Cloud Manufacturing Rescheduling Under New Task Arrival Disturbances: A Hybrid Metaheuristic Approach

Atefeh Rajabi-Kafshgar [a], Mostafa Hajiaghaei-Keshteli [b,*], Mohammad Reza Mohammad Aliha [c]

[a] *School of Industrial Engineering, Iran University of Science & Technology, Tehran, Iran*
E-mail: rajabi_atefeh@ind.iust.ac.ir
Orcid: 0000-0002-2769-8694

[b] *Tecnologico de Monterrey, Escuela de Ingeniería y Ciencias, Mexico*
E-mail: mostafahaji@tec.mx
Orcid: 0000-0002-9988-2626
(Corresponding Author)

[c] *School of Industrial Engineering, Iran University of Science & Technology, Tehran, Iran*
E-mail: mrm_aliha@iust.ac.ir
Orcid: 0000-0001-9451-0949

Abstract

Recently, the growing demand for customized products and advances in smart technologies have accelerated the shift toward cloud manufacturing (CMg). Although CMg offers high flexibility, its dynamic nature introduces major scheduling challenges, such as new task arrivals and strict delivery constraints, which are often overlooked in existing models. To address these limitations, this study formulates a dynamic scheduling problem in CMg (DSPCMg) that integrates new task arrivals with the objective of minimizing delivery time deviations. Given the NP-hardness of the problem, five well-established metaheuristic algorithms are implemented, and six hybrid algorithms are developed to achieve a better balance between global exploration and local exploitation. In addition to modeling dynamic task arrivals, the proposed framework incorporates sequence-dependent setup times, delivery time windows, and logistics considerations within a unified formulation. The performance of the proposed algorithms is evaluated using test problems and 30 benchmark instances for both the scheduling and rescheduling stages. Computational experiments show that the hybrid KA-TS algorithm achieves the best performance in the scheduling stage, whereas GA-TS performs best in rescheduling scenarios. Moreover, the proposed rescheduling approach reduces delivery deviation by up to 45% and machine idle time by up to 32% compared with fixed initial schedules. Finally, sensitivity analysis further highlights that increases in logistics times and the number of new tasks significantly raise delivery time deviations.

Keywords: Cloud Manufacturing, Dynamic Scheduling, Reactive Rescheduling, Hybrid Metaheuristics, Delivery Time Deviation, Task Insertion.

---

[*] *Corresponding author*

1. Introduction:

In response to the growing demand for customized, on-demand production services, manufacturing systems are undergoing a significant shift toward service-oriented and network-based architectures. Among these innovations, cloud manufacturing (CMg) has gained attention as a model that virtualizes manufacturing capabilities and delivers them as intelligent services over a distributed network [1]. Rather than relying on fixed, factory-centric production lines, CMg enables dynamic resource sharing and global collaboration across geographically dispersed service providers [2]. With such potential, CMg has started to attract increasing attention from the manufacturing industry.

The report by Research and Markets indicates that 74% of manufacturers are already using or planning to use artificial intelligence in their operations, and 83% aim to upgrade their processes toward smart factories[1]. These trends suggest a clear shift toward smarter and more connected production environments such as CMg. This rapid evolution underscores the urgency of addressing key operational challenges, particularly scheduling and resource allocation of the CMg system [3].

In the CMg system, customers submit manufacturing requirements to the platform, which then allocates the necessary manufacturing services to fulfill them [4]. The cloud platform must match each order's requirements to available machines or service providers [5]. The matching involves solving a complex scheduling problem: deciding which machine executes each task and in what order [6]. This highlights not only the need for efficient scheduling but also for accommodating practical constraints.

Effective scheduling in CMg must respect several realistic constraints that directly impact system performance and customer satisfaction [7]. These include sequence-dependent setup times (SDST), delivery time windows (DTW), task arrival times (TAT)[8, 9], and final logistics, which is often overlooked in the literature. Accurately modeling these constraints is essential for developing practical scheduling solutions, particularly under dynamic and uncertain conditions.

CMg systems also face frequent disruptions like equipment breakdowns, task cancellations, and urgent task arrivals [10]. While traditional models assume static environments, real-world systems require reactive scheduling that dynamically adjusts plans in response to disruptions [11]. One common trigger for such rescheduling is the arrival of new tasks that must fit into the current schedule [12].

[1] https://www.researchandmarkets.com/report/global-cloud-manufacturing-market Retrieved on April 21, 2025.

New tasks often arrive after the initial schedule has been generated and partially executed, making static plans inefficient or infeasible. These tasks arrive unpredictably and must be immediately integrated into the existing schedule. Hence, accurately modeling dynamic environments and designing effective rescheduling strategies is essential [13].

This study addresses a dynamic scheduling problem in CMg systems (DSPCMg), considering new task arrival, delivery time windows, SDST, and final logistics. The primary objective is to minimize delivery window deviations to improve system performance. Due to NP-hard nature of DSPCMg [14], exact algorithms become computationally infeasible for large-scale or real-time instances. Six hybrid metaheuristic algorithms (MAs) have been designed and implemented to address the proposed model and deal with the identified challenge. The hybrid approach leverages population-based metaheuristics (PM) for global search abilities and single-solution-based metaheuristics (SM) for local exploitation, achieving a balanced and adaptive solution framework [15]. This study puts forward the following key contributions.

(1) This study firstly introduces a DSPCMg that supports new task insertion and simultaneously integrates several constraints, including delivery time windows, SDST, task arrival times, and final logistics.
(2) The study develops six hybrid MAs for the proposed model.
(3) An experimental evaluation was carried out on well-established benchmark datasets, and statistical analysis using the Friedman test was conducted to validate the performance of the proposed algorithms.

The structure of the paper is as follows. Section 2 provides an overview of the relevant literature and existing solution methods. Section 3 presents the problem definition along with the rescheduling framework. The proposed hybrid MAs are introduced in Section 4. Section 5 discusses the experimental design, including benchmark-based evaluations and statistical analysis. Lastly, the study is concluded in Section 6, which also discusses directions for future research.

2. Literature review

The dynamic behavior of CMg systems driven by real-time changes in resource availability necessitates adaptive scheduling strategies. For instance, Zhang et al. [16] employed a game theory-based approach to address machine failures, while Ding et al. [17]

proposed a robust scheduling model using a two-stage Genetic Algorithm (GA) to enhance stability under service failure.

Another line of research focuses on task-level variations, including deviations in task requirements [18], task modifications [19], urgent tasks [20], or new task insertions [21, 22]. These studies employed MA to adapt to these dynamic changes. Additionally, Zhang et al. [23] applied a learning based Markov decision framework to handle dynamic task arrival.

Service-related changes, such as failures in services, were addressed in several works. Wang et al. [24] and Xiong et al. [25] employed RL-based MDPs and multi-objective MAs, respectively, to respond to service disruptions. Zhang et al. [26] focused on logistics cost optimization in a digital twin-based CMg system, while Hu et al. [27] used a game-theoretic model to manage both random arrivals and breakdowns.

Further studies have explored dynamic service quality and availability. Jing et al. [9] applied MA to cope with service quality variations. In contrast, some papers developed RL-based optimization methods, such as the AC algorithm [28] and PPO algorithms [29] to adjust service rates or respond to unavailability.

Recent studies have introduced integrated models to address multiple dynamic factors. Shao and Ren [30] used a blockchain-based system and GA to minimize delays due to task and service variability. Similarly, Xu et al. [31] utilized a MIP model and MA to handle dynamic task arrivals. Same in dynamic event and different in model, Lei et al. [12] focused on optimization of disruption value and makespan.

Overall, existing research emphasizes the significant impact of task- and service-level dynamics on CMg scheduling. Researchers implemented a variety of optimization techniques, including MAs, RL-based, and hybridized strategies to handle these challenges. These efforts underscore the increasing shift from static to dynamic scheduling models capable of responding to real-time changes. An overview of the modeling and applied solution methods in the literature is presented in Table 1. Such limitations highlight unresolved challenges and open the path toward identifying critical research gaps, which are discussed in detail in the next section.

2.1. Research gap

Despite the growing body of research on DSPCMg, several critical aspects remain insufficiently explored. Firstly, SDST are often ignored, leading to inaccurate schedules and inefficient resource use [29]. Second, logistics constraints are often limited to intermediate stages, while neglecting final logistics can result in suboptimal plans. In recent cloud manufacturing studies, several models have incorporated final logistics into static scheduling frameworks [32]. Some works even focused specifically on logistics services and their role in integrated scheduling decisions [33, 34]. While these contributions have advanced the modeling of logistics in cloud-based production systems, they are largely confined to static environments. To the best of our knowledge, there is a lack of studies that explicitly address logistics costs and final delivery considerations in dynamic scheduling contexts. This limitation highlights a significant gap that our proposed model aims to fill by integrating real-time scheduling and rescheduling mechanisms with logistics decisions.

Additionally, delivery windows and their violations play a crucial role in customer satisfaction, yet they have received limited attention in existing research. Also, despite extensive research on DSPCMg, few studies integrate dynamic challenges such as new task arrivals with key constraints like SDST, final logistics, and delivery windows in a unified model. Finally, while MAs are commonly applied to CMg scheduling problems, hybrid approaches have received limited attention, an aspect this study seeks to address. Therefore, this study fills these gaps by proposing a dynamic optimization model and developing hybrid MAs. The following sections present comprehensive explanations of the model and solution approaches.

3. Model description and formulation

This research addresses a scheduling problem that involves I distinct tasks, denoted as set T. Each task (T_i) is characterized by a defined set of operations (Ot_i), arrival time (AT_i), delivery time window (DT_i=[d_i^-, d_i^+]), and customer location (CL_i). The operations r of task i (O_{ir}) has an eligible machine set (AS_{ir}), indicating the machines capable of processing it (e.g., if operation O_{13} can be processed on M_{12}, M_{15}, and M_{42}, then AS_{13} = {M_{12}, M_{15}, M_{42}}).

The manufacturing system consists of N distributed factories F, where factory F_n offers a set of machines Mf_n located at FL_n. Machine M_{ns} in factory F_n can process operation O_{ir} if $M_{ns} \in AS_{ir}$, and is characterized by the setup time SeT_{irns} and processing time PrT_{irns}.

Additionally, if a task requires processing across multiple factories, middle transportation between factories is needed (Figure 1). The middle transportation and the final transportation time from the last factory to the customer, are computed using Euclidean distance (equations (1) and (2)).

$$MTT_{nm} = \left\| FL_n - FL_m \right\| \tag{1}$$

$$FTT_{ni} = \left\| CL_i - FL_n \right\| \tag{2}$$

This research aims to improve operation scheduling across distributed machines by minimizing overall deviations from specified delivery windows, considering both early and late deliveries. Moreover, unexpected events can disrupt the pre-established schedules, rendering static scheduling approaches inadequate. To address this, the proposed model integrates a reactive scheduling mechanism that adjusts the production schedule in response to new task insertions. The detailed formulation and implementation are presented in the following sections. The proposed dynamic scheduling problem in cloud manufacturing (DSPCMg) is formulated under the following assumptions:

- Tasks consist of sequential operations with predefined processing orders.
- A predefined set of eligible machines is available for processing each operation.
- Machines are continuously available and process only one operation at a time.
- All system parameters are deterministic and known in advance.
- New tasks may appear during the scheduling horizon, requiring schedule updates.
- Each task has a specific delivery time window.
- Final transportation stages are considered, which affect the actual delivery time.

The last three practical features enhance realism by reflecting system responsiveness, deadlines, and delivery constraints. The following notations are used for the DSPCMg model.

Sets:

$T = \{T_i \mid 1 \le i \le I\}$	All tasks under consideration
$F = \{F_i \mid 1 \le n \le N\}$	Available factories,
$Ot_i = \{O_{ir} \mid 1 \le r \le R_i\}$	Operations sequence associated with T_i
$Mf_n = \{M_{ns} \mid 1 \le s \le S_n\}$	Available machines in F_n

$T' = \{T_p' \mid 1 \le i \le I'\}$ Newly arrived tasks

$T^u = \{T_p' \mid 1 \le i \le I^u\}$ Incomplete tasks

$Ot_i^u = \{O_{ir} \mid 1 \le r \le R_i\}$ Unprocessed operations of T_i at rescheduling time

Indices:

i, j	Task index
r, u	Operation index
n, m	Factory index
s, v	Machine index
p	New task index
q	Tasks index in rescheduling problem

Parameters:

I	Total tasks
N	Total factories
R_i	Total operations of T_i
S_n	Total machines in F_n
P	Total new tasks
d_i^-	Earliest delivery time of T_i
δ_i	Earliest penalty of T_i
d_i^+	Latest delivery time of T_i
λ_i	Latest penalty time of T_i
AT_i	Arrival time of T_i
ET	Entrance time of new tasks
PrT_{irns}	Manufacturing time of the O_{ir} on M_{ns}
SeT_{runs}	Setup time of O_{ir} after O_{ju} on M_{ns}
MTT_{nm}	Middle transportation time from F_n to F_m
FTT_{ni}	Final transportation time from F_n to customer i

Variables:

AD_i	Actual delivery time of T_i
Er_i	Earliness of T_i
Tr_i	Tardiness of T_i
FT_{ir}	Finish time of O_{ir}
ST_{ir}	Start time of O_{ir}
φ_{irnm}	1 if O_{ir} is transported between F_n and F_m; otherwise, 0
Y_{irju}^{ns}	1, if O_{ir} precede O_{ju} on M_{ns}; otherwise, 0
Z_{irns}	1, if O_{ir} is allocated to M_{ns}; otherwise, 0
X_{irn}	1, if O_{ir} is allocated to the F_n; otherwise, 0

3.1. Objective function

In CMg systems, timely delivery is essential for both customer satisfaction and system efficiency [37]. Deviations from the preferred delivery window, whether early or late, may cause disruptions or penalties. Thus, the model minimizes total deviation from predefined delivery time windows [d_i^- , d_i^+] by penalizing both earliness and tardiness, as formulated in Equation (3). Based on Prospect Theory, delays (tardiness) receive higher penalties than early deliveries ($\delta < \lambda$) due to their greater negative impact on efficiency and customer satisfaction [38]. Importantly, the actual delivery time AD_i includes both the finish time of the last operation and final transportation time, enabling the model to capture actual completion time beyond the traditional makespan.

$$\min DP = \sum_i \delta_i \max\left\{(d_i^- - AD_i), 0\right\} + \lambda_i \max\left\{(AD_i - d_i^+), 0\right\} \tag{3}$$

3.2. Constraints

Several constraints were considered based on the operational and logistical requirements of the problem. Each constraint is described in detail below. Constraints (4) and (5) represent the machine and factory allocation constraints. These allocation constraints ensure that the operation O_{ir} is assigned to exactly one machine and one factory in the system.

$$\sum_n \sum_s X_{im} = 1, \forall i, r \tag{4}$$

$$\sum_s Z_{ims} \leq X_{im}, \forall i, r, n \tag{5}$$

Constraint (6) guarantees that the first operation for each task does not precede its arrival time. Moreover, constraint (7) integrates the middle transportation time into the scheduling of subsequent operations for ($r > 1$).

$$ST_{ir} \geq AT_i, \forall i, (r = 1) \tag{6}$$

$$ST_{ir} \geq FT_{i(r-1)} + \sum_n \sum_m \sum_s \sum_v MTT_{nm} Z_{i(r-1)ns} \times Z_{irmv}, \forall i, r \ (r > 1) \tag{7}$$

Constraints (8)-(10) enforce that if multiple operations should be executed by one machine, only one sequence is accepted, and the SDST between tasks is correctly accounted for, depending on the sequence of operations in a machine.

$$Y_{irju}^{ns} + Y_{juir\,i}^{ns} \le Z_{irns}, \forall i, r, j, u, n, s \tag{8}$$

$$Y_{irju}^{ns} + Y_{juir}^{ns} + 1 \ge Z_{irns} + Z_{juns}, \forall i, r, j, u, n, s \tag{9}$$

$$ST_{ir} \ge ST_{ju} + PrT_{juns} + SeT_{runs} - M\,(1 - Y_{irju}^{ns}), \forall i, r, j, u, n, s \tag{10}$$

Constraint (11) guarantees that no task finishes before it begins or without accounting for its processing time. Moreover, constraint (12) represents that the actual delivery time of a task is not earlier than the finish time of its final operation and the associated final transportation time.

$$FT_{ir} \ge ST_{ir} + \sum_{n}\sum_{s} PrT_{irns}\, Z_{irns}, \forall i, r \tag{11}$$

$$AD_{i} \ge FT_{iR_i} + \sum_{n}\sum_{s} FTT_{in}\, Z_{iR_i ins}, \forall i \tag{12}$$

The linearized model is provided in follow which equation (13) is the objective function, equations (14)-(28) are the linearized version of equations (4)-(12), and equations (29)-(31) illustrate that the variables are binary and positive.

$$\min DP = \sum_{i} \delta_i (Er_i) + \lambda_i (Tr_i) \tag{13}$$

s.t:

$$\sum_{n}\sum_{s} X_{irn} = 1, \forall i, r \tag{14}$$

$$\sum_{s} Z_{irns} \le X_{irn}, \forall i, r, n \tag{15}$$

$$ST_{ir} \ge AT_i, \forall i, (r = 1) \tag{16}$$

$$\varphi_{irnm} \ge Z_{i(r-1)ns} + Z_{irmv} - 1, \forall i, r, n, m, s, v,\ (r > 1) \tag{17}$$

$$\varphi_{irnm} \le Z_{i(r-1)ns}, \forall i, r, n, m, s, v,\ (r > 1) \tag{18}$$

$$\varphi_{irnm} \le Z_{irmv}, \forall i, r, n, m, s, v,\ (r > 1) \tag{19}$$

$$ST_{ir} \geq FT_{i(r-1)} + \sum_{n}\sum_{m}\sum_{s}\sum_{v} MTT_{nm}\,\varphi_{irnm}, \forall i, r \ \left(r > 1\right) \quad (20)$$

$$Y_{irju}^{ns} + Y_{juir\,i}^{ns} \leq Z_{irns}, \forall i, r, j, u, n, s \quad (21)$$

$$Y_{irju}^{ns} + Y_{juir}^{ns} + 1 \geq Z_{irns} + Z_{juns}, \forall i, r, j, u, n, s \quad (22)$$

$$ST_{ir} \geq ST_{ju} + PrT_{juns} + SeT_{runs} - M\,(1 - Y_{irju}^{ns}), \forall i, r, j, u, n, s \quad (23)$$

$$FT_{ir} \geq ST_{ir} + \sum_{n}\sum_{s} PrT_{irns}\,Z_{irns}, \forall i, r \quad (24)$$

$$AD_{i} \geq FT_{iR_i} + \sum_{n}\sum_{s} FTT_{in}\,Z_{iR_i ins}, \forall i \quad (25)$$

$$Er_{i} \geq d_{i}^{-} - AD_{i}, \forall i \quad (26)$$

$$Tr_{i} \geq AD_{i} - d_{i}^{+}, \forall i \quad (27)$$

$$AD_{i}, Er_{i}, Tr_{i} \geq 0, \forall i \quad (28)$$

$$FT_{ir}, ST_{ir} \geq 0, \forall i, r \quad (29)$$

$$\varphi_{irnm}, Z_{irns} \in \left\{0,1\right\}, \forall i, r, n, s, m \quad (30)$$

$$Y_{irju\,i}^{ns} \in \left\{0,1\right\}, \forall i, j, r, u, n, s \quad (31)$$

4. Reactive scheduling strategies

In dynamic manufacturing systems like CMg, the arrival of new tasks is frequent and can significantly disrupt preplanned [20]. Ignoring such dynamic events and fixed static schedules often leads to inefficient production outcomes. Therefore, reactive strategies are essential to ensure responsiveness, reduce disruptions, and maintain efficient resource utilization under uncertainty [21, 22]. To integrate newly arrived tasks into the existing schedule, two reactive strategies are proposed.

Strategy 1 (fixed initial schedule): This approach aims to minimize disruption to the original schedule while accommodating new tasks as efficiently as possible within the available capacity. This strategy maintains the original schedule (AD_i^*, FT_{ir}^*, and ST_{ir}^*) and machine allocations (Z_{irns}^*, X_{irn}^*) and assigns new tasks to machines based on machine availability (MAT_{ns}) after the finish time of current operations (equations (32) and (33)). The new tasks are considered by incorporating equations (32)-(34) into the initial model which replace ($i \in T$) with ($q \in T'$) in original scheduling and allocation constraints. Although this approach stabilizes the schedule, it may compromise resource efficiency.

$$MAT_{ns} \geq \max_{q \in T, r \in Ot_i} \left\{ FT_{qr} \times Z^*_{qms} \right\}, \forall n,s \tag{32}$$

$$MAT_{ns} \geq 0, \forall n,s \tag{33}$$

$$ST_{qr} \geq \max_{q \in T', r \in Ot_q} \left\{ ET, MAT_{ns} \times Z_{qms} \right\}, \forall q,r,n,s,(q \in T') \tag{34}$$

Strategy 2 (rescheduling): This model just maintains the original factory allocation (X_{irn}^*) but allows resequencing and reassignment of machines for a combined set of operations of the unprocessed operations and operations of newly arrived tasks ($Ot_q \in \{Ot_i^u \cup Ot_p\}$). The rescheduling process solves a new scheduling optimization problem over (O_{qr}). In this model, equations (35)-(36) are added into the initial model to consider the machine available time at arrival time of new tasks. Moreover, Equation (37) defines the operation start times involved in the rescheduling problem and Equation (38) maintains original factory assignments for current operations in the initial schedule.

$$MAT_{ns} \geq \max \left\{ FT_{qr} \times Z_{qms} \right\}, \forall q,r,n,s,(q \in T, r \in (Ot_i - Ot_i^u)) \tag{35}$$

$$MAT_{ns} \geq 0, \forall q,r,n,m,s,v \tag{36}$$

$$ST_{qr} \geq \max \left\{ ET, MAT_{ns} \times Z_{qms} \right\}, \forall q,r,n,s,(q \in (T' \cup T^u), r \in Ot_q) \tag{37}$$

$$X_{qm} = X^*_{qm}, \forall q,r,(q \in T, r \in Ot_q) \tag{38}$$

These two strategies offer distinct trade-offs between schedule stability and responsiveness. A detailed comparison of their impacts on performance indicators is presented in the following sections.

5. Solution methodology

MAs are well-suited for combinatorial optimization problems, such as the proposed scheduling problem, by combining global exploration with local exploitation to navigate large search spaces and obtain high-quality solutions [39]. Accordingly, five MAs are employed and six hybrid approaches are developed in this work. This section describes the representation strategy and key features of the metaheuristics.

5.1. Solution representation

Effective solution encoding is essential in metaheuristic optimization [40]. For permutation-based combinatorial problems, a widely used approach is the Random-Key method [41, 42]. This study employs this method to encode both machine assignment and operation sequencing. The decoding process consists of two main stages: machine allocation and operation sequencing.

In the first stage, each gene in the chromosome represents an operation and holds a value (ω_{ij}) in [0,1), which determines the assigned machine from its eligible set using the formula $\lfloor CM_{ir} \times \omega_{ij} \rfloor + 1$, where CM_{ir} is number of available machines for operation O_{ir}. For instance, if the third operation of the first task has a candidate machine set $AS_{13} = \{M_{12}, M_{15}, M_{42}, M_{33}\}$, and the corresponding gene value is 0.361, the selected index is $\lfloor 4 \times 0.361 \rfloor + 1 = 2$, and thus the machine M_{15} is allocated to the operation (Figure 2).

In the sequencing stage, given the sequential nature of the tasks, operations with higher priority are scheduled first, and operations assigned to the same resource are prioritized based on gene values, with smaller values scheduled earlier.

5.2. Applied metaheuristic algorithms

To establish the baseline for our hybrid framework, three single-solution metaheuristics, Tabu Search (TS) [43], Simulated Annealing (SA) [44], and the Social Engineering Optimizer (SEO) [45], are employed. TS uses adaptive memory to avoid revisiting solutions and escapes local optima via a tabu list [46]. SA explores neighbors and accepts worse solutions based on Boltzmann probability to escape local optima [47]. SEO, inspired by social manipulation, iteratively updates a defender under an attacker's influence, with roles exchanged to enhance exploration [48]. These algorithms excel at refining solutions and maintaining strong local search.

In parallel, two population-based metaheuristics, the Keshtel Algorithm (KA) [49] and Genetic Algorithm (GA) [50], are applied for global search. KA models Keshtel ducks' foraging by dividing solutions into elite, intermediate, and randomly regenerated groups [51]. GA improves populations through selection, crossover, and mutation, maintaining diversity and improving quality [52]. These methods provide complementary exploration and exploitation mechanisms for complex combinatorial optimization problems. Algorithm steps appear in Supplementary material, Appendix A, Figures A.1-A5.

Following the description of the applied metaheuristics, it is essential to justify their selection based on evidence from existing literature in the scheduling domain. GA, TS, and SA are widely recommended for complex scheduling [53, 54], while KA and SEO have shown strong performance in combinatorial scheduling problems [49, 51]. Accordingly, these metaheuristics were combined to develop efficient hybrid algorithms for scheduling and rescheduling tasks.

5.3. Hybrid Algorithms

MAs are guided by two complementary goals: exploration (global search) and exploitation (local refinement) [39]. PMs like GA and KA are effective in exploration due to stochastic operators and population diversity, yet they often lack sufficient intensification. In contrast, SMs such as SA, TS, and SEO excel in exploitation, enhancing local search efficiency.

To leverage the strengths of both, this study adopts the Low-Level Teamwork Hybrid (LTH) approach [15], wherein SMs are integrated into PMs at the algorithmic operation level to strengthen local search without compromising global diversity. This research employed the LTH approach and presents six developed hybrid MAs, which are applied for the first time to the addressed DSPCMg. The detailed explanations of hybrid algorithms are provided in the following subsections.

5.3.1. Hybrid algorithms with KA

The KA, with its structured population-based search and subgroup classifications, offers strong exploration through population diversity and swirling mechanisms [49]. However, its local search capabilities remain limited when tackling complex scheduling landscapes. To enhance intensification, this study integrates local search MAs, SA, SEO, and TS into KA's framework. The integration is performed within the N1 subgroup, responsible for fine-tuning high-quality candidates. These hybrid structures, namely KASA, KATS, and KASEO, capitalize on KA's broad exploration while leveraging SMs' targeted exploitation abilities to intensify promising regions of the search space. As a result, these hybrids strike an effective balance between global diversification and local convergence. Detailed pseudocodes are provided in Supplementary material, Appendix A, Figures A6-A8.

5.3.2. Hybrid algorithms with GA

GA is widely known for its robust diversification mechanisms, utilizing crossover and mutation to explore the search space [41]. Despite GA's strong abilities, its performance tends to slow down in the absence of effective local search. To overcome this, the proposed hybrids introduce SA, TS, and SEO into GA's workflow, forming GASA, GATS, and GASEO. These GA-based hybrids incorporate SMs as local search strategies during offspring evaluation. This enables the GA to maintain diversity while enhancing search intensity within high-potential areas [55]. The synergy between stochastic global search and strategic local refinement results in improved performance on the DSPCMg problem. The corresponding pseudocode is presented in Supplementary material, Appendix A, Figures A9-A11.

6. Computational results

Computational experiments were performed to both assess the performance of the hybrid MAs and validate the extended scheduling model. In the following subsections, the generation of the numerical instances, algorithm tuning, statistical analyses of algorithms, development of benchmark instances, and sensitivity analyses are discussed. Additional technical materials are provided in Supplementary material to support the experimental analysis. Appendix B details the structure of problem instances and parameter settings and Appendix C reports extended computational results.

6.1. Experimental setup

To comprehensively assess the model's scalability, a diverse set of fifteen problem instances in three categories: small, medium, and large sizes, was randomly generated. Instances are characterized by the number of tasks (I), factories (N), all operations ($R = \sum_i R_i$), all machines ($S = \sum_n S_n$), the maximum number of operation types (O), and the number of new tasks (P). In Supplementary material, Appendix B, Table B1 summarizes the structure and size of each generated problem instance. The generation process is based on [14] and detailed in Supplementary material, Appendix B, Table B2.

In addition, to enhance solution quality, the parameters of the metaheuristic algorithms were optimized using the Taguchi method, which allows efficient tuning of multiple parameters through minimal experimentation [56, 57]. An appropriate orthogonal array was selected based on parameter counts and levels, and three representative instances (Problems

3, 8, and 13) were used to reflect different problem scales. Objective function (OF) values were normalized through Relative Percentage Deviation (RPD), as defined in Equation (39), and the average value was used as the performance indicator. Final parameter levels for each algorithm are provided in Supplementary material, Appendix B, Table B3.

$$RPD = \frac{(Sol_{Alg} - Sol_{min})}{(Sol_{max} - Sol_{min})} \tag{39}$$

6.2. Statistical comparison of metaheuristic algorithms

A detailed examination of the experimental findings derived from the proposed algorithms is provided in this section. To assess the performance of the metaheuristic approaches, each method was independently executed 30 times per instance, with average OF and RPD values serving as the key evaluation metrics. Results across different instance sizes are summarized in Supplementary material, Appendix B, Tables B4–B6, with visual comparisons shown in Figures 3 and 4, distinguishing performance across small, medium, and large instances as well as problem types (scheduling vs. rescheduling).

The results indicate that KATS exhibited the best performance in scheduling tasks, whereas GATS outperformed others in rescheduling scenarios. Interval plots of RPD values (Figure 5) further illustrate algorithmic behavior across different problem categories. The results indicate that among the baseline algorithms (SMs and PMs), SMs consistently yielded better results than population-based methods. However, when hybrid algorithms were introduced, significant improvements were observed. KATS, GATS, and KASA led the scheduling category, while GATS, KATS, and TS performed best in rescheduling. Notably, KA consistently produced the weakest results. To ensure the statistical validity of these observations, additional analysis was conducted.

Moreover, these findings underscore the effectiveness of hybridization, where the exploratory strength of PMs (GA, KA) is complemented by the intensification power of SMs (SA, TS, SEO). Hybrids such as GATS and KATS outperformed their respective standalone components; GATS and KATS surpassed TS; GASA and KASA exceeded SA; and GASEO and KASEO outperformed SEO.

First, Levene's Test was applied to assess homogeneity of variances. Due to the violation of this assumption, the non-parametric Friedman test was used to rank algorithm performance. As shown in Table 2, KATS ranked highest for scheduling problems, followed by GATS, while this order was reversed in rescheduling problems. When considering all instances, GATS emerged as the overall best performer, offering a balanced approach across both problem types.

In summary, while KATS is the best choice for pure scheduling problems, the hybrid approach employed in GATS makes it an effective method for solving the rescheduling problems. Moreover, the general applicability and adaptability of GATS suggest that it may be the preferred algorithm when a single method is to be employed for solving both scheduling and rescheduling problems.

6.3. Algorithm evaluation on benchmarks

To address the lack of suitable benchmarks for the proposed scheduling problem, characterized by delivery time windows, SDST, logistics considerations, and task arrival time, a new benchmark set (BMT01–BMT30) was developed based on [58]. These instances, divided into small, medium, and large sizes, vary in the number of tasks, factories, operations, and machines ($i \times n \times r \times s$). Detailed structures are provided in Supplementary material, Appendix C, Table C1.

Besides, for parameters not explicitly defined in the original benchmark, such as intermediate and final logistics times, SDST, and task arrival times, relevant ranges reported in [14], which addresses similar production settings, were used. The SeT_{runs}, MTT_{nm}, and FTT_{ni} follow uniform distributions over the intervals [5, 15], [30,180], and [30, 300], respectively. Furthermore, task arrival times were modeled using an exponential distribution with a mean of 120. Finally, the number of new tasks was assigned up to {1, 2}, {2, 3}, and {3, 4, 5, 6} for small-, medium-, and large-sized instances, respectively.

Summary results are presented in Supplementary material, Appendix C, Tables C2 and C3 and visualized in Figures 6 and 7. Figure 6 compares the performance of algorithms for 30 scheduling-stage benchmark instances, while Figure 7 presents the corresponding results for 30 rescheduling-stage instances, both based on average objective function values computed over 30 independent runs. The findings confirm the algorithmic trends previously discussed that KATS excels in scheduling scenarios due to its effective use of the tabu mechanism, while GATS outperforms others in rescheduling problems, highlighting its balanced search dynamics under dynamic conditions. To assess the effectiveness of the rescheduling strategy, the next section compares rescheduling strategies described in section 3.3.

6.4. Dynamic scheduling

The evaluation of the performance of the proposed rescheduling strategy (Strategy 2) against a first reactive approach (Strategy 1) is summarized in Supplementary material, Appendix D, Table D1. As seen in Figures 8 and 9, Strategy 2 significantly outperforms Strategy 1 by enabling dynamic machine reassignment and resequencing. On average, it reduces total delivery deviation and average machine idle time by 45.45% and 32.48%, respectively. In contrast, the rigidity of Strategy 1 leads to underutilized resources and longer idle periods. These results highlight the advantages of rescheduling problems in improving overall system efficiency and responsiveness.

6.5. Sensitive analysis

The robustness of the proposed models is assessed against changes in the delivery time window, logistics durations, and the number of newly arriving tasks. Figure 10 illustrates, a ±30% variation in delivery time window, logistics times significantly impact delivery deviation and logistics duration. An increase in the delivery time window reduces deviation to nearly zero at +30% by offering greater scheduling flexibility. In contrast, a -30% reduction raises deviation above 1600 units due to tighter constraints. Meanwhile, logistics times increase with higher parameter values, growing from about 100 units at -30% to over 700 units at +30%, indicating longer delivery durations and greater difficulty in meeting deadlines.

Figure 11 shows the effect of increasing newly added tasks on model performance. A sharp, nonlinear increase in the OF is observed, from under 500 with 2 new tasks to nearly 5000 when the number reaches 12. As the number of new tasks increases, the complexity of scheduling and resource allocation also rises and leading to a significant increase in delivery deviation, especially when resources are limited or variable. This behavior underlines the system’s sensitivity to this dynamic event and the importance of adaptable scheduling mechanisms.

6.6. Managerial insights

Insights from the sensitivity analysis provide practical guidance for improving operational efficiency and minimizing delivery deviations in distributed environments such as CMg and shared manufacturing. The results suggest that a greater delivery window makes it easier to coordinate the operations across various machines in different factories and reduces the chances of delays. For instance, increasing the delivery time window by 30% led to a sharp decrease in delivery deviation, from over 1600 units to nearly zero, demonstrating the significant scheduling flexibility it provides. Conversely, tighter windows (+30% to −30%) impose rigid constraints, increasing delivery deviation by over 1600 units. Managers should therefore identify high-risk, tight-deadline orders early and allocate extra logistical or scheduling buffers to prevent failures in delivery targets.

Moreover, the findings showed that delivery deviation increases approximately from 100 to 800 when logistics times rise by 30%. This negative impact of underscore the need for stable and efficient transportation strategies, such as optimizing distribution hubs or collaborating with reliable transportation providers. Additionally, the arrival of new tasks intensifies scheduling complexity, potentially leading to resource conflicts and delivery delays. Hence, prioritization mechanisms and task classification based on urgency can mitigate these effects. Finally, due to the system's inherent sensitivity to dynamic changes, static scheduling proves inadequate. Reactive rescheduling strategies are essential for maintaining performance under uncertainty, enabling timely adjustments and better resource utilization.

7. Conclusion

This study presents a DSPCMg, incorporating practical constraints such as delivery time windows, SDST, final logistics, and task arrival times. To address the model's NP-hard nature, six hybrid MAs were developed by integrating GA and KA with SMs (SA, TS, SEO), and benchmarked against five baseline methods.

Experimental evaluations based on benchmark instances and Friedman test revealed that hybridization significantly improves algorithms' performance. Specifically, KATS was most effective for scheduling problems, while GATS excelled in rescheduling scenarios. Notably, the proposed rescheduling approach outperformed the Strategy 1, with fixed initial schedule, by reducing total delivery time deviation by up to 45% and machine idle time by 32%, emphasizing the value of adaptive scheduling strategies in dynamic environments.

The findings underscore the importance of reactive scheduling in dynamic CMg environments. Future research could focus on enhancing the proposed dynamic scheduling framework by incorporating stochastic elements to better capture uncertainties in processing times and logistics operations. Additionally, integrating machine learning techniques to guide adaptive metaheuristic hybridization can improve search efficiency and responsiveness in highly dynamic cloud manufacturing environments. Moreover, expanding the model to multi-objective formulations would allow simultaneous optimization of delivery time, energy consumption, and operational costs, aligning with sustainable manufacturing goals. Besides, exploring alternative reactive scheduling strategies, including hybrid reactive-predictive approaches and decentralized decision-making, could further improve flexibility and scalability. Finally, validating the proposed methods through real-world industrial implementations would provide critical insights for practical deployment and continuous improvement.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Notations

The current study employed AI tools for passage checking and grammatical revision. We guarantee the validity of the content, and its accuracy has been verified through human expertise.

Appendices

Appendices are presented in the supplementary material.

References

[1] Ji, T., and Xu, X. "Exploring the Integration of cloud manufacturing and cyber-physical systems in the era of industry 4.0 – An OPC UA approach." *Robotics and Computer-Integrated Manufacturing*, **93**, p. 102927 (2025). https://doi.org/10.1016/j.rcim.2024.102927

[2] Ghomi, E. J., Rahmani, A. M., and Qader, N. N. "Cloud manufacturing: challenges, recent advances, open research issues, and future trends." *The International Journal of Advanced Manufacturing Technology*, **102**, pp. 3613–3639 (2019). https://doi.org/10.1007/s00170-019-03398-7

[3] Gharibvand, V., Kolamroudi, M. K., Zeeshan, Q., et al. "Cloud based manufacturing: A review of recent developments in architectures, technologies, infrastructures, platforms and associated challenges." *The International Journal of Advanced Manufacturing Technology*, **131**, pp. 93–123 (2024). https://doi.org/10.1007/s00170-024-12989-y

[4] Čustović, I., Cao, J., and Hall, D. M. "Cloud manufacturing for industrialized construction: Opportunities and challenges for a new manufacturing model." *Journal of Infrastructure Intelligence and Resilience*, **2**(1), p. 100027 (2023). https://doi.org/10.1016/j.iintel.2023.100027

[5] Duran, E., Ozturk, C., and O'Sullivan, B. "Planning and scheduling shared manufacturing systems: key characteristics, current developments and future trends." *International Journal of Production Research*, **63**(13), pp. 4958–4990. (2024). https://doi.org/10.1080/00207543.2024.2442549

[6] Liu, S., Deng, Q., Liu, X., et al. "Dual-service integrated scheduling of manufacturing and logistics for multiple tasks in cloud manufacturing." *Expert Systems with Applications*, **235**, p. 121129 (2024). https://doi.org/10.1016/j.eswa.2023.121129

[7] Ren, L., Zhang, L., Wang, L., et al. "Cloud manufacturing: key characteristics and applications." *International Journal of Computer Integrated Manufacturing*, **30**(6), pp. 501–515 (2017). https://doi.org/10.1080/0951192X.2014.902105

[8] Erden, C., Demir, H. I., and Canpolat, O. "A modified integer and categorical PSO algorithm for solving integrated process planning, dynamic scheduling and due date assignment problem." *Scientia Iranica*, **30**(2), pp. 738–756 (2021). https://doi.org/10.24200/sci.2021.55250.4130

[9] Jing, W., Zhang, Y., Chen, Y., et al. "Cloud-edge collaboration composition and scheduling for flexible manufacturing service with a multi-population co-evolutionary algorithm." *Robotics and Computer-Integrated Manufacturing*, **90**, p. 102814 (2024). https://doi.org/10.1016/j.rcim.2024.102814

[10] Liu, J., Sun, B., Li, G., et al. "Multi-objective adaptive large neighbourhood search algorithm for dynamic flexible job shop schedule problem with transportation resource." *Engineering Applications of Artificial Intelligence*, **132**, p. 107917 (2024). https://doi.org/10.1016/j.engappai.2024.107917

[11] Elgendy, A., Yan, J., and Zhang, M. "A Parallel distributed genetic algorithm using Apache Spark for flexible scheduling of multitasks in a cloud manufacturing environment." *International Journal of Computer Integrated Manufacturing*, **37**(5), pp. 652–667 (2024). https://doi.org/10.1080/0951192X.2023.2228277

[12] Lei, W., Yang, L., Yan, P., et al. "Production coordination of local and cloud orders in shared manufacturing: a bi-objective pre-scheduling approach." *Annals of Operations Research*, **345**, pp. 207–245 (2025). https://doi.org/10.1007/s10479-024-06380-z

[13] Tariq, A., Khan, S. A., But, W. H., et al. "An IoT-Enabled Real-Time Dynamic Scheduler for Flexible Job Shop Scheduling (FJSS) in an Industry 4.0-Based Manufacturing Execution System (MES 4.0)." *IEEE Access*, **12**, pp. 49653–49666 (2024). https://doi.org/10.1109/ACCESS.2024.3384252

[14] Zhang, Y., Zhang, Z., Lu, Y., et al. "Dynamic decision-making for knowledge-enabled distributed resource configuration in cloud manufacturing considering stochastic order arrival." *Robotics and Computer-Integrated Manufacturing*, **87**, p. 102712 (2024). https://doi.org/10.1016/j.rcim.2023.102712

[15] Talbi, E.-G. "A Taxonomy of Hybrid Metaheuristics." *Journal of Heuristics*, **8**, pp. 541–564 (2002). https://doi.org/10.1023/A:1016540724870

[16] Zhang, Y., Wang, J., Liu, S., et al. "Game Theory Based Real-Time Shop Floor Scheduling Strategy and Method for Cloud Manufacturing." *International Journal of Intelligent Systems*, **32**(4), pp. 437–463 (2017). https://doi.org/10.1002/int.21868

[17] Ding, J., Wang, Y., Zhang, S., et al. "Robust and stable multi-task manufacturing scheduling with uncertainties using a two-stage extended genetic algorithm." *Enterprise Information Systems*, **13**(10), pp. 1442–1470 (2019). https://doi.org/10.1080/17517575.2019.1656290

[18] Yuan, M., Cai, X., Zhou, Z., et al. "Dynamic service resources scheduling method in cloud manufacturing environment." *International Journal of Production Research*, **59**(2), pp. 542–559 (2019). https://doi.org/10.1080/00207543.2019.1697000

[19] Yang, C., Peng, T., Lan, S., et al. "Towards IoT-enabled dynamic service optimal selection in multiple manufacturing clouds." *Journal of Manufacturing Systems*, **56**, pp. 213–226 (2020). https://doi.org/10.1016/j.jmsy.2020.06.004

[20] Zhang, S., Xu, Y., and Zhang, W. "Multitask-oriented manufacturing service composition in an uncertain environment using a hyper-heuristic algorithm." *Journal of Manufacturing Systems*, **60**, pp. 138–151 (2021). https://doi.org/10.1016/j.jmsy.2021.05.012

[21] Dai, Z., Zhang, Z., and Chen, M. "Collaborative task scheduling with new task arrival in cloud manufacturing using improved multi-population biogeography-based optimization." *Journal of Intelligent and Fuzzy Systems*, **41**(2), pp. 3849–3872 (2021). https://doi.org/10.3233/JIFS-201066

[22] Aghili, S. A. S., Valilai, O. F., Haji, A., et al. "Dynamic mutual manufacturing and transportation routing service selection for cloud manufacturing with multi-period service-demand matching." *PeerJ Computer Science*, **7**, pp. 1–30 (2021). https://doi.org/10.7717/peerj-cs.461

[23] Zhang, L., Yang, C., Yan, Y., et al. "Distributed Real-Time Scheduling in Cloud Manufacturing by Deep Reinforcement Learning." *IEEE Transactions on Industrial Informatics*, **18**(12), pp. 8999–9007 (2022). https://doi.org/10.1109/TII.2022.3178410

[24] Wang, X., Zhang, L., Liu, Y., et al. "Dynamic scheduling of tasks in cloud manufacturing with multi-agent reinforcement learning." *Journal of Manufacturing Systems*, **65**, pp. 130–145 (2022). https://doi.org/10.1016/j.jmsy.2022.08.004

[25] Xiong, W., Lim, M. K., Tseng, M.-L., et al. "An effective adaptive adjustment model of task scheduling and resource allocation based on multi-stakeholder interests in cloud manufacturing." *Advanced Engineering Informatics*, **56**, p. 101937 (2023). https://doi.org/10.1016/j.aei.2023.101937

[26] Zhang, Z., Qu, T., Zhang, K., et al. "Digital twin-based production logistics resource optimisation configuration method in smart cloud manufacturing environment." *IET Collaborative Intelligent Manufacturing*, **6**(4), e12118 (2024). https://doi.org/10.1049/cim2.12118

[27] Hu, Y., Pan, L., and Pan, X. "Dynamic scheduling of workshop resource in cloud manufacturing environment." *Engineering Applications of Artificial Intelligence*, **138**, p. 109405 (2024). https://doi.org/10.1016/j.engappai.2024.109405

[28] Wang, X., Zhang, L., Liu, Y., et al. "An improved deep reinforcement learning-based scheduling approach for dynamic task scheduling in cloud manufacturing." *International Journal of Production Research*, **62**(11), pp. 4014–4030 (2024). https://doi.org/10.1080/00207543.2023.2253326

[29] Xiao, J., Cai, Y., and Chen, Y. "Study on deep reinforcement learning for multi-task scheduling in cloud manufacturing." *International Journal of Computer Integrated Manufacturing*, pp. 1–18 (2025). https://doi.org/10.1080/0951192X.2025.2452981

[30] Shao, L., and Ren, M. "A blockchain-based self-adaptive collaboration mechanism for service-oriented manufacturing system." *International Journal of Computer Integrated Manufacturing*, pp. 1–19 (2025). https://doi.org/10.1080/0951192X.2025.2452609

[31] Xu, L., Zhou, T., Li, K., et al. "Q-learning-driven multi-population cooperative evolutionary algorithm with local search for scheduling of network-shared manufacturing resources." *Computers & Operations Research*, **180**, p. 107076 (2025). https://doi.org/10.1016/j.cor.2025.107076

[32] Salmasnia, A., Kiapasha, Z., and Pashaeenejad, M. "Subtasks scheduling of tasks with different structures in cloud manufacturing systems under maintenance policy and focusing on logistics, tardiness, and earliness aspects." *Operational Research*, **24**(3), p. 48 (2024). https://doi.org/10.1007/s12351-024-00857-2

[33] Akbaripour, H., Houshmand, M., van Woensel, T., et al. "Cloud manufacturing service selection optimization and scheduling with transportation considerations: mixed-integer programming models." *The International Journal of Advanced Manufacturing Technology*, **95**, pp. 43–70 (2018). https://doi.org/10.1007/s00170-017-1167-3

[34] Zhao, Z., Zhou, H., and Zheng, W. "Collaborative optimization for scheduling manufacturing tasks and transport vehicles considering manufacturer's time availability in cloud manufacturing." *International Journal of Computer Integrated Manufacturing*, pp. 1–26 (2024). https://doi.org/10.1080/0951192X.2024.2382211

[35] Helo, P., Phuong, D., and Hao, Y. "Cloud manufacturing – Scheduling as a service for sheet metal manufacturing." *Computers and Operations Research*, **110**, pp. 208–219 (2019). https://doi.org/10.1016/j.cor.2018.06.002

[36] Zhu, H., Li, M., Tang, Y., et al. "A Deep-Reinforcement-Learning-Based Optimization Approach for Real-Time Scheduling in Cloud Manufacturing." *IEEE Access*, **8**, pp. 9987–9997 (2020). https://doi.org/10.1109/ACCESS.2020.2964955

[37] Chen, M., Xu, J., Zhang, W., et al. "A new customer-oriented multi-task scheduling model for cloud manufacturing considering available periods of services using an improved hyper-heuristic algorithm." *Expert Systems with Applications*, **269**, p. 126419 (2025). https://doi.org/10.1016/j.eswa.2025.126419

[38] Kahneman, D., and Tversky, A. "Prospect theory: An analysis of decision under risk". *In Handbook of the fundamentals of financial decision making: Part I*, pp. 99-127 (2013), https://doi.org/10.1142/9789814417358_0006

[39] Rajabi-Kafshgar, A., Gholian-Jouybari, F., Seyedi, I., et al. "Utilizing hybrid metaheuristic approach to design an agricultural closed-loop supply chain network." *Expert Systems with Applications*, **217**, p. 119504 (2023). https://doi.org/10.1016/j.eswa.2023.119504

[40] Fathollahi Fard, A. M., Gholian-Jouybari, F., Paydar, M. M., et al. "A Bi-Objective Stochastic Closed-loop Supply Chain Network Design Problem Considering Downside Risk." *Industrial Engineering & Management Systems*, **16**(3), pp. 342–362 (2017). https://doi.org/10.7232/iems.2017.16.3.342

[41] Rajabi-Kafshgar, A., Seyedi, I., and Tirkolaee, E. B. "Circular closed-loop supply chain network design considering 3D printing and PET bottle waste." *Environment, Development and Sustainability*, pp. 1–37 (2024). https://doi.org/10.1007/s10668-024-04767-3

[42] Hajiaghaei-Keshteli, M. "The allocation of customers to potential distribution centers in supply chain networks: GA and AIA approaches." *Applied Soft Computing*, **11**(2), pp. 2069–2078 (2011). https://doi.org/10.1016/j.asoc.2010.07.004

[43] Glover, F. "Future paths for integer programming and links to artificial intelligence." *Computers & Operations Research*, **13**(5), pp. 533–549 (1986). https://doi.org/10.1016/0305-0548(86)90048-1

[44] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. "Optimization by simulated annealing." *Science*, **220**(4598), pp. 671–680 (1983). https://doi.org/10.1126/science.220.4598.671

[45] Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., and Tavakkoli-Moghaddam, R. "The Social Engineering Optimizer (SEO)." *Engineering Applications of Artificial Intelligence*, **72**, pp. 267–293 (2018). https://doi.org/10.1016/j.engappai.2018.04.009

[46] Behmanesh, R., and Kianfar, K. "A new mathematical formulation and a hybrid evolutionary algorithm for re-entrant flow-shop problem with release date." *Scientia Iranica*, **31**(6), pp. 500–517 (2024). https://doi.org/10.24200/sci.2021.56555.4787

[47] Ghiasvand Ghiasi, F., Yazdani, M., Vahdani, B., et al. "Multi-depot home health care routing and scheduling problem with multimodal transportation: Mathematical model and solution methods." *Scientia Iranica*, **31**(11), pp. 825–846 (2024). https://doi.org/10.24200/sci.2021.57338.5183

[48] Aghamohamadi-Bosjin, S., Rabbani, M., and Manavizadeh, N. "A hybrid metaheuristic algorithm for data driven leagile sustainable closed-loop supply chain modeling under disruption risk." *Scientia Iranica*, **29**(3), pp. 1685–1704 (2022). https://doi.org/10.24200/sci.2020.53949.3506

[49] Hajiaghaei-Keshteli, M., and Aminnayeri, M. "Solving the integrated scheduling of production and rail transportation problem by Keshtel algorithm." *Applied Soft Computing Journal*, **25**, pp. 184–203 (2014). https://doi.org/10.1016/j.asoc.2014.09.034

[50] Holland, J. H. "Genetic Algorithms." Scientific American, **267**(1), pp. 66–72 (1992). https://doi.org/10.1038/scientificamerican0792-66

[51] Goli, A., Ala, A., and Hajiaghaei-Keshteli, M. "Efficient multi-objective meta-heuristic algorithms for energy-aware non-permutation flow-shop scheduling problem." *Expert Systems with Applications*, 213, p. 119077 (2023). https://doi.org/10.1016/j.eswa.2022.119077

[52] Ayyoubzadeh, B., Ebrahimnejad, S., Bashiri, M., et al. "A reactive approach for flexible job shop scheduling problem with tardiness penalty under uncertainty." *Scientia Iranica*, (2022). https://doi.org/10.24200/sci.2022.58491.5754

[53] Agnetis, A., Billaut, J. C., Pinedo, M., et al. "Fifty years of research in scheduling — Theory and applications." *European Journal of Operational Research*, **327**(2), pp. 367-393 (2025). https://doi.org/10.1016/j.ejor.2025.01.034

[54] Ahmadian, M. M., Khatami, M., Salehipour, A., et al. "Four decades of research on the open-shop scheduling problem to minimize the makespan." *European Journal of Operational Research*, **295**(2), pp. 399–426 (2021). https://doi.org/10.1016/j.ejor.2021.03.026

[55] Hamdi-Asl, A., Amoozad-Khalili, H., Tavakkoli-Moghaddam, R., et al. "Toward sustainability in designing agricultural supply chain network: A case study on palm date." *Scientia Iranica*, **31**(18), pp. 1691–1709 (2021). https://doi.org/10.24200/sci.2021.58302.5659

[56] Golmohamadi, S., Tavakkoli-Moghaddam, R., and Hajiaghaei-Keshteli, M. "Solving a fuzzy fixed charge solid transportation problem using batch transferring by new approaches in meta-heuristic." *Electronic Notes in Discrete Mathematics*, **58**, pp. 143–150 (2017). https://doi.org/10.1016/j.endm.2017.03.019

[57] Delavar, M. R., Hajiaghaei-Keshteli, M., and Molla-Alizadeh-Zavardehi, S. "Genetic algorithms for coordinated scheduling of production and air transportation." *Expert Systems with Applications*, **37**(12), pp. 8255–8266 (2010). https://doi.org/10.1016/j.eswa.2010.05.060

[58] Zhu, K., Gong, G., Peng, N., et al. "Dynamic distributed flexible job-shop scheduling problem considering operation inspection." *Expert Systems with Applications*, **224**, p. 119840 (2023). https://doi.org/10.1016/j.eswa.2023.119840

Biographies

Atefeh Rajabi-Kafshgar is a Ph.D. candidate at the School of Industrial Engineering, Iran University of Science and Technology. She obtained her M.Sc. degree in Industrial Engineering from Al-Zahra University in Iran. Her research interests include system modeling and intelligent optimization methods, especially in optimization algorithms, and cloud manufacturing system operation management.

Prof. Mostafa Hahiaghaei-Keshteli is a Research Associate Professor at the Department of Industrial Engineering, School of Engineering and Science, Tecnológico de Monterrey, Puebla Campus, Mexico. He is a co-founder and senior researcher at the Center of Sustainable Smart Logistics (CSSL). His main research areas are Sustainable and Smart Logistics, Supply Chain Integration and Optimization, Industry 4.0 and 5.0, and Combinatorial Optimization. He has first designed and developed some food and agricultural supply chains, as well as designing some famous and capable Metaheuristics Algorithms in optimizations such as Keshtel Algorithm (2012), Red Deer Algorithm (2016), Social Engineering Optimizer (2018), and Tree Growth Algorithm (2018). He was ranked as one of the top 10 scholars in the area of Supply Chain and one of the top 2% scientists, according to the SciVal and Scopus reports. According to research.com, he is one of the top 10 scientists in all Engineering fields in Mexico, since 2022.

Dr. Mohammad Reza Mohammad Aliha received his PhD from Iran University of Science and Technology (IUST) in 2009 and he is currently an associate professor at this university. His main research interests cover system modeling, optimization algorithms, and modern manufacturing systems. He has published more than 240 journal papers and has more than 14,000 citations he is now ranked among the top 1% of world scientists.

List of Figures

List of Tables

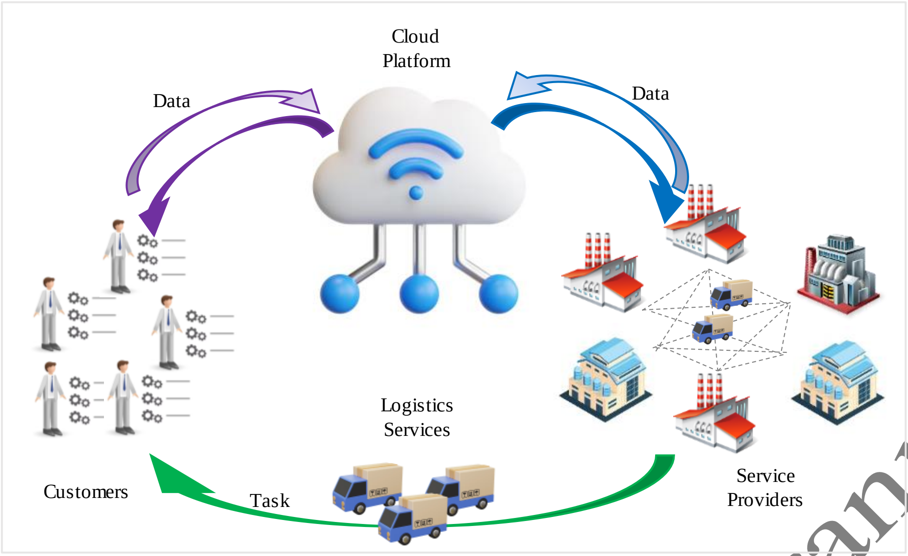

Figure 1. A typical CMg environment.

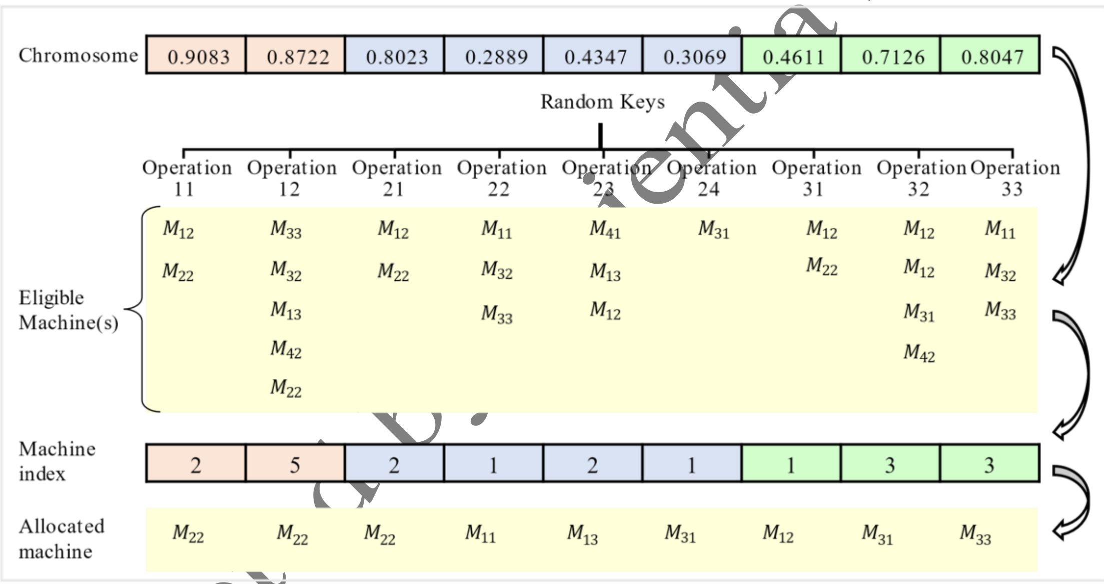

Figure 2. An example of the applied Random-Key representation.

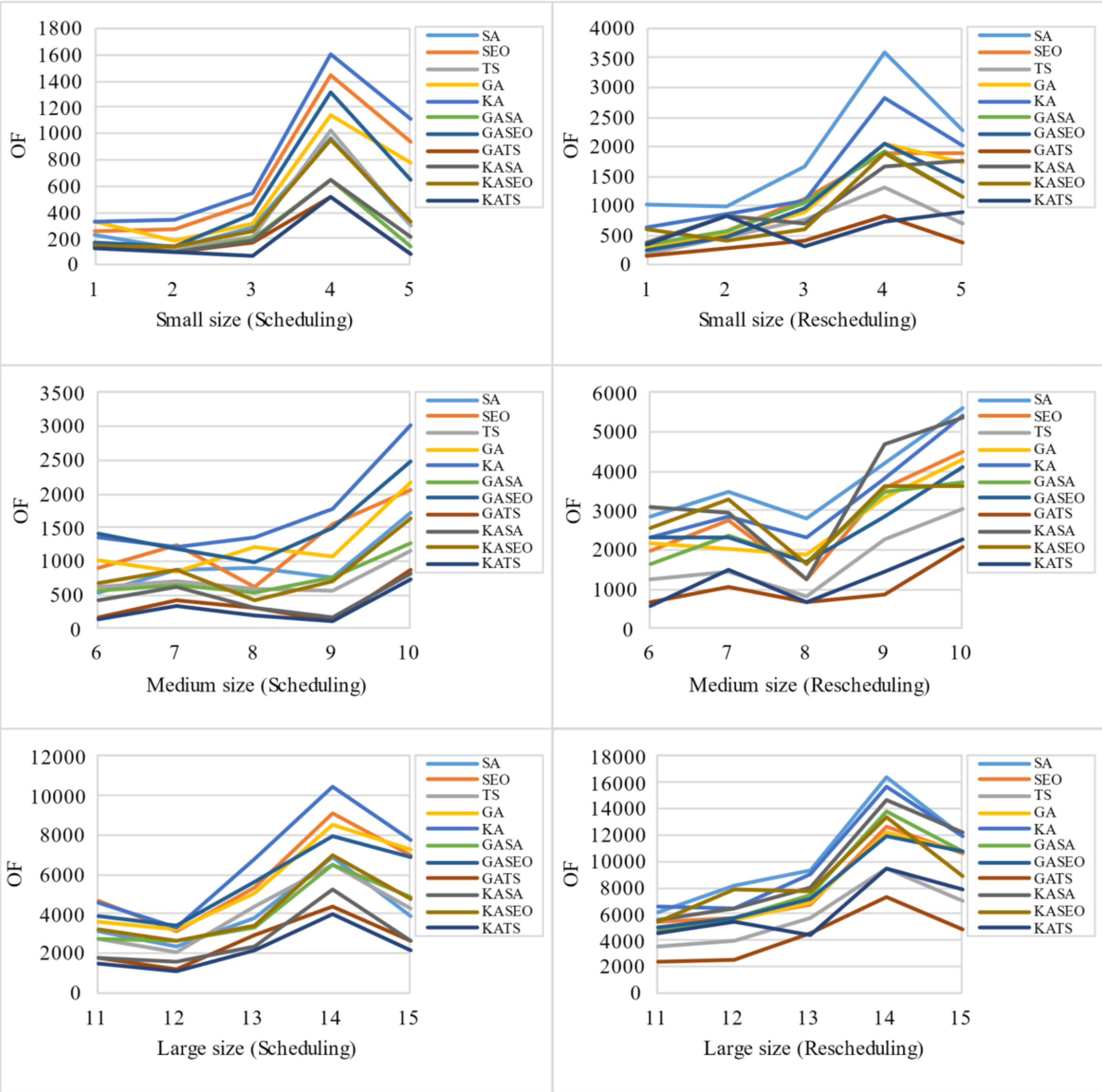

Figure 3. The objective values for different problem sizes before and after rescheduling.

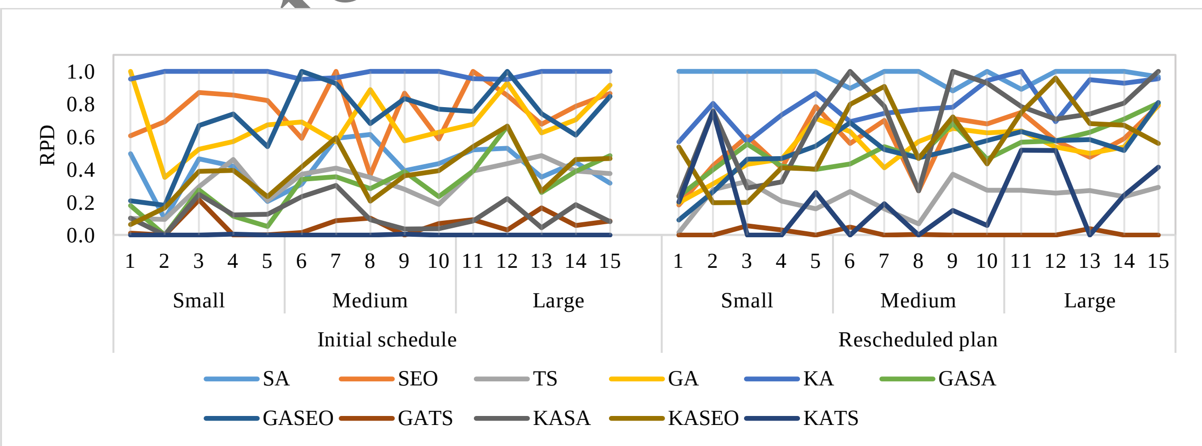

Figure 4. The PRD values for scheduling and rescheduling problems.

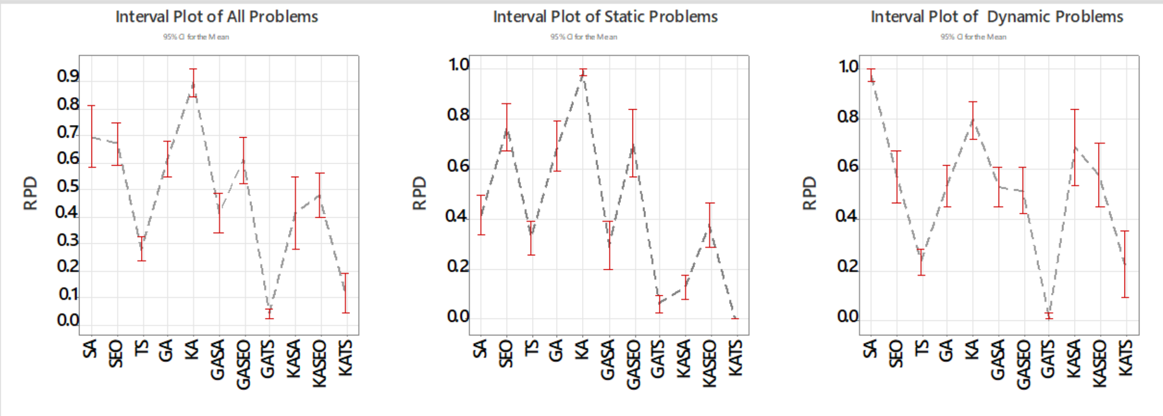

Figure 5. Interval Plot of algorithms for scheduling and rescheduling problems.

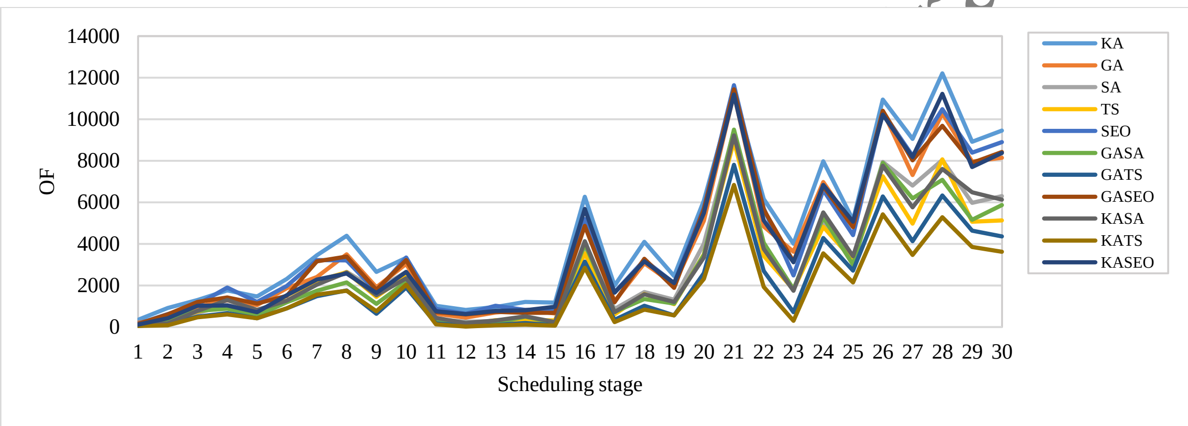

Figure 6. Comparison of applied metaheuristic algorithms on 30 benchmark instances in the scheduling stage, based on average objective function values.

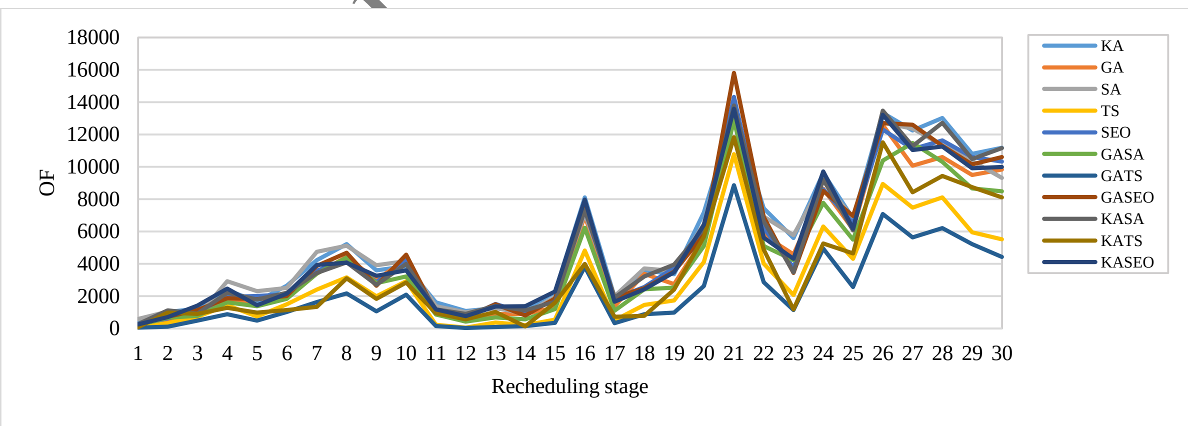

Figure 7. Performance comparison of the applied algorithms on 30 benchmark instances in the rescheduling stage, based on the average objective function values.

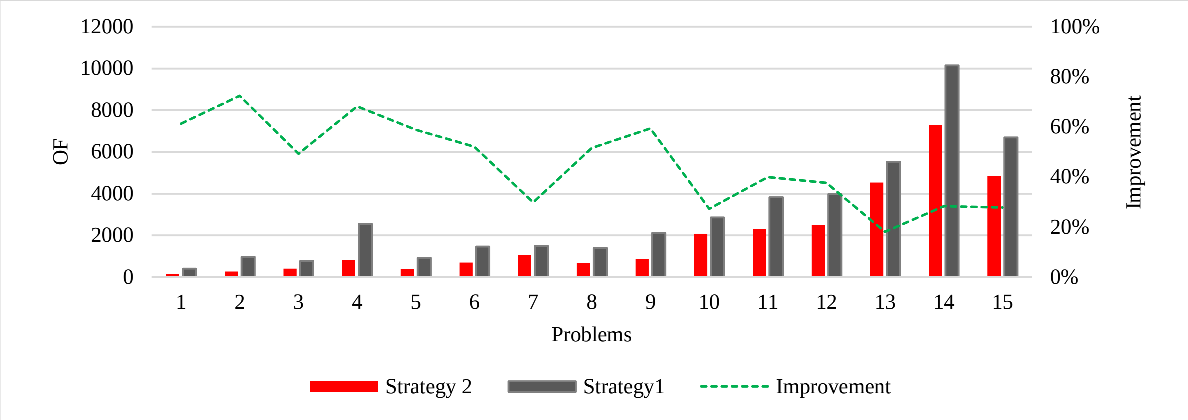

Figure 8. The comparison of delivery time deviation across rescheduling strategies.

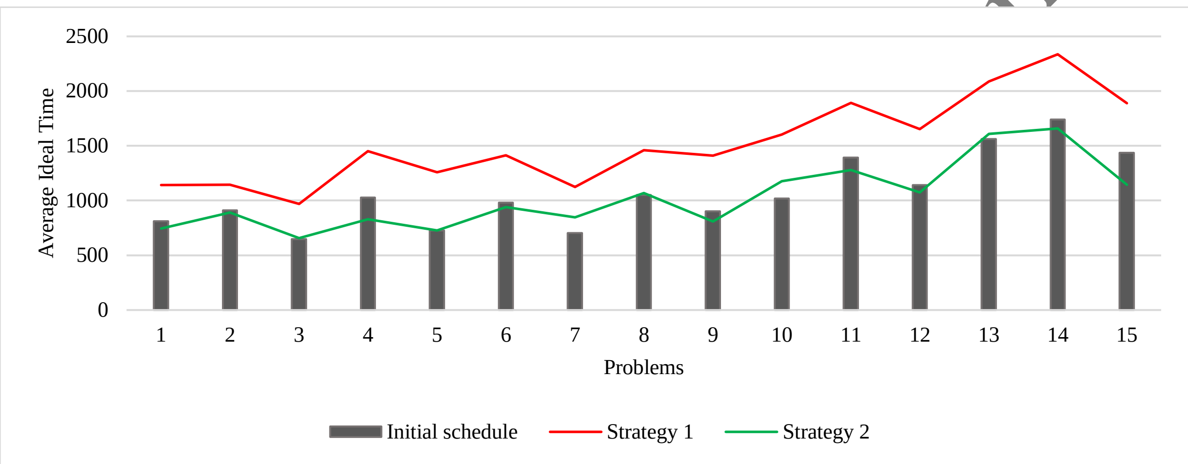

Figure 9. The comparison of machine idle time across rescheduling strategies.

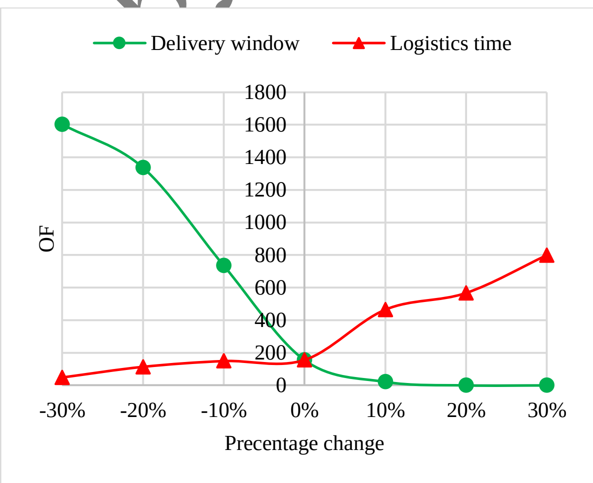

Figure 10. The impact of logistics time and delivery window on OF.

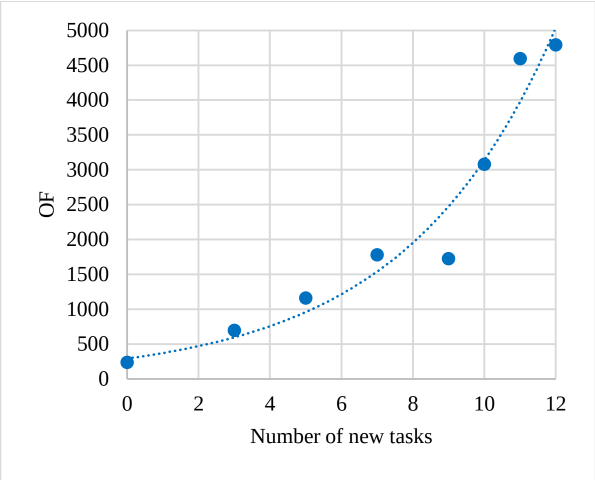

Figure 11. The effect of the number of newly assigned tasks on OF.

Table 1. Model characteristics and solution methodologies of DSPCMg.

Paper	Model characteristics							Dynamic event(s)	Solution approach	
	Objective(s)	MIP	AT	MTT	SeT	FTT	DT		SR	OT
[16]	Utilization rate	*						SB	Game theory	
[17]	Combination of cost, time, reliability, robustness, & stability	*						SB		GA
[35]	Makespan				*			SeT change		GA
[18]	Weighted sum of time, cost, quality, & capability	*		*				Requirement deviation		Hybrid of GA-ACO
[36]	Weighted sum of completion time, utilization, & task overhead		*				*	DTA	RL	
[19]	Completion time & cost	*		*				Delay or early completion & task change		GA
[21]	Completion time, cost, & reliability	*	*	*	*			NT		MPBBO
[20]	Makespan, cost, & reliability	*		*	*			NT & materials delay		GA
[22]	Total cost	*		*				NT		GA
[23]	Average tardiness		*	*			*	DTA		RL-based MDP
[24]	Total makespan			*				SB, new service		RL-based MDP
[25]	Weighted sum of time, cost, & quality Sum of flexibility & user evaluation Provider profit			*				SB, NT, service preemption		Hybrid of HHO and NSGA-II
[27]	Weighted sum of time, cost, quality & utilization		*					DTA, SB, & maintenance		PSO
[14]	Completion time Resource utilization		*	*			*	DTA	RL	
[26]	Weighted sum of cost of vehicle rent, power, & maintenance	*		*				Demand change		TLBO
[9]	Weighted sum of quality, reliability, & total cost & due time deviation	*		*			*	Service quality change		MTMC
[28]	Makespan & cost			*				Service rate change		AC
[29]	Weighted sum of time, cost, & reliability	*		*	*		*	Service unavailability		PPO
[30]	Finish time deviation	*					*	Service & task change		GA
[12]	Makespan Total disturbances		*	*				NT		BHDDE
[31]	Total tardiness	*	*				*	DTA		MPCEA
This work	Total delivery deviation	*	*	*	*	*	*	NT		KATS, GATS, KASA, GASA, KASEO, GASEO

* Mix integer programming (MIP), optimization technique (OT), task arrival (TA), middle transportation time (MTT), setup time (SeT), final transportation time (FTT), due date or delivery time (DT), Service breakdown (SB), Dynamic task arrival (DTA), New task (NT), scheduling rules (SR), Genetic algorithm (GA), meta-knowledge transfer based multi-population co-evolution (MTMC), particle swarm optimization (PSO), teaching -learning based optimization (TLBO), multi-objective elitist Jaya (MOEJ), multi-population biogeography-based optimization (MPBBO), ant colony optimization (ACO), non-dominated sorting GA-II (NSGA-II), harris hawks optimization (HHO), multi-population cooperative evolutionary algorithm (MPCEA), bi-objective hybrid discrete differential evolution (BHDDE), actor-critic (AC), proximal policy optimization (PPO), Markov decision process (MDP), reinforcement learning (RL).

Table 2. Results of ranking the algorithms based on the Friedman test.

Algorithm	All problems		Scheduling problem		Rescheduling problem	
	Median	Rank	Median	Rank	Median	Rank
GA	0.597	7	0.669	8	0.547	5
GASA	0.404	4	0.310	4	0.555	6
GASEO	0.575	8	0.734	9	0.540	4
GATS	0.025	1	0.071	2	0.018	1
KA	0.900	11	0.996	11	0.830	10
KASA	0.330	5	0.126	3	0.728	9
KASEO	0.453	6	0.406	6	0.603	8
KATS	0.052	2	0.000	1	0.197	2
SA	0.677	10	0.424	7	1.000	11
SEO	0.645	9	0.798	10	0.595	7
TS	0.275	3	0.347	5	0.241	3