterogeneous-skill coefficients were adopted from the set $\{0.5, 0.75, 1.00, 1.25, 1.5\}$ used by Costa *et al* [20]. However, not all values of that set were used. Instead of being drawn randomly as in Costa *et al* [20], the coefficients were assigned deterministically for each scenario: $\eta_k = \{0.75, 1.25\}$ for the two-operator scenario and $\eta_k = \{0.75, 1.00, 1.25\}$ for the three-operator scenario. These subsets were chosen so that the mean coefficient equals 1.00 in both scenarios, which allows a controlled comparison with the homogeneous scenario, where all operators have a coefficient of 1.00. The extreme values 0.50 and 1.50 were excluded to keep the skill differences among operators moderate. Therefore, differences between the homogeneous and heterogeneous results reflect the distribution of operator skills rather than a change in average workforce capacity. Model validation was conducted by comparing the structure and limiting conditions of the proposed model with the previous MTSSDRC formulations developed by Akbar and Irohara [8] and Larasati *et al.* [13]. when operators are assumed to be homogeneous $\eta_k = 1$ to ensure that the addition of the operator skill parameter does not compromise the fundamental logic of the MTSSDRC model.

The computational results were evaluated using makespan reduction, true machine idle time, operator workload, operator utilization, and workload share. These performance measures were calculated consistently for homogeneous and heterogeneous skill scenarios. Makespan reduction measures the relative improvement obtained by explicitly representing operator-skill heterogeneity and is calculated as follows:

$$\text{Makespan reduction} = \frac{C_{\max}^{Hom} - C_{\max}^{Het}}{C_{\max}^{Hom}} \times 100\% \tag{17}$$

where $C_{\max}^{Hom}$ and $C_{\max}^{Het}$ denote the makespan values obtained under the homogeneous and heterogeneous skill scenarios, respectively. A positive value indicates that the heterogeneous skill representation produces a lower makespan than the homogeneous skill representation. The true machine idle (TMI) time represents the total unoccupied time of all machines within the scheduling horizon. For each machine, occupied time includes the setup, machining, and unloading durations of the jobs assigned to that machine. It is calculated as follows:

$$TMI = \sum_{i \in I} (C_{max} - B_i) \tag{18}$$

where B_i is the total occupied time of machine i, measured from the start of setup to the completion of unloading for each job assigned to that machine. It includes setup, machining, waiting-for-unloading, and unloading times. Thus, $(C_{max} - B_i)$ represents the total idle time of machine i within the scheduling horizon. The operator workload is defined as the total time an operator spends on assigned manual activities. In this study, the workload of operator k includes the setup and unloading durations of all activities assigned to that operator, together with the movement time preceding each activity. The workload of operator k is expressed as follows:

$$W_k = \sum_{b \in A} \sum_{l \in J} \sum_{h \in I} \sum_{a \in A} \sum_{j \in J,} \sum_{i \in I} (O_{kbl} + V_{hi})\, X_{khajibl} \tag{19}$$

The term inside parentheses equals 1 when manual activity b of real job l is assigned to operator k, and 0 otherwise. Utilization and workload share are expressed as follows:

$$Util\ k = \frac{Wk}{C_{max}} \times 100\%\ , Share\ k\ = \frac{Wk}{\sum_{q \in K} \text{Wq}} \times 100\% \tag{20}$$

where $Util\ k$ represents the percentage of the scheduling horizon during which operator k performs manual activities, whereas $Share\ k$ represents the proportion of the total operator workload assigned to operator k. These measures were used to examine whether makespan minimization concentrates manual activities on particular operators.

Results and Discussions

This study adapts and further develops the MTSSDRC model originally proposed by Akbar and Irohara [8]. The principal contribution lies in extending the model to an unrelated parallel machine environment that incorporates operator skill-level heterogeneity η_k. We evaluate model performance in terms of makespan C_{max}, computational effort, and solution feasibility. The effect of skill heterogeneity on the coordination of setup, machining, and unloading operations is also investigated. To assess solution accuracy, we employ the optimality MIP Gap (%), defined as the relative deviation between the best feasible solution and the optimal bound provided by Gurobi. A smaller MIP Gap (%) signifies greater solution precision and closer alignment with global optimality. The terms small and medium are used as computational instance-size classifications and do not refer to the physical size of a company or manufacturing facility. An instance denoted by $n \times m \times w$ represents n jobs, m unrelated parallel machines, and w operators considered within one scheduling horizon. In this study, instances with 4–5 jobs are classified as small, whereas instances with 6–10 jobs are classified as medium. This classification is used to examine the effect of increasing problem size on the number of constraints and variables, computational time, solution quality, and the solver's ability to prove optimality. The experimental results obtained using Gurobi Optimizer are presented in Table 2.

Table 2 presents the results for six small and 11 medium instances. All small instances (4–5 jobs) reached proven optimality with zero MIP gaps within 1.93–55.48 seconds. All medium instances (6–10 jobs) reached the 600-second limit but produced feasible solutions, with MIP gaps ranging from 31.15% to 67.75%. These gaps indicate the remaining difference between the best feasible solution and the best bound, rather than solution infeasibility. The results also demonstrate a substantial increase in computational complexity as the instance

Table 2. MILP Computational performance

Case ($n \times m \times w$)	Rows	Columns	Makespan ($C_{\max}$)	MIP Gap (%)	Gurobi runtime (s)	Status
4x3x2	1512	1481	146.5	0	2.07	Optimal
4x4x2	2552	2601	132.75	0	2.54	Optimal
4x4x3	3737	3881	116	0	1.93	Optimal
5x3x2	2284	2215	180	0	32.19	Optimal
5x4x2	3869	3895	162.25	0	55.48	Optimal
5x4x3	5670	5815	135	0	19.14	Optimal
6x3x2	3214	3095	228	44.63	600.03	Best Found
6x4x2	5458	5447	209.25	44.02	600.03	Best Found
6x4x3	8003	8135	168.5	31.15	600.02	Best Found
7x4x2	7319	7257	259.5	53.55	600.03	Best Found
7x4x3	10736	10841	199.75	41.92	600.05	Best Found
8x4x2	9452	9325	294.25	60.57	600.04	Best Found
8x4x3	13869	13933	249.5	53.50	600.10	Best Found
9x4x3	17402	17411	308.75	62.42	600.06	Best Found
9x4x2	11857	11651	359.75	67.75	600.06	Best Found
10x4x2	14534	14235	359	67.68	600.07	Best Found
10x4x3	21335	21275	356	67.41	600.04	Best Found

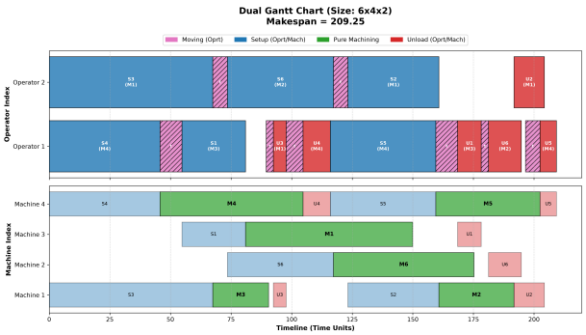


Figure 2. Gantt chart for the 6×4×2 instance Size

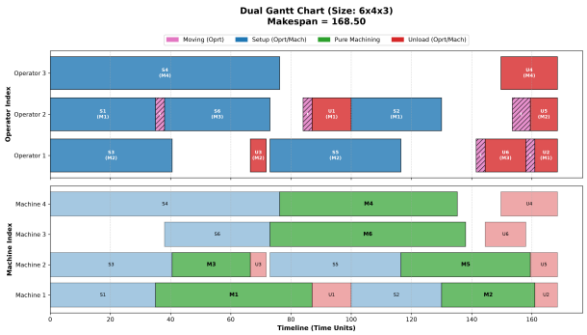


Figure 3. Gantt chart for the 6×4×3 instance size

size increases. The model size increased from 1,512 constraints and 1,481 variables for the $4 \times 3 \times 2$ instance to 21,335 constraints and 21,275 variables for the $10 \times 4 \times 3$ instance. This increase is accompanied by longer solution times and larger optimality gaps. These findings indicate that the exact MILP approach is appropriate for obtaining optimal solutions for small instances and feasible benchmark solutions for medium instances. For larger or real-time scheduling applications, the development of a metaheuristic or hybrid solution approach may be considered in future research.

Figures 2 and 3 present the Gantt charts of the best solutions in Table 2 for the $6 \times 4 \times 2$ and $6 \times 4 \times 3$ instances. The upper panel shows operator activities (setup, unloading, and movement), and the lower panel shows machine occupation (setup, machining, and unloading). In Figure 2, Operator 1 ($\eta = 0.75$)performs eight manual activities, whereas Operator 2($\eta = 1.25$)performs four. In Figure 3, the additional operator reduces the makespan from 209.25 to 168.50, while the slowest operator ($\eta = 1.25$)performs only two activities. Gaps between the end of machining and the start of unloading indicate that machines are waiting for an available operator.

To isolate the effect of operator skill heterogeneity, homogeneous and heterogeneous scenarios were designed with the same mean skill coefficient of 1.00. For two-operator instances, the homogeneous skill vector was $\{1.00, 1.00\}$, whereas the heterogeneous vector was $\{0.75, 1.25\}$. For three-operator instances, the corresponding vectors were $\{1.00, 1.00, 1.00\}$ and $\{0.75, 1.00, 1.25\}$. Therefore, differences in the resulting schedules can be attributed to the distribution of operator skills rather than to differences in average workforce capacity. The results are presented in Table 3.

Following the comparison of the schedule performance in Table 3, the operator assignment results were examined to identify how the MILP model distributes manual activities among operators with different skill coefficients. Table 4 reports the number of setup and unloading assignments, total workload, utilization, and workload share for each operator in the heterogeneous scenario at $\delta = 0.25$.

Table 3. Effect of operator skill heterogeneity on makespan and machine idle time

Case ($n \times m \times w$)	Hom. $C_{\max}$	Het. $C_{\max}$	Difference	Reduction (%)	Hom. idle	Het.idle	Idle difference	Idle reduction (%)
$4 \times 3 \times 2$	158.00	146.50	11.50	7.28	63.00	53.50	9.50	15.08
$4 \times 4 \times 2$	141.00	132.75	8.25	5.85	146.00	141.75	4.25	2.91
$4 \times 4 \times 3$	135.00	116.00	19.00	14.07	118.00	60.25	57.75	48.94
$5 \times 3 \times 2$	190.00	180.00	10.00	5.26	59.00	61.50	−2.50	−4.24
$5 \times 4 \times 2$	177.00	162.25	14.75	8.33	169.00	157.25	11.75	6.95
$5 \times 4 \times 3$	155.00	135.00	20.00	12.90	108.00	38.75	69.25	64.12
Average	159.33	145.42	13.92	8.95	110.50	85.50	25.00	22.29

Table 4. Operator assignment, workload, and utilization under the heterogeneous skill scenario

Case	Operator	Skill coefficient η_k	Assignment (s+u)	Workload	Utilization (%)	Workload share (%)
$4 \times 3 \times 2$	1	0.75	6	129.75	88.57	58.78
	2	1.25	2	91.00	62.12	41.22
$4 \times 4 \times 2$	1	0.75	4	126.00	94.92	51.59
	2	1.25	4	118.25	89.08	48.41
$4 \times 4 \times 3$	1	0.75	3	80.75	69.61	33.93
	2	1.00	3	81.00	69.83	34.03
	3	1.25	2	76.25	65.73	32.04
$5 \times 3 \times 2$	1	0.75	6	170.25	94.58	60.05
	2	1.25	4	113.25	62.92	39.95
$5 \times 4 \times 2$	1	0.75	6	157.75	97.23	51.98
	2	1.25	4	145.75	89.83	48.02
$5 \times 4 \times 3$	1	0.75	6	128.25	95.00	44.00
	2	1.00	2	76.00	56.30	26.07
	3	1.25	2	87.25	64.63	29.93

Table 5. Sensitivity of makespan and true machine idle time to operator skill heterogeneity

Case	Performance indicator	$\delta = 0.00$	$\delta = 0.05$	$\delta = 0.10$	$\delta = 0.15$	$\delta = 0.20$	$\delta = 0.25$
$4 \times 3 \times 2$	Makespan	158.00	152.95	150.90	147.25	149.40	146.50
	True machine idle	63.00	49.90	51.80	38.30	56.00	53.50
$4 \times 4 \times 2$	Makespan	141.00	143.05	138.20	135.70	134.00	132.75
	True machine idle	146.00	168.85	138.00	138.60	132.40	141.75
$4 \times 4 \times 3$	Makespan	135.00	131.20	127.40	123.60	119.80	116.00
	True machine idle	118.00	124.00	107.50	66.20	81.40	60.25
$5 \times 3 \times 2$	Makespan	190.00	186.30	187.90	189.35	184.80	180.00
	True machine idle	59.00	46.95	57.00	67.85	59.40	61.50
$5 \times 4 \times 2$	Makespan	177.00	171.50	168.40	168.05	165.60	162.25
	True machine idle	169.00	147.90	152.30	101.40	136.80	157.25
$5 \times 4 \times 3$	Makespan	155.00	149.95	147.10	144.35	138.00	135.00
	True machine idle	108.00	61.65	72.80	44.95	43.80	38.75

After examining operator allocation and workload distribution, a sensitivity analysis was conducted to evaluate how different levels of skill heterogeneity affect schedule performance. The skill-spread parameter $\delta$was varied from 0.00 to 0.25 while maintaining the mean operator skill coefficient at 1.00. A value of $\delta = 0.00$ represents a homogeneous scenario, whereas larger values represent increasing differences in operator skill. Table 5 presents the resulting makespan and true machine idle time for each test instance.

Benchmarking

The MILP model was internally benchmarked using Gurobi's incumbent solutions and best bounds. All six small instances reached zero MIP gaps within 1.93–55.48-s, confirming optimality. For the 11 medium

instances, feasible solutions were obtained within the 600-s limit, with MIP gaps of 31.15%–67.75%. These results provide reference solutions and bounds for evaluating future algorithms. Because no direct comparison with metaheuristics was conducted, this study does not claim computational superiority.

Managerial Implications

The results show that operatorskill heterogeneity reduced makespan by 5.26%–14.07% across the six small instances, averaging 8.95% (Table 3). Table 4 suggests that assigning skilled operators to critical setup and unloading activities may improve scheduling performance, although it can increase workload imbalance. Managers should therefore consider operator rotation, utilization limits, and workload-balancing objectives. Table 5 also shows that makespan reduction does not always correspond to lower machine idle time. The MILP model is suitable for offline analysis of small instances and may provide benchmark solutions for medium instances. However, validation using actual production data is required before industrial application.

Conclusions

This study developed a skill-aware MTSSDRC scheduling model for unrelated parallel machines. The model integrates job allocation, machine selection, operator assignment, setup and unloading activities, and operator movement within a single MILP formulation. All six small instances were solved to optimality, while the 11 medium instances produced feasible solutions within the 600-second time limit, although global optimality could not be proven. Under the same average workforce skill, the heterogeneous scenario reduced makespan in all small instances by an average of 8.95%. True machine idle time decreased by an average of 22.29%, but the improvement was not consistent across all cases. The results also show that makespan reduction may concentrate workload on particular operators. Therefore, operator skill differences can improve scheduling performance, however managers should consider the trade-off between production efficiency and workload balance. The main limitations are the reliance on synthetic instances and the increasing computational difficulty for medium cases. Future research should validate the model using industrial data and develop metaheuristic or hybrid approaches for larger and real-time scheduling problems.

References

[1] S. Zhang, H. Du, S. Borucki, S. Jin, T. Hou, and Z. Li, "Dual resource constrained flexible job shop scheduling based on improved quantum genetic algorithm," *Machines*, vol. 9, no. 6, p. 108, 2021, doi: https://doi.org/10.3390/machines9060108.

[2] G. Yang, B. Do Chung, and S. J. Lee, "Limited search space-based algorithm for dual resource constrained scheduling problem with multilevel product structure," *Appl. Sci.*, vol. 9, no. 19, p. 4005, 2019, doi: https://doi.org/10.3390/app9194005.

[3] Z. Gao and H. Qiao, "A study of flexible job-shop scheduling with dual constraints," *Processes*, vol. 12, p. 2649, 2024, doi: https://doi.org/10.3390/pr12122649.

[4] X. Gong, Q. Deng, G. Gong, W. Liu, and Q. Ren, "A memetic algorithm for multi-objective flexible job-shop problem with worker flexibility," *Int. J. Prod. Res.*, vol. 56, no. 7, pp. 2506–2522, 2017, doi: https://doi.org/10.1080/00207543.2017.1388933.

[5] M. Akbar and T. Irohara, "Dual resource constrained scheduling considering operator working modes and moving in identical parallel machines using a permutation-based genetic algorithm," *IFIP Adv. Inf. Commun. Technol.*, vol. 535, pp. 464–472, 2018, doi: https://doi.org/10.1007/978-3-319-99704-9_57.

[6] A. Elidrissi, R. Benmansour, K. Hasani, and F. Werner, "Minimizing the makespan on two parallel machines with a common server in charge of loading and unloading operations," *Comput. Oper. Res.*, vol. 167, p. 106638, 2024, doi: https://doi.org/10.1016/j.cor.2024.106638.

[7] P. Baptiste, D. Rebaine, and M. Zouba, "FPTAS for the two identical parallel machine problem with a single operator under the free changing mode," *Eur. J. Oper. Res.*, vol. 256, no. 1, pp. 55–61, 2017, doi: https://doi.org/10.1016/j.ejor.2016.05.062.

[8] M. Akbar and T. Irohara, "Metaheuristics for the multi-task simultaneous supervision dual resource-constrained scheduling problem," *Eng. Appl. Artif. Intell.*, vol. 96, p. 104004, 2020, doi: https://doi.org/10.1016/j.engappai.2020.104004.

[9] M. Akbar and T. Irohara, "NSGA-II variants for solving a social-conscious dual resource-constrained scheduling problem," *Expert Syst. Appl.*, vol. 162, p. 113754, 2020, doi: https://doi.org/10.1016/j.eswa.2020.113754.

[10] M. Akbar and T. Irohara, "NSGA families for solving a dual resource-constrained problem to optimize the

total tardiness and labor productivity in the spirit of sustainability," *Comput. Ind. Eng.*, vol. 188, p. 109883, 2024, doi: https://doi.org/10.1016/j.cie.2024.109883.

[11] Z. He, B. Tang, and F. Luan, "An improved African vulture optimization algorithm for dual-resource constrained multi-objective flexible job shop scheduling problems," *Sensors*, vol. 23, no. 1, p. 90, 2023, doi: https://doi.org/10.3390/s23010090.

[12] X. Zhu, J. Xu, J. Ge, Y. Wang, and Z. Xie, "Multi-task multi-agent reinforcement learning for real-time scheduling of a dual-resource flexible job shop with robots," *Processes*, vol. 11, no. 1, p. 267, 2023, doi: https://doi.org/10.3390/pr11010267.

[13] K. A. Larasati, Sukoyo, and M. Akbar, "A dual resource constrained unrelated parallel machine scheduling model considering tardiness and workload balance," *Jurnal Teknik Industri: Jurnal Keilmuan dan Aplikasi Teknik Industri*, vol. 26, no. 2, pp. 87–102, 2024, doi: https://doi.org/10.9744/jti.26.2.87-102.

[14] I. M. Al-harkan, A. A. Qamhan, A. Badwelan, A. Alsamhan, and L. Hidri, "Modified harmony search algorithm for resource-constrained parallel machine scheduling problem with release dates and sequence-dependent setup times," *Processes*, vol. 9, no. 4, p. 654, 2021, doi: https://doi.org/10.3390/pr9040654.

[15] A. Rizka, Sukoyo, and M. Akbar, "Multi-task simultaneous supervision: Dual resource-constrained scheduling problem in identical parallel machines considering differences in operator skill levels," *Jurnal Teknik Industri: Jurnal Keilmuan dan Aplikasi Teknik Industri*, vol. 26, no. 2, pp. 119–130, 2024, doi: https://doi.org/10.9744/jti.26.2.119-130.

[16] C. Wang, D. Fan, Y. Liu, S. Ren, and J. Wang, "Dual-resource flexible job shop scheduling considering worker proficiency differences," *Comput. Oper. Res.*, vol. 184, p. 107216, 2025, doi: https://doi.org/10.1016/j.cor.2025.107216.

[17] L. Zhang, Q. Deng, R. Lin, G. Gong, and W. Han, "A combinatorial evolutionary algorithm for unrelated parallel machine scheduling problem with sequence and machine-dependent setup times, limited worker resources and learning effect," *Expert Syst. Appl.*, vol. 175, p. 114843, 2021, doi: https://doi.org/10.1016/j.eswa.2021.114843.

[18] F. Costa, M. Thürer, and A. Portioli-Staudacher, "Heterogeneous worker multi-functionality and efficiency in dual resource constrained manufacturing lines: an assessment by simulation," *Oper. Manag. Res.*, vol. 16, pp. 1476–1489, 2023, doi: https://doi.org/10.1007/s12063-023-00371-2.

[19] J. I. Vázquez-Serrano, L. E. Cárdenas-Barrón, and R. E. Peimbert-García, "Agent scheduling in unrelated parallel machines with sequence-and agent–machine–dependent setup time problem," *Mathematics*, vol. 9, no. 22, p. 2955, 2021, doi: https://doi.org/10.3390/math9222955.

[20] A. Costa, F. A. Cappadonna, and S. Fichera, "A hybrid genetic algorithm for job sequencing and worker allocation in parallel unrelated machines with sequence-dependent setup times," *Int. J. Adv. Manuf. Technol.*, vol. 69, no. 9–12, pp. 2799–2817, 2013, doi: https://doi.org/10.1007/s00170-013-5221-5.

[21] P. Heuser, P. Letmathe, and T. Vossen, "Skill development in the field of scheduling: A structured literature review," *Eur. J. Oper. Res.*, vol. 321, no. 3, pp. 697–716, 2025, doi: https://doi.org/10.1016/j.ejor.2024.04.005.