

# OPTIMAL JOB SCHEDULING TO MINIMIZE TOTAL TARDINESS BY DISPATCHING RULES AND COMMUNITY EVALUATION CHROMOSOMES

Prasad Bari<sup>1,2</sup> and Prasad Karande<sup>1</sup>

<sup>1</sup> Department of Mechanical Engineering, Veermata Jijabai Technological Institute, Mumbai, India

<sup>2</sup> Department of Mechanical Engineering, Fr. C. Rodrigues Institute of Technology, Vashi, Navi-Mumbai, India

Received: 23 February 2023;

Accepted: 2 May 2023;

Available online: 4 July 2023.

*Original scientific paper*

**Abstract:** *In traditional scheduling, job processing times are assumed to be fixed. However, this assumption may not be applicable in many realistic industrial processes. Using the job processing time of real industrial processes instead of a fixed value converts the deterministic model to a stochastic one. This study provides three approaches to solving the problem of stochastic scheduling: stochastic linguistic, stochastic scenarios, and stochastic probabilistic. A combinatorial algorithm, Dispatching Rules and Community Evaluation Chromosomes (DRCEC) is developed to generate an optimal sequence to minimize the tardiness performance measure in the scheduling problem. Thirty-five datasets of scheduling problems are generated and tested with the model. The DRCEC is compared to the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) in terms of total tardiness, the tendency of convergence, execution time, and accuracy. The DRCEC has been discovered to outperform the PSO and GA. The computational results show that the DRCEC approach gives the optimal response in 63 per cent of cases and the near-optimal solution in the remaining 37 per cent of cases. Finally, a manufacturing company case study demonstrates DRCEC's acceptable performance. The use of DRCEC with realistic data from a manufacturing company reveals that the sequence acquired by the model gives the less tardiness value when equated to the company's first come first serve method.*

**Key words:** *Scheduling, sequencing, tardiness, genetic algorithm, dispatching rules.*

\* Corresponding author.

E-mail addresses: \*[prasad.bari@fcrit.ac.in](mailto:prasad.bari@fcrit.ac.in) (P. Bari), [pmkarande@me.vjti.ac.in](mailto:pmkarande@me.vjti.ac.in) (P. Karande)

## 1. Introduction

The scheduling process defines when a specific task can be completed. The scheduling problem's boundary can be described by specifying the resources, the task's time length, when it is expected to begin, and when it is scheduled to end. This decision-making procedure optimizes one or more objectives (Pinedo, 2004). The intention of the scheduling process is thus to reduce the task's end time and the expense associated with performing it. The scheduling idea initially appeared in the mid-twentieth century. Since then, the importance of this issue to industries has grown due to the layout of the shops, shops with multiple identical machines, operations requiring multiple resources at once, or operations utilizing multipurpose machines, all of which increase complexity (T'Kindt & Billaut, 2005).

A scheduling problem requires decisions on work allocation and sequencing. Scheduling is mostly just allocation, and mathematical programming methods are employed to discover the best solution in such instances. On the other hand, very often, scheduling is purely sequencing. As a result, sequencing is a specialized scheduling problem where the job order governs the schedule. Furthermore, simple sequencing is a single-machine problem with deterministic processing times for each work on the machine. Thus, sequencing is used to determine the correct order for a fixed number of different jobs to be performed on a machine. This is useful in determining the appropriate ranking of the jobs so that the overall time spent completing the jobs is minimized and they are not delayed.

Several jobs are waiting to be done on a machine. The problem becomes more challenging to solve as the number of jobs increases and becomes non-deterministic polynomial-time hard (NP-hard). Because the scheduling problem of minimizing tardiness is NP-hard, heuristic and meta-heuristic approaches have been frequently used to solve it, as accurate methods are unfeasible for medium and large cases (Gupta & Chauhan, 2015; Sayadi et al., 2010). These methods require less calculation but do not guarantee optimality; instead, they yield approximately ideal satisfactory answers for real-world applications. Jobs in job shops are thus processed on machines in a predetermined sequence specified by priority criteria, which are simplified heuristics guidelines. The experienced human dispatcher decides to use these guidelines. One of the various priority rules is used to schedule the jobs in the job shop that are due for dispensation. Shortest processing time (SPT) first - efficient for minimizing mean completion time, earliest due date (EDD) first - efficient for minimizing tardiness measures, and first come first served (FCFS) - efficient for minimizing the variance of completion time, are some of the frequently used priority rules for sequencing the jobs. Because the objective function determines which priority rules are used, it is often advantageous to explore the alternative option.

The scheduling models can be classified by recognizing the resource arrangement and the type of jobs (Baker & Trietsch, 2009). Static and Dynamic scheduling are the two types of scheduling problems. In static problems, fixed jobs are available for scheduling, whereas new jobs emerge over time in dynamic problems. In most cases, studying static problems yields valuable insights, which are then applied in dynamic situations using heuristic methods. The shortest processing time rule gives an appropriate sequence for flow time problems in a static scheduling problem. On the other hand, the earliest due date rule generates a suitable sequence for tardiness problems (K. K. Kumar et al., 2017). If a job arrives for operation in a fixed set in a dynamic scheduling problem, it must be determined whether the arriving job should be processed before the following one. It is not required to resort to this set of jobs;

Optimal job scheduling to minimize total tardiness by dispatching rules and community ...

the newly arrived job needs to be compared to the present work with the maximum priority (Baker & Trietsch, 2009). The simplest model for the machine environment is a single machine. There are also multi-machine models, in which machines are either parallel, performing the same functions, or specialized, conducting specific tasks.

Another classification is that the model may be deterministic (where certain assumptions are made with certainty) or stochastic (where uncertainty is known with explicit probability distributions). The deterministic scheduling operation is done considering only the present scenario at hand. This type of scheduling requires merely the jobs' processing time and due date. The due date given by the client remains fixed. The processing time changes depending on the nature of the factors affecting the jobs. Deterministic scheduling does not consider these factors. Hence, deterministic scheduling can be referred to as an idealistic operation. Deterministic scheduling revolves around the seven assumptions (French, 1982). After relaxing one or more assumptions, the model becomes stochastic, considered to be more realistic. In this research, the assumptions that the processing time as a job descriptor is deterministic and the machine is continuously available are relaxed. Heuristic and metaheuristic methods have been frequently used to solve scheduling problems with the least amount of tardiness. The Genetic Algorithm (GA) is the most widely used metaheuristic for solving the scheduling problem to reduce tardiness (Li et al., 2015). GA has a high exploration capacity that allows it to search a wide variety of search spaces. It does not, however, offer a robust search mechanism for precisely looking like a near good solution. As a result, many researchers are enhancing GA's effectiveness by adding various search and heuristic strategies to solve the job shop scheduling problem (JSP). The manuscript is focused on ideas of minimizing the tardiness, application of metaheuristic approaches like GA and PSO on data and improvement in lowering the tardiness and minimizing randomness of GA with a new approach to population selection. An optimization for future improvement has always come with the development of engineering science technologies (Uniyal et al., 2022). In this research work, GA is combined with dispatching rules to find a near-optimal solution by community evaluation and injecting the optimal solution of antecedent iteration to successor iteration.

## 2. Literature Review

The JSP is a well-known type of production scheduling problem. The JSP seeks to find job sequences that best meet specific production goals. A good JSP solution can help manufacturers increase production efficiency while decreasing expenses (Dao et al., 2018; Gong et al., 2019). As a result, the JSP has been the subject of extensive study for decades.

A dispatching rule was introduced in job shop scheduling, which uses a modified due date (MDD) to identify the performance measure of mean tardiness, taking into account the varying pattern of the problem and diverse measures based on flow time and tardiness of jobs (Raman et al., 1989). The authors demonstrated that the best dispatching rule is reversed modified due date (RMDD), whereas the worst is minimum slack time (MST), considering multiple objectives in a scheduling problem. They further stated that MDD, insertion, and greedy rules supported minimizing the total tardiness of jobs which helps in making decisions while scheduling the jobs (Bari & Karande, 2021). A. Kumar et al. (2022) considered a real-world decision-making issue and further explained the weighted sum model, weighted product model, and

weighted aggregated sum-product assessment to help in decision-making. Other than dispatching rules, exact solution methods such as branch and bound algorithm (Carlier & Pinson, 1989) is used to handle the problem. A model built on mixed-integer programming was developed by Yu & Lee (2018). They proposed an approach for reducing total family tardiness in job shop scheduling problems when jobs are organized into families using the branch and bound technique. Martínez et al. (2019) used branch and check to find the best solutions to the production planning problem of the packaging industry. However, as an exact solution approaches, this approach becomes inefficient as the problem grows. The JSP has remained an NP-hard problem, meaning that an optimal solution cannot be found in a finite amount of time (He et al., 2021). Nonetheless, the heuristic approach is preferable for larger problems since it finds near-optimal solutions in less computational time (Garey et al., 1976). The metaheuristic rollout technique was studied and it was observed that heuristics are more successful at directing the rollout algorithm to better options. But so far, in terms of the average quality of solutions found and in worst-case performance, the rollout strategy is significantly more stable than the single standalone heuristics (Meloni et al., 2004).

A heuristic approach for sequencing production orders was applied and aimed to reduce overall tardiness with setup time (Cayo & Onal, 2020). A heuristic algorithm works better than the mathematical model for truck-to-door sequencing (Ardakani et al., 2020). Population-based meta-heuristic algorithms could get an approximate solution in an acceptable amount of time. Parallelism, diversity, robustness, and high compatibility are meta-heuristics features. As a result, meta-heuristic methods have been extensively used to address the JSP in current years. Meta-heuristic algorithms can be thought of as stand-alone approaches (He et al., 2021).

Sergienko et al. (2009) categorized optimization approaches into seven groups. These are 1) sequential algorithms, 2) deterministic local search, 3) stochastic local search (SLS), 4) swarm intelligence (SI), 5) evolutionary algorithm (EA), 6) scanning methods, and 7) other unique methods, comprising exact algorithms. Among these strategies, 3, 4, and 5 are often used in answering scheduling problems, as per the literature. Algorithms in the SLS group are simulated annealing (SA), iterated local search (ILS), greedy randomized adaptive search procedure (GRASP), and tabu search (TS). SI is a set of approaches established on the social activities of insects in resolving complex problems by interrelating with one another and their surroundings. These methods comprise ant colony optimization (ACO), particle swarm optimization (PSO), and bee colony optimization (BCO). EAs are a set of population-based algorithms which consists of a genetic algorithm (GA) and a memetic algorithm (MA) (Defersha & Rooyani, 2020).

When the objective consists of minimizing tardiness, which also helps minimize a tardiness penalty cost and thus total cost in the industry, the job sequence is consequently found to minimize tardiness-associated objectives (Zammori et al., 2014). Scheduling problems under proportional linear deterioration was considered to minimize total tardiness, applied branch and bound algorithm and showed that the algorithm could solve the problem in a reasonable time and the heuristic algorithm does it efficiently (Bank et al., 2012). PSO and SA algorithms for lesser job-sized problems were recommended, while SA is recommended for large job-sized problems to minimize total tardiness (Lee et al., 2014). Volgenant and Teerhuis (1999) suggested a technique for dealing with the multiple jobs and single-machine weighted tardiness problem. The authors investigated the relationship between due date allocation approaches and scheduling procedures in a dynamic job shop. The

system is evaluated using scheduling rules and performance criteria such as flow time and job tardiness (Vinod & Sridharan, 2011). To maximize job shop system performance, the authors used ACO. The GA constantly produces answers with lower total earliness and tardiness compared to SA techniques, a neighbourhood search, a variable greedy algorithm, and fast ant colony processes (Schaller & Valente, 2013).

However, no single technique can address all JSP in a reasonable period and with a good answer (Hasan et al., 2009). Mixed-integer programming (MIP) model for the problem was proposed and solved by a hybrid method combining variable neighbourhood search and mixed-integer linear programming. It was revealed that the hybrid method reached the best solution to minimize earliness and tardiness in most instances and was found to be better than the MIP solver (M’Hallah, 2014). A metaheuristic of category intelligent optimization techniques for minimizing makespan and tardiness was studied. The findings revealed that combining metaheuristics with variable neighbourhood search increased their performance (Anjana et al., 2020). A method for minimizing the makespan of a scheduling problem by combining concepts from dispatching rules, GA, and data mining is developed. The study findings suggested that the method effectively solves scheduling problems in real-time (Habib Zahmani & Atmani, 2021). SA with GA was used to sequence several courses related to the classroom in education (Czibula et al., 2016). Many researchers have chosen GA as their method of choice among these and all of the other strategies described earlier. GA has been widely used in JSP as a stand-alone or principal algorithm in hybrid techniques (Sergienko et al., 2009). An overview of the reviewed research articles is shown in Table 1.

**Table 1.** Summary of literature

Study by	Objective/s	Approach	Description
(Zhao et al., 2023)	Earliness/tardiness penalty	GA and Tabu search	Combined a genetic algorithm and tabu search to optimize the goal and an average improvement of 45.3% was made.
(Gil-Gala et al., 2023)	Total tardiness	Priority rules	Combined priority rules to minimize total tardiness.
(Valledor et al., 2022)	Makespan, total weighted tardiness	Hybrid dynamic non-dominated sorting GA II metaheuristic (HDNSGA-II)	HDNSGA-II discussed to find the optimal solution for minimizing makespan and tardiness.
(Ahmadian et al., 2021)	Makespan	Variable neighbourhood search (VNS)	The VNS algorithm is used to identify a near-optimal solution to reduce completion time. Optimal solutions were obtained in approximately 57% of the cases.
(Bari & Karande, 2021)	Flow time and tardiness	Dispatching rules	The best dispatching rule is RMDD, whereas the worst is minimum slack time MST, considering multiple objectives in a scheduling problem.

Study by	Objective/s	Approach	Description
(He et al., 2021)	Total flow time and mean tardiness	Effective multi-objective Jaya algorithm (EMOJaya)	Object-based learning is incorporated into the EMOJaya to enhance the search quality and effectiveness of the population.
(Negi et al., 2021)	Multiple objectives	Metaheuristic grey wolf optimization (GWO)	Combined GWO with different metaheuristic approaches to achieve a solution for objective functions.
(Anjana et al., 2020)	Makespan and tardiness	Metaheuristic of category intelligent optimization	Combined metaheuristics with variable neighbourhood search increased the performance.
(Ardakani et al., 2020)	Makespan	Heuristic algorithm	The heuristic algorithm works better than the mathematical model for truck-to-door sequencing.
(Cayo & Onal, 2020)	Tardiness with setup time	Heuristic approach	Sequencing production orders in near-real-time, primarily to minimize total tardiness.
(Gong et al., 2019)	Completion time	Metaheuristic approach - Effective memetic algorithm (EMA)	To increase the algorithm's performance and fully utilize the solution space, a novel efficient local search strategy is suggested and included in the EMA.
(Martínez et al., 2019)	Setup times and costs	Exact solution - Branch and check	Find the best solutions to the production planning problem of the packaging industry.
(Dao et al., 2018)	Flow time	Versions of the bat algorithm	Versions of the bat algorithm implemented, communication strategy schemes, and the makespan scheme were used to solve the NP-hard job shop scheduling problems.
(Yu & Lee, 2018)	Total family tardiness	Exact solution - Branch and bound	Jobs are organized into families using the branch and bound technique.
(K. K. Kumar et al., 2017)	Flow time, tardiness	Dispatching rules	The SPT rule gives an appropriate sequence for flow time problems in a static scheduling problem. EDD rule generates a suitable sequence for tardiness problems.
(Pranzo & Pacciarelli, 2016)	Completion time	Iterated Greedy (IG) algorithm	Metaheuristic technique described for increasing accuracy of greedy constructive heuristic.
(Gupta & Chauhan, 2015)	Makespan	Heuristic	Heuristics have been frequently used to solve NP-hard problems.

Study by	Objective/s	Approach	Description
(Li et al., 2015)	Tardiness	Metaheuristic-GA	The most widely used meta-heuristic for solving the scheduling problem to reduce tardiness.
(Lee et al., 2014)	Total tardiness	PSO and SA algorithms	PSO and SA algorithms are for lesser job-sized problems, while SA is recommended for large job-sized problems to minimize total tardiness.
(M'Hallah, 2014)	Earliness and tardiness	Hybrid variable neighbourhood search and mixed-integer linear programming	The hybrid method reached the best solution to minimize earliness and tardiness in most instances.
(Schaller & Valente, 2013)	Total earliness and tardiness	GA	GA constantly produces answers with lower total earliness and tardiness compared to SA techniques, a neighbourhood search, a variable greedy algorithm, and fast ant colony processes studied in the article.
(Sayadi et al., 2010)	Makespan	Metaheuristic	Meta-heuristic approaches have been frequently used to solve complex problems, as accurate methods are unfeasible for medium and large cases.

### 2.1. Research gap

- The existing work used a hybrid approach focusing on makespan criteria of scheduling problems. Some of the literature used combined algorithms to reduce tardiness. However, the researchers did not use the combination of dispatching rules and GA to minimize tardiness and the randomness of the population.
- Different approaches are required to minimize tardiness to increase production efficiency while decreasing expenses.

## 3. Methodology

### 3.1. Dispatching Rules for Scheduling and its Formulation

Dispatching rules determine the sequence in which jobs are worked on a machine. Using different rules in scheduling results in a distinct scheduling sequence and significantly impacts the performance measures. As a result, more testing is required before making a definitive conclusion on the best sequencing rule to use in scheduling. Thus a comprehensive strategy for selecting the optimal sequence is necessary for managerial decision-making. However, these principles may not always provide the best sequence for scheduling jobs, so they must be combined with an evolutionary strategy. GA can scan a wide range of search spaces because of its substantial

exploration capacity. The lack of a powerful search engine prevents it from precisely searching closely for a good solution. To solve the job shop scheduling challenge, numerous researchers are increasing GA's performance by incorporating various search and heuristic methodologies for minimizing the makespan. In this research work, to minimize tardiness, GA is combined with dispatching rules to find a solution that is close to optimal through community evaluation and the injection of the best solution of antecedent iteration to successor iteration.

The following sequencing rules are considered alternatives for allowing a workstation's schedule to progress through time, and their formulation procedure is presented below. Table 2 lists the notations used in the paper.

**Table 2.** Notations and their description

Notations	Description
$1, 2, 3, \dots, N-1, N$	Jobs $j$
$P_j$	Processing time of job $j$
$D_j$	Due date of job $j$
$C_j$	The time by which the processing of job $j$ is completed
$T_j$	The tardiness of job $j$
$Df(j, P_j, D_j)$	Data frame for a set of jobs with a processing time of job $j$ and due date of job $j$
$S_{lki}$	Slack time of job $j$
$S_{lkmin}$	Minimum slack time
$t$	The period at which a job is selected for operation
$\partial$	Change in time $t$
$M_{ptj}$	Modified processing time of job $j$ at time $t$
$M_{ddj}$	Modified due date of job $j$
$RM_{ddj}$	Reversed modified due date of job $j$
$M_{rmd}$	The modified due date for reversed modified due date of job $j$
$1, 2, 3, \dots, n-1, n$	Factors that affect the processing time of jobs
$x$	(Excellent-5, Very Good-4, Good-3, Satisfactory-2, Bad-1) Not limited can have less or more levels
$c$	Condition of job $j$ for a specific no. of factors with levels used
$P_{jc}$	Processing time of job $j$ at given condition $c$ with factors affecting the job at a given time with levels of factors
$T_{Nc}$	Total number of condition $c$ of job $j$ for a specific number of factors with levels used
$SC_1, SC_2, SC_3, \dots, SC_{n-1}, SC_n$	Scenarios affecting the job's processing time
$\mu_{sc_j}$	Average processing time for the selected scenarios concerning job $j$



Notations	Description
$P_{Sc_j}$	Processing time of job $j$ for $Sc_1, Sc_2, Sc_3, \dots, Sc_{n-1}, Sc_n$
$T_{N_s}$	Total number of scenarios for job $j$
$p(Sc_1), p(Sc_2), \dots, p(Sc_n)$	Probabilities associated with scenarios
$T_{t_{s1}}, T_{t_{s2}}, \dots, T_{t_{sn-1}}, T_{t_{sn}}$	Total tardiness of scenarios
$E(T_i)$	Expected total tardiness
$A_{dd}$	Average due date
$V_{dd}$	Varied due date

SPT: Jobs are processed following their processing time. The job that requires the least amount of processing time on the system is scheduled as soon as possible.

Procedure 1: Formulation of SPT rule  
 Input:  $Df(j = 1, 2, \dots, N, P_j, D_j)$   
 begin  
      $Sorted\_Df(j, P_j, D_j) = Df.Sort(P_j, Ascending = True)$   
 end  
 Output:  $SPT\_Rule\_Df(j, P_j, D_j) = Sorted\_Df(j, P_j, D_j)$

EDD: Jobs are performed in sequential order to be supplied to the user.

Procedure 2: Formulation of EDD rule  
 Input:  $Df(j = 1, 2, \dots, N, P_j, D_j)$   
 begin  
      $Sorted\_Df(j, P_j, D_j) = Df.sort(D_j, Ascending = True)$   
 end  
 Output:  $EDD\_Rule\_Df(j, P_j, D_j) = Sorted\_Df(j, P_j, D_j)$

MST: Choose the job  $j$  with the least slack time.

$$j^* = \min_{j \in N} \{D_j - t - P_j\} \quad (1)$$

where  $j^*$  is selected job with least slack time.

Procedure 3: Formulation of MST Rule  
 Input:  $Df(j = 1, 2, \dots, N, P_j, D_j)$   
 begin  
     Step 1. Set  $t = 0$   
     Step 2. Compute  $S_{lkj} = D_j - P_j$   
     Step 3.  $S_{lkmin} = \min(S_{lkj})$   
     Step 4.  $P_{jmin} =$  Processing time of job with  $S_{lkmin}$

```

Step 5.  $\partial = t + P_{jmin}$ 
Step 6. Drop the job from  $Df(j, P_j, D_j)$  with  $S_{lkmin}$  and append it in  $New\_Df(j, P_j, D_j)$ 
Step 7. Take updated  $Df(j, P_j, D_j)$ 
Step 8. Repeat Step 2 - Step 6
Step 9.  $New\_Df(j, P_j, D_j)$ 
end

Output:  $MST\_Rule\_Df(j, P_j, D_j) = New\_Df(j, P_j, D_j)$ 

```

MDD: Choose the job with the least modified due date.

$$j^* = \min_{j \in N} \{M_{ddj}\} \tag{2}$$

where,  $M_{ddj} = \max(D_j, t + P_j)$  and  $j^*$  is selected job with the least modified due date.

```

Procedure 4: Formulation of MDD Rule
Input:  $Df(j = 1, 2, \dots, N, P_j, D_j)$ 
begin
Step 1. Set  $t = 0$ 
Step 2. Calculate  $M_{ptj} = P_j + t$ 
Step 3.  $M_{ddj} = \max(D_j, M_{ptj})$ 
Step 4.  $M_{ddmin} = \min(M_{ddj})$ 
Step 5.  $P_{jmin}$  = Processing time of job with  $M_{ddmin}$ 
Step 6.  $\partial = t + P_{jmin}$ 
Step 7. Drop the job from  $Df(j, P_j, D_j)$  with  $M_{ddmin}$  and append it in  $New\_Df(j, P_j, D_j)$ 
Step 8. Take updated  $Df(j, P_j, D_j)$ 
Step 9. Repeat Step 2 - Step 8
Step 10.  $New\_Df(j, P_j, D_j)$ 
end

Output :  $MDD\_rule\_Df(j, P_j, D_j) = New\_Df(j, P_j, D_j)$ 

```

Greedy: Choose the job with the least amount of tardiness when it is finished last.

```

Procedure 5: Formulation of Greedy Rule
Input:  $Df(j = 1, 2, \dots, N, P_j, D_j)$ 
begin
Step 1. Calculate  $\text{sum}(P_j)$ 
Step 2.  $M_{xdd} = \max(M_{ddj})$ 
Step 3.  $\partial = \text{sum}(P_j) - M_{xdd}$ 
Step 4. Drop the job from  $Df(j, P_j, D_j)$  with  $M_{xdd}$  and append it in  $New\_Df(j, P_j, D_j)$ 
Step 5. Take updated  $Df(j, P_j, D_j)$ 
Step 6. Repeat Step 2 - Step 5
Step 7.  $New\_Df(j, P_j, D_j)$ 
end

Output:  $Greedy\_Rule\_Df(j, P_j, D_j) = New\_Df(j, P_j, D_j)$ 

```

Optimal job scheduling to minimize total tardiness by dispatching rules and community ...

RMDD: Select the job with the greatest reversed adjusted due date value.

$$j^* = \min_{j \in N} \{P_j + \min(M_{ddj}, 0)\} \quad (3)$$

where  $M_{ddj} = D_j - P_j - t$  and  $j^*$  is a selected job with reversed adjusted due date.

Procedure 6: Formulation of RMDD Rule  
 Input:  $Df(j = 1, 2, \dots, N, P_j, D_j)$   
 begin  
   Step 1. Set  $t = 0$   
   Step 2. Compute  $M_{ddj} = D_j - P_j - t$   
   Step 3. Calculate  $RM_{ddj} = (P_j + \min(M_{ddj}, 0))$   
   Step 4.  $M_{rmdj} = \min(RM_{ddj})$   
   Step 5.  $P_{jmin}$  = Processing time of job with  $M_{rmdj}$   
   Step 6.  $\partial = t + P_{jmin}$   
   Step 7. Drop the job from  $Df(j, P_j, D_j)$  with  $M_{rmdj}$  and include it in  $New\_Df(j, P_j, D_j)$   
   Step 8. Take Updated  $Df(j, P_j, D_j)$   
   Step 9. Repeat Step 2 - Step 8  
   Step 10.  $New\_Df(j, P_j, D_j)$   
 end  
 Output:  $RMDD\_rule\_Df(j, P_j, D_j) = New\_Df(j, P_j, D_j)$

### 3.2. Objective Function

Sequencing rules are applied to a given dataset of jobs to obtain a sequence of the jobs. Each sequencing rule provides a result as per the requirement of the operator. The operators are primarily concerned with performance measures such as flow time and job tardiness of the sequence. Performance measures are used to quantify the effectiveness of a sequencing rule. One of the most common scheduling criteria found in practical problems is the performance measure of meeting job due dates. While fulfilling deadlines is just a qualitative goal, it generally entails imposing time-based fines on late jobs while providing no advantages for finishing jobs early. This understanding naturally leads to the tardiness measure quantifying the scheduling goal, and the minimization of total tardiness is a fundamental sequencing problem. EDD sequence gives no more than one tardy job, it gives the minimum tardiness value. SPT rule focuses on processing time, and tardiness is related to meeting job due dates; SPT sequencing minimizes tardiness when all jobs have the same due date. The statement that tardiness is not a linear function of completion time makes dealing with the total tardiness measure difficult. As a result, finding optimal solutions frequently necessitates combinatorial optimization. If the number of late jobs is reduced, some jobs may have to wait an unacceptably long time. Instead, if the total amount of tardiness is reduced, the likelihood of an excessively long wait for any specific job is reduced. Thus this paper focuses on minimizing total tardiness performance measure with the combination of dispatching rules and procedure of community evaluation. The following is a basic explanation of tardiness, total tardiness, maximum tardiness and average tardiness.

Tardiness: It is the measure of the delay in completing a job beyond the due date. Tardiness can have either a positive or zero value. If the completion time minus the

due date is a negative value, the job is early and not tardy; hence, it will be associated with 0.

$$T_j = \max(0, C_j - D_j) \quad (4)$$

where,  $C_j - D_j$  is the lateness of job  $j$ .

Total Tardiness: By aggregating the tardiness of all jobs in the set, the cumulative delay of all jobs in the set is computed and shown in Eq. (5).

$$T = \sum_{j=1}^N T_j \quad (5)$$

Maximum Tardiness: It is the measure of a job with the most delay beyond the due date.

Average Tardiness: It is the proportion of total tardiness and the number of jobs in the system presented in Eq. (6).

$$T_{avg} = \frac{T}{N} \quad (6)$$

### 3.3. Proposed Algorithm

The proposed algorithm of the combinatorial method with Dispatching Rules and Community Evaluation Chromosomes (DRCEC) employs a traditional GA by injecting dispatching rules. Then, the best sequence is determined using a community evaluation of sequences. Below is a description of the proposed algorithm in detail.

DRCEC, a combinatorial algorithm, is used to develop an optimal sequence for minimizing tardiness in scheduling and sequencing problems. The community evaluation chromosomes procedure depends on the concept of GA. GA is based on the famous quote by Charles Darwin – "Survival of the Fittest". It is not the strongest or the most intelligent species that survive, but the one that is most responsive to change. The proposed algorithm is compared with PSO, which is a metaheuristic technique for selecting the best scheduling sequence with the least amount of tardiness. The PSO consists of creating the population, assigning velocity to the particles, updating the particle's location until the stopping condition is reached, and selecting the optimal alternative with the least amount of tardiness. The detailed process used in DRCEC is shown in Figure 1. The classical GA is based on selecting a random population that generates feasible chromosomes as an initial population. This starting population will affect the solutions and the time taken to obtain the optimal solution. The GA described by Bari et al. (2022) in their research work is also employed to compare the proposed algorithm. This classical GA is improved by introducing dispatching rules related to processing time and due date in the initial population of the DRCEC algorithm. When the DRCEC algorithm is implemented, chromosomes are generated with SPT, EDD, MST, MDD, RMDD and Greedy dispatching rules mentioned in section 3.1 and injected into the initial community population. The remaining chromosomes are generated to prevent the predilection that may be because of selected dispatching rules chromosomes. Once the opening community is generated using dispatching rules and randomness, this community undergoes the operation of crossover and mutation in series and generates more chromosomes. Finally, the chromosomes generated in the above steps and the opening community are merged to get the total chromosomes.

Optimal job scheduling to minimize total tardiness by dispatching rules and community ...

The community is evaluated based on tardiness by applying Eq. (4) and Eq. (5). The chromosome with minimum total tardiness is selected as the optimal solution. This iteration's optimal solution is now inserted as one of the members of the population of the successor iteration. If stopping criteria are met, the selected chromosome is declared a near-optimal sequence or the above process is applied to the next iteration. The formulation of the DRCEC algorithm is presented in procedure 7.

Procedure 7: DRCEC algorithm

Input:  $Df(j = 1, 2, \dots, N, P_j, D_j)$ ,  $num\_iteration$

begin

Step 1. Read  $Df(j, P_j, D_j)$

Step 2. Read  $num\_iteration$

Step 3.  $Rules\_sequence = Sequences$  (use Procedure 1 - Procedure 6 for generating sequences)

Step 4. Set  $num\_iteration = 1$

Step 5. Select  $population\_list = Rules\_sequences + Random\_sequences$

Step 6.  $parent\_list = population\_list +$   
 $Optimal\_sequence$

Step 7.  $offspring\_list =$  Apply Crossover operator on  $parent\_list$

Step 8.  $offspring\_list = offspring\_list +$  Apply Mutation operator on  $offspring\_list$

Step 9.  $total\_chromosomes = parent\_list +$   
 $offspring\_list$

Step 10. Calculate  $T$  for all  $total\_chromosomes$

Step 11. Compare  $T$  of all chromosomes

Step 12.  $Optimal\_Sequence =$  Select the  $total\_chromosome$  with minimum  $T$

Step 13. Inject  $Optimal\_sequence$  as member of population for next iteration

Step 14. Increment  $num\_iteration$  by 1

Step 15. If  $num\_iteration = C$ , then STOP  
else go to Step 5

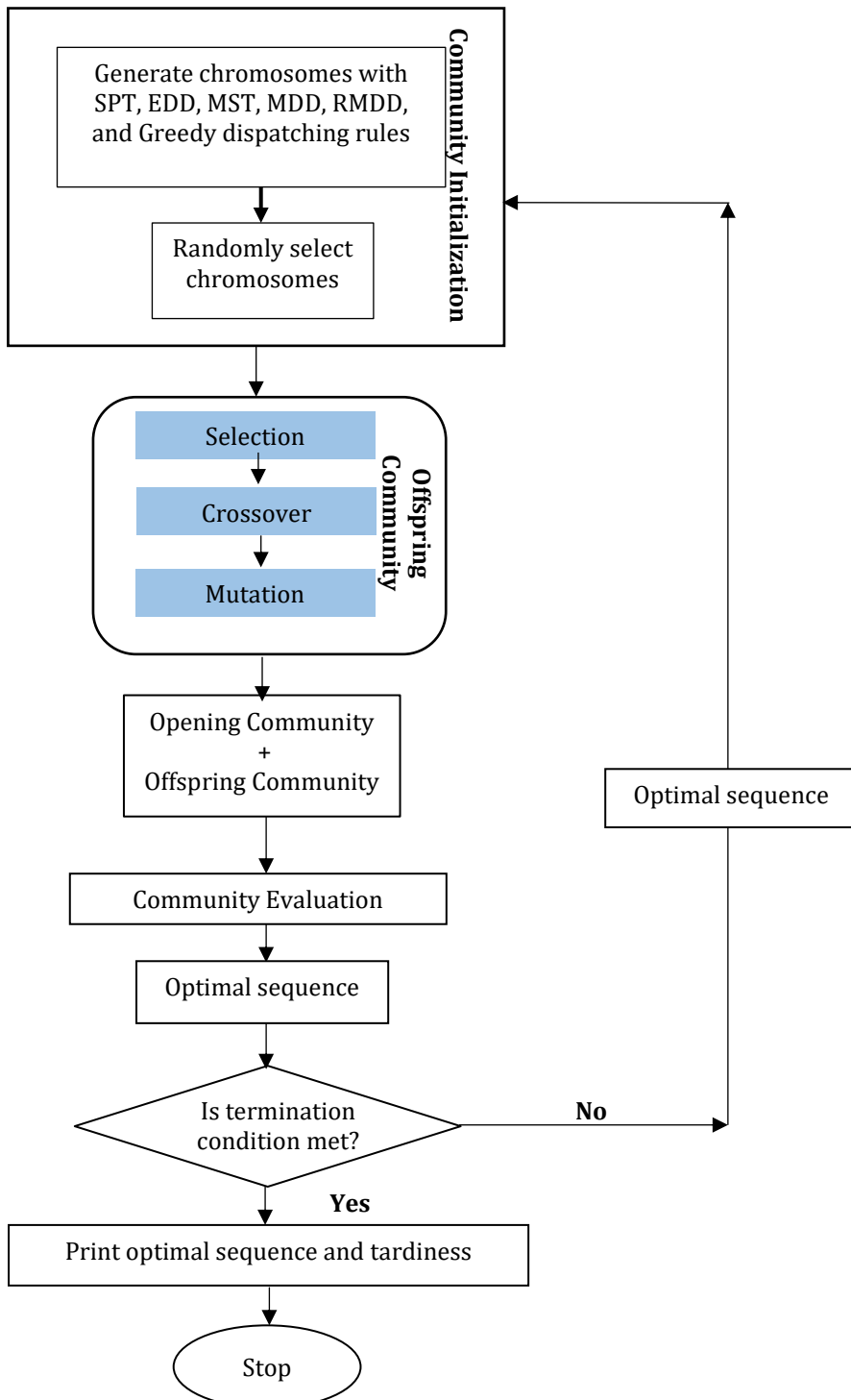
Step 16. Print  $Optimal\_Sequence$  and  $T$

end

Output:  $Optimal\_Sequence$  and  $T$

### 3.4. Stochastic Approaches for Scheduling

In real-world scheduling operations, various factors affect the processing time of jobs and cannot be neglected. The deviation produced in the accuracy of the processing time values can hamper the efficacy of the job shop and create a delay in scenarios. Stochastic scheduling provides relatively accurate results to assist the industry in scheduling jobs with real-time data. This also gives the operators an idea of when and how to sequence the jobs and provides the client with an idea to set the due date of procurement. In this research, the assumptions that job descriptors are deterministic and the machine is continuously available are relaxed. Three stochastic scheduling models, as explained below, are reformulated as deterministic and optimal sequence generated by applying the DRCEC algorithm.



**Figure 1.** Flowchart for DRCEC algorithm

Optimal job scheduling to minimize total tardiness by dispatching rules and community ...

### 3.4.1. Stochastic linguistic approach

In the stochastic linguistic approach, the processing time of jobs is estimated by considering various factors like machine conditions, raw material used, weather conditions, workers' availability, and so on that affect their operations. These factors can either increase or decrease the processing time. If a particular factor increases the job's processing, it may not be in its best form. Conversely, if the same factors are in their best form, the processing time of the job will decrease. In the linguistic method, one or all the factors can be in their best or worst shape under different circumstances. Thus, if there are ' $n$ ' factors affecting the processing time of a job and each factor can have ' $x$ ' levels, then the number of conditions is given by  $x^n$ .

Procedure 8 is used to generate stochastic linguistic scheduling data with processing time for all conditions, affecting factors with a predetermined level and jobs' due dates. Then, procedure 9 is applied to the stochastic job data in the form of linguistic level to reformulate the stochastic problem as a deterministic problem. Finally, the DRCEC algorithm is further applied to find the optimum sequence.

Procedure 8: Stochastic linguistic scheduling data representation  
while  $i$  in range  $T_{N_c}$   
  for  $j$  in range  $N$   
     $P_j = P_{j_c}$   
  end  
end  
while  $j$  in range  $N$   
   $D_j =$  Due date of job is common irrespective of conditions of job  
end  
 $Df(j, P_{j_c}, D_j)$

Procedure 9: Reformulation of a stochastic linguistic problem as a deterministic problem  
for  $j$  in range  $N$   
   $P_j =$  Processing time of selected condition  $C$  of job  $j$   
   $D_j =$  Due date of job is common irrespective of conditions of job  
end  
 $Df(j, P_j, D_j)$

### 3.4.2. Stochastic scenarios approach

In the stochastic scenarios approach, scheduling is done considering more than one scenario of the jobs. These scenarios are for the same jobs, which means the end product is the same, but the path to prepare the end product may differ. For example, in some scenarios, jobs can be made of different materials, and the size of the raw material or the machine used to process can be different. These various factors affecting the jobs can increase or decrease the processing time of the jobs. Thus, different scenarios are created that can assist in stochastic scheduling with real-time data.

Procedure 10 is used to formulate the stochastic scenarios scheduling data with processing time for all scenarios of the jobs and the due date for jobs. Procedure 11 is used to reformulate a stochastic problem into a deterministic problem using job data

from stochastic scenarios. The DRCEC algorithm is further applied to find the optimum sequence.

Procedure 10: Stochastic scenarios scheduling data representation

```

while  $i$  in range  $T_{N_s}$  //where  $i$  is the iterator variable
  for  $j$  in range  $N$ 
     $P_j = P_{Sc_j}$ 
  end
end
while  $j$  in range  $N$ 
   $D_j =$  Due date of the job is common irrespective of job conditions.
end

 $Df(j, P_{Sc_j}, D_j)$ 
    
```

Procedure 11: Reformulation of stochastic scenarios problem as a deterministic problem

```

for  $j$  in range  $N$ 
   $P_j = \mu_{sc_j}$ 
   $D_j =$  Due date of the job is common irrespective of job conditions.
end

 $Df(j, P_j, D_j)$ 
    
```

3.4.3. Stochastic probabilistic approach

The stochastic probabilistic scheduling approach is similar to the stochastic scenarios approach, wherein each scenario has a different probability. Procedure 10 is used to represent data of stochastic probabilistic scheduling, with the addition of probability of scenarios as  $p(Sc_1), p(Sc_2), p(Sc_3), \dots, p(Sc_{n-1}), p(Sc_n)$ . The summation of probabilities of all scenarios should be one, represented in Eq. (7).

$$\sum_{i=1}^n p(S_{c_i}) = 1 \tag{7}$$

where,  $p(S_{c_i})$  is the probability associated with the scenario.

DRCEC algorithm is applied to find the optimum sequence for stochastic probabilistic scheduling problems. Procedure 12 is used to find the total tardiness of the optimal sequence.



Procedure 12: Calculation of total tardiness in a stochastic probabilistic problem

$E(T_t) = 0$

while  $i$  in range  $T_{Ns}$

$$E(T_t) = E(T_t) + [p(S_{c_i}) * T_{t_{si}}]$$

end

## 4. Hypothetical Testing

This section generates and analyzes a dataset to evaluate the performance of the GA and DRCEC approaches. The computations were conducted on a computer with Intel(R) Core(TM) i3-9100F CPU @ 3.60GHz processor running on Windows 10 operating system with 8GB RAM and 500GB storage space. The computation code is written in Python programming language and comparisons are made between the results obtained through these algorithms. Thirty-five test datasets with a varied number of jobs are arbitrarily created to assess the performance of the proposed DRCEC technique.

### 4.1. Dataset Generation

A typical strategy presented by Zammori et al. (2014) was used to produce a dataset for scheduling problems. Job processing times are normally distributed, with a mean of 100 time units and a standard deviation of 25 time units picked at random; due dates are uniformly distributed with the average due date ( $A_{dd}$ ) and vary with the due date ( $V_{dd}$ ) given by Eq. (8) and Eq. (9) respectively.

$$A_{dd} = \left( \frac{1}{N} \sum_{j=1}^N P_j \right) N * (1 - T_f) \quad (8)$$

$$V_{dd} = \left( \frac{1}{N} \sum_{j=1}^N P_j \right) N * V_d \quad (9)$$

where,  $T_f$  is a tardiness factor in the range [0.2, 0.8] that roughly correlates to the predicted proportion of tardy jobs in a random sequence of jobs, and  $V_d$  [0.2, 0.6] is the relative variation in due dates. Parameters  $T_f$  and  $V_d$  considered in this paper for generating due dates are 0.3 and 0.4, respectively.

### 4.2. Application of PSO, GA and DRCEC on Datasets

Five datasets are generated for each with 5, 10, 15, 25, 35, 45, and 100 jobs. D005\_1 to D005\_5 are the five datasets with five jobs (D represents the dataset, 005 represents the number of jobs present in the dataset, and 1 to 5 indicates five datasets consecutively). All datasets are labelled with numbers in this manner for simple identification. The total tardiness is calculated for each dataset using different priority

rules like SPT, EDD, MST, MDD, Greedy, and RMDD methods which are recorded in Table 3. When compared to all other rules, the highlighted value has the lowest tardiness, indicating that it is the best for the dataset. It is observed that the MDD rule contributes more to obtaining optimal sequence concerning the tardiness of jobs. Each dataset was tested with combinations of parameters such as population size 200, crossover rate 0.8, and mutation rate 0.1 with a varied number of iterations. Table 4 briefly describes the parameters used for testing 35 datasets. With each combination mentioned in Table 4, the dataset is run 10 times for PSO, GA and the proposed DRCEC approach. In this way, a total of 5250 test results of total tardiness are found for 35 datasets. The results of average and standard deviations of 10 runs' total tardiness with varying iterations for datasets are recorded and shown in Table 5.

**Table 3.** Tardiness of the dataset using priority rules

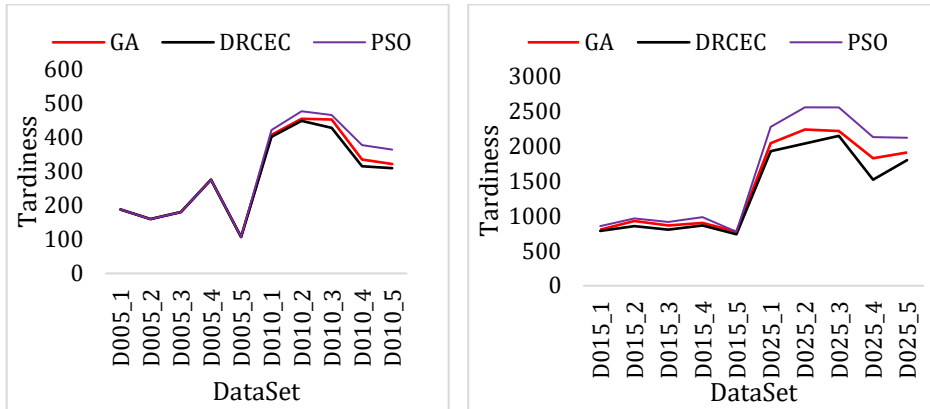
Dataset	SPT	EDD	MST	MDD	Greedy	RMDD
D005_1	216	227	268	<b>188</b>	227	229
D005_2	<b>184</b>	196	196	196	196	<b>184</b>
D005_3	210	<b>192</b>	230	<b>192</b>	<b>192</b>	224
D005_4	<b>275</b>	286	434	<b>275</b>	286	286
D005_5	<b>107</b>	127	157	<b>107</b>	127	127
D010_1	403	<b>401</b>	506	<b>401</b>	<b>401</b>	403
D010_2	496	516	652	<b>448</b>	516	496
D010_3	578	536	536	<b>491</b>	536	597
D010_4	437	<b>315</b>	476	<b>315</b>	<b>315</b>	440
D010_5	404	386	620	<b>309</b>	386	404
D015_1	947	900	973	<b>838</b>	900	909
D015_2	964	920	1087	<b>854</b>	920	1007
D015_3	976	896	1119	<b>824</b>	896	974
D015_4	1030	1041	1267	<b>945</b>	1041	1051
D015_5	<b>906</b>	1004	1067	961	1004	1006
D025_1	2147	2789	3328	<b>2112</b>	2789	2356
D025_2	2520	2321	2452	<b>2257</b>	2321	2522
D025_3	2357	2593	3219	<b>2260</b>	2593	2360
D025_4	2031	1747	1926	<b>1516</b>	1747	2033
D025_5	1895	2145	2291	<b>1801</b>	2145	1914
D035_1	4349	1011	1093	<b>891</b>	1011	4048
D035_2	4549	1761	1865	<b>1540</b>	1761	4358
D035_3	7306	<b>1004</b>	1119	<b>1004</b>	<b>1004</b>	5745
D035_4	5324	2010	2010	<b>1767</b>	2010	5517
D035_5	6717	<b>900</b>	1008	<b>900</b>	<b>900</b>	6452
D045_1	9568	2708	3031	<b>2334</b>	2708	9667
D045_2	7952	508	508	<b>456</b>	<b>456</b>	5270
D045_3	6537	1662	1681	<b>1571</b>	1662	6180
D045_4	8481	605	644	<b>536</b>	605	6935
D045_5	6037	2401	2460	<b>2224</b>	2401	6084
D100_1	182215	210869	214489	<b>174612</b>	210839	194431
D100_2	183832	208941	211376	<b>174661</b>	208953	194554

Dataset	SPT	EDD	MST	MDD	Greedy	RMDD
D100_3	<b>202532</b>	247432	304699	251976	250476	208143
D100_4	212641	234417	236794	<b>200488</b>	234417	223177
D100_5	193488	220262	222790	<b>185405</b>	220300	203872

**Table 4.** Parameters for the dataset testing

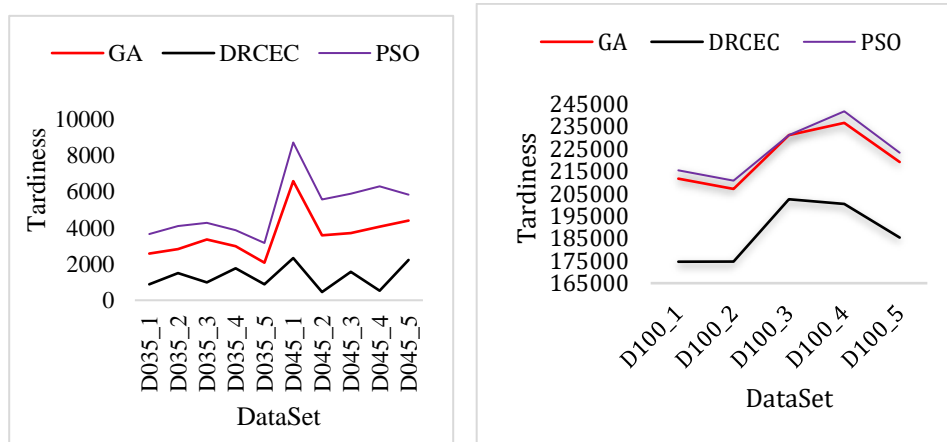
Dataset	Group of Dataset	No. of Iterations	No. of Jobs
D005_1 to D005_5	Small	10	5
		20	
		30	
		40	
		50	
D010_1 to D010_5	Small	10	10
		20	
		30	
		40	
		50	
D015_1 to D015_5	Medium	10	15
		20	
		30	
		40	
		50	
D025_1 to D025_5	Medium	10	25
		20	
		30	
		40	
		50	
D035_1 to D035_5	Large	10	35
		20	
		30	
		40	
		50	
D045_1 to D045_5	Large	10	45
		20	
		30	
		40	
		50	
D100_1 to D100_5	With 100 jobs	10	100
		20	
		30	
		40	
		50	

Figure 2 (a - d) shows a graphical representation of trends in average total tardiness of 10 runs for 10 iterations of each dataset. From Figure 2, it is observed that the DRCEC does better than the GA and PSO. All the datasets are tested for 10, 20, 30, 40 and 50 iterations and observed that the total tardiness of the DRCEC approach for all the datasets is less than GA and PSO.



(a) Dataset with 5 and 10 jobs

(b) Dataset with 15 and 25 jobs

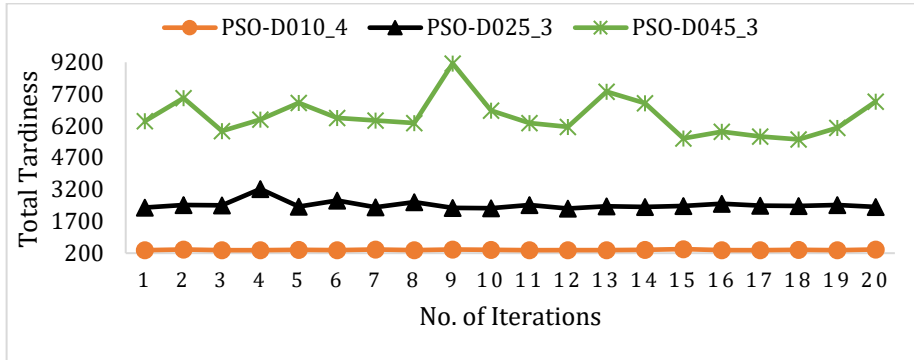


(c) Dataset with 35 and 45 jobs

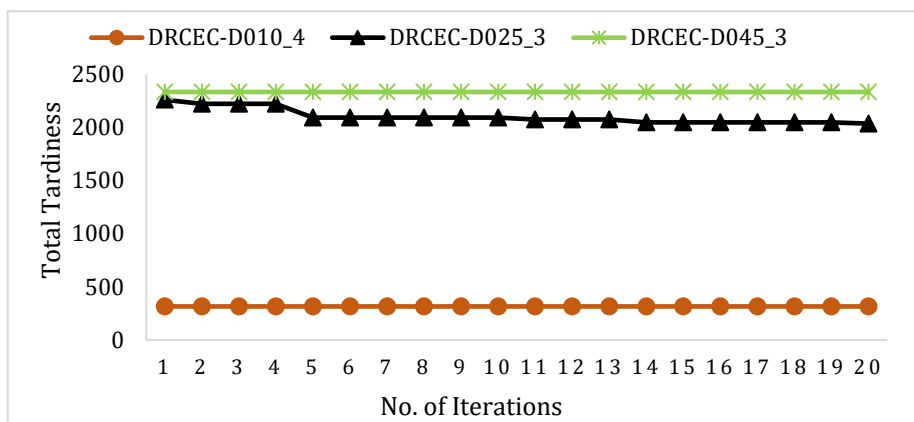
(d) Dataset with 100 jobs

**Figure 2.** Trends of total tardiness for PSO, GA and DRCEC

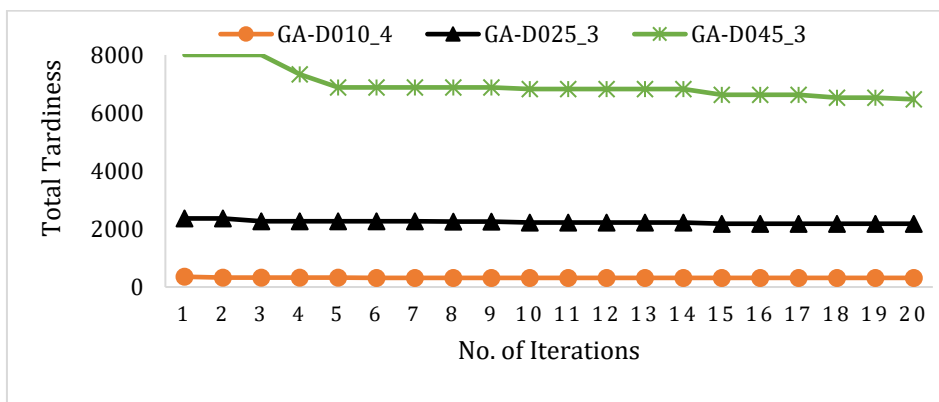
The convergence tendency of PSO, GA and DRCEC algorithms for randomly selected sample datasets is shown in Figure 3. The convergence rate of the DRCEC is more than the PSO and the GA algorithm and assures a better value of total tardiness with a smaller number of iterations. In DRCEC, convergence occurs because priority rules are introduced to the initial population. Furthermore, in the DRCEC method, the best sequence of each iteration is added to the population of the next iteration, enhancing the fitness function of all chromosomes. This promises the convergence of the DRCEC.



(a) Convergence for PSO



(b) Convergence for GA



(c) Convergence for DRCEC

**Figure 3.** Comparison of convergence in PSO, GA and DRCEC

**Table 5.** Results of average and standard deviation of 10 runs for datasets convergence process

Dataset	No. of Iterations = 10											
	Min value of Tardiness for 10 runs			Average Tardiness			Std. Deviation					
	DRCEC	GA	PSO	DRCEC	GA	PSO	DRCEC	GA	PSO	DRCEC	GA	PSO
D005_1	188	188	188	188	188	188	188	188	188	188	0	0
D005_2	160	160	160	160	160	160	160	160	160	160	0	0
D005_3	180	180	180	180	180	180	180	180	180	180	0	0
D005_4	275	275	275	275	275	275	275	275	275	275	0	0
D005_5	107	107	107	107	107	107	107	107	107	107	0	0
D010_1	401	401	401	401	406.7	421.3	401	406.7	421.3	421.3	0	5.41
D010_2	448	448	448	448	454.1	476.2	448	454.1	476.2	476.2	0	3.8
D010_3	425	444	448	427.3	452.3	465.3	427.3	452.3	465.3	465.3	4.63	7.09
D010_4	315	315	331	315	334.1	377.2	315	334.1	377.2	377.2	0	12.66
D010_5	309	309	326	309	320.9	363.2	309	320.9	363.2	363.2	0	18.77
D015_1	780	787	793	786.6	805.1	853.6	786.6	805.1	853.6	853.6	5.81	15.22
D015_2	845	886	909	852.3	928.5	961.9	852.3	928.5	961.9	961.9	2.76	27.46
D015_3	796	813	854	804.2	862.3	913.9	804.2	862.3	913.9	913.9	7.81	33.19
D015_4	842	842	875	863.5	897.7	980.2	863.5	897.7	980.2	980.2	14.64	28.8
D015_5	735	735	735	737	772.9	772.9	737	772.9	772.9	772.9	3.81	37.19

D025_1	1860	1867	2092	1923.8	2042.2	2273	31.36	86.64	148.2
D025_2	1933	2121	2199	2036.4	2236.6	2552.3	66.93	67.84	137.28
D025_3	2089	2144	2273	2145.6	2215.8	2552.8	43.82	37.93	202.27
D025_4	1516	1732	1952	1516	1827.3	2128.3	0	67.44	89.03
D025_5	1778	1868	1960	1796.9	1908.2	2120.1	7.89	23.25	103.59
D035_1	891	2217	2844	891	2589.8	3671.9	0	171.05	465.65
D035_2	1456	2512	3291	1499.3	2829.6	4102	17.98	195.39	526.62
D035_3	957	2933	3840	982.6	3358.2	4288.4	20.65	226.93	287.54
D035_4	1757	2429	3018	1760.3	2985.2	3877.8	3.35	270.54	426.45
D035_5	849	1820	2096	875.6	2078.1	3168.8	16.73	245.37	462.13
D045_1	2334	5187	7050	2334	6589.3	8727.9	0	623.74	968.17
D045_2	456	2331	3680	456	3600.8	5579.8	0	800.05	1035.78
D045_3	1571	2557	5083	1571	3718.2	5892.6	0	495.65	513.25
D045_4	536	3005	5413	536	4076.1	6301.5	0	492.96	680.2
D045_5	2206	3940	4883	2220.4	4398.5	5842.4	6.86	252.87	529
D100_1	17461	209189	211215	174612	211771.6	215405.7	0	1194.46	3370.03
D100_2	17466	200800	205529	174661	207148	210851.5	0	2265.33	2892.11
D100_3	20253	228260	247260	202532.2	231220.1	221450.1	0	1843.07	1973.07
D100_4	20048	234082	239429	200488	236685.1	241847.7	0	1355.09	1864.24
D100_5	18540	215804	216542	185405	219121.2	223289.2	0	1665.08	3510.14

Dataset	No. of Iterations = 20											
	Min value of Tardiness for 10 runs					Average Tardiness					Std. Deviation	
	DRCEC	GA	PSO	DRCEC	GA	PSO	DRCEC	GA	PSO	DRCEC	GA	PSO
D005_1	188	188	188	188	188	188	188	188	188	0	0	0
D005_2	160	160	160	160	160	160	160	160	160	0	0	0
D005_3	180	180	180	180	180	180	180	180	180	0	0	0
D005_4	275	275	275	275	275	275	275	275	275	0	0	0
D005_5	107	107	107	107	107	107	107	107	107	0	0	0
D010_1	401	401	403	401	401	428.2	401	401	428.2	0	0	22.53
D010_2	448	448	450	448	448	465.1	448	448	465.1	0	0	11.75
D010_3	425	425	425	425	425	476.9	425	449.5	476.9	0	12.98	35.5
D010_4	315	315	325	315	322.4	361.9	315	322.4	361.9	0	8.3	18.11
D010_5	309	309	309	309	309	340.4	309	309	340.4	0	0	21.64
D015_1	780	780	802	780.4	789.3	873.4	780.4	789.3	873.4	0.76	11.55	75.67
D015_2	845	845	936	845.5	869.1	976.3	845.5	869.1	976.3	1.43	20.32	22.63
D015_3	796	798	863	796	835.2	915.2	796	835.2	915.2	0	24.28	30.19
D015_4	842	854	878	846.7	881.7	926.2	846.7	881.7	926.2	5.65	16.57	43.16
D015_5	735	735	804	735	745.1	862.2	735	745.1	862.2	0	16.02	47.69



Optimal job scheduling to minimize total tardiness by dispatching rules and community ...

D025_1	1799	1846	2046	1826.9	1960.8	2237.2	16.33	51.85	101
D025_2	1896	2075	2349	1935.2	2133.8	2700.3	31.72	39.52	245.03
D025_3	1989	2121	2247	2025.9	2165.4	2465.8	33.4	27.86	159.03
D025_4	1516	1600	1833	1516	1685.3	2053.7	0	48.92	131.13
D025_5	1761	1832	1976	1776.9	1883	2094.3	12.58	25.75	74.95
D035_1	891	1435	3001	891	1678.3	3605.9	0	194.95	387.22
D035_2	1460	2020	3919	1485.3	2431.1	4315.8	13.1	209.95	334.45
D035_3	940	2084	4419	973.4	2476.8	4875.4	23.79	159.66	353.32
D035_4	1755	2161	3324	1755.6	2497.5	3941.1	0.87	170.33	394.37
D035_5	783	1135	2663	839.3	1526.4	3376.7	24.5	224.61	305.37
D045_1	2334	4667	7692	2334	5448.7	8471.4	0	380.02	559.74
D045_2	456	1477	5266	456	2263.3	6495.4	0	427.33	1818.6
D045_3	1571	2662	4437	1571	3042.4	6022	0	265.05	659.86
D045_4	536	1818	5125	536	2686.3	6577.8	0	490.47	921.7
D045_5	2224	3270	5030	2224	3518.7	6044.2	0	206.28	508.13
D100_1	174612	206060	211724	174612	208489	214770.4	0	1301.58	1990.99
D100_2	174661	202435	207843	174661	206063.5	211388.3	0	1759.24	2117.1
D100_3	202532	227081	238081	202532.2	229408.3	230508.4	0	2260.41	2345.31
D100_4	200488	233064	233093	200488	236061.5	233093	0	1843.3	0
D100_5	185405	215181	219866	185405	217198.2	224223.2	0	1358.37	3578.78

Dataset	No. of Iterations = 30														
	Min value of Tardiness for 10 runs						Average Tardiness						Std. Deviation		
	DRCEC	GA	PSO	DRCEC	GA	PSO	DRCEC	GA	PSO	DRCEC	GA	PSO	DRCEC	GA	PSO
D005_1	188	188	188	188	188	188	188	188	188	188	188	0	0	0	0
D005_2	160	160	160	160	160	160	160	160	160	160	160	0	0	0	0
D005_3	180	180	180	180	180	180	180	180	180	180	180	0	0	0	0
D005_4	275	275	275	275	275	275	275	275	275	275	275	0	0	0	0
D005_5	107	107	107	107	107	107	107	107	107	107	107	0	0	0	0
D010_1	401	401	403	401	401.2	424.6	401	401.2	401.2	424.6	424.6	0	0.57	24.59	24.59
D010_2	448	448	448	448	452.8	456.9	448	452.8	452.8	456.9	456.9	0	2.58	7.05	7.05
D010_3	425	425	425	425	440.2	460.9	425	440.2	440.2	460.9	460.9	0	15.06	18.16	18.16
D010_4	315	315	315	315	320.5	369.9	315	320.5	320.5	369.9	369.9	0	9.8	32.65	32.65
D010_5	309	309	309	309	309	325.2	309	309	309	325.2	325.2	0	0	14.52	14.52
D015_1	780	780	793	780	792.3	849.1	780	792.3	792.3	849.1	849.1	0	10.81	44.6	44.6
D015_2	845	845	862	845	870.4	959.3	845	870.4	870.4	959.3	959.3	0	16.6	54.56	54.56
D015_3	796	796	864	796	833.7	940.5	796	833.7	833.7	940.5	940.5	0	24.49	48.5	48.5
D015_4	842	844	878	842.1	866.5	929.8	842.1	866.5	866.5	929.8	929.8	0.28	16.97	34.98	34.98
D015_5	735	735	831	735	744.4	875.1	735	744.4	744.4	875.1	875.1	0	7.4	58.21	58.21

Optimal job scheduling to minimize total tardiness by dispatching rules and community ...

D025_1	1766	1833	2095	1784.5	1893.4	2307	18.7	27.2	78.3
D025_2	1867	1934	2198	1902.3	2014.2	2438.4	20.39	64.94	138.63
D025_3	1975	2066	2139	2000.9	2115.3	2381.5	12.75	30.79	109.77
D025_4	1516	1563	1973	1516	1655.3	2208.6	0	46.63	198.62
D025_5	1761	1795	1959	1765.6	1838.8	2111.2	5.8	29.43	160.27
D035_1	884	1112	2635	889.6	1266	3586.7	2.66	101.8	417.56
D035_2	1456	1568	4152	1480.6	1939.8	4421.1	14.79	235.85	180.95
D035_3	930	1502	3787	962	1841	4657	24.09	229.4	637.92
D035_4	1755	1935	3449	1755.2	2142.7	4217.9	0.57	163.87	646.84
D035_5	783	915	2838	826.4	1174.8	3503.9	39.75	176.54	295.31
D045_1	2334	4214	7651	2334	4785.9	8537.3	0	486.88	641.1
D045_2	456	689	5456	456	1115.1	6535.8	0	237.01	685.37
D045_3	1571	2031	4392	1571	2435.4	6146.8	0	226.42	1002.85
D045_4	536	1137	6077	536	1594.9	6724	0	355.99	915.33
D045_5	2204	2777	5524	2222	3167.7	6700.5	5.72	277.22	867.37
D100_1	174186	206568	209213	174505.1	209510.5	216315.2	147.06	1476.32	3554.74
D100_2	171891	200936	208379	174371.8	204142.7	212143.1	789.21	1439.24	2822.28
D100_3	202532.24	226394	234763.74	202532.24	230177	242188.71	0	2008.682	2109.322
D100_4	200280	228858	233863	200467.2	234311	240164	59.49	2237.991	3717.1873
D100_5	183640	212980	218396	185050.8	217064.3	222344.1	496.55	1754.689	2259.9433

Dataset	No. of Iterations = 40													
	Min value of Tardiness for 10 runs					Average Tardiness					Std. Deviation			
	DRCEC	GA	PSO	DRCEC	GA	DRCEC	GA	PSO	DRCEC	GA	PSO	DRCEC	GA	PSO
D005_1	188	188	188	188	188	188	188	188	0	0	0	0	0	0
D005_2	160	160	160	160	160	160	160	160	0	0	0	0	0	0
D005_3	180	180	180	180	180	180	180	180	0	0	0	0	0	0
D005_4	275	275	275	275	275	275	275	275	0	0	0	0	0	0
D005_5	107	107	107	107	107	107	107	107	0	0	0	0	0	0
D010_1	401	401	401	401	401.2	401	401.2	427.2	0	0.57	25.86	0	0.57	25.86
D010_2	448	448	450	448	449.6	448	449.6	465.1	0	1.86	15.31	0	1.86	15.31
D010_3	425	425	440	425	433.7	425	433.7	462	0	7.78	16.72	0	7.78	16.72
D010_4	315	315	344	315	315.9	315	315.9	385.2	0	1.31	73.33	0	1.31	73.33
D010_5	309	309	309	309	309	309	309	386.4	0	0	57.6	0	0	57.6
D015_1	780	780	780	780	784.1	780	784.1	851.3	0	2.84	31.51	0	2.84	31.51
D015_2	845	845	900	845	862.5	845	862.5	994.5	0	17.75	72.99	0	17.75	72.99
D015_3	796	796	798	796	812.2	796	812.2	911.3	0	14.62	58.18	0	14.62	58.18
D015_4	842	842	880	842	855.4	842	855.4	950.7	0	11.74	55.11	0	11.74	55.11
D015_5	735	735	769	735	735	735	735	887.3	0	0	67.07	0	0	67.07

Optimal job scheduling to minimize total tardiness by dispatching rules and community ...

D025_1	1766	1816	2156	1769	1872.8	2279.5	6	37.39	68.59
D025_2	1862	1928	2408	1880.4	1967.8	2579.1	14.16	31.65	140.36
D025_3	1975	2046	2310	1984.6	2097.9	2421.1	8.85	27.7	86.19
D025_4	1516	1564	2014	1516	1609.7	2164.8	0	25.99	157.52
D025_5	1761	1787	1962	1762.6	1827.5	2057.4	2.92	20.45	68.03
D035_1	891	989	2884	891	1137	3551.3	0	86.54	401.71
D035_2	1456	1544	3319	1480	1683	4493.7	11.72	73.72	696.01
D035_3	930	1365	3778	964	1489.3	4657.1	29.26	89	360.12
D035_4	1755	1902	3222	1755	2012	4043.5	0	72.25	367.69
D035_5	783	828	2335	835.2	939	3312.6	33	91.33	411.77
D045_1	2315	2961	7716	2332.1	3876.1	8552.6	5.43	361.3	666.89
D045_2	456	456	5125	456	622.2	6063.8	0	113.92	824.08
D045_3	1571	1773	5031	1571	2026.4	5701.5	0	197.24	560.97
D045_4	536	891	4884	536	1219.2	6305.2	0	238.29	685.38
D045_5	2224	2603	5714	2224	2819.8	6343.4	0	210.75	488.81
D100_1	172944	205595	211895	174088.4	208649	216786.9	500.43	1742.3	3551.96
D100_2	172671	199948	208571	173968.2	202954.3	212027.4	636.15	1856.86	2203.28
D100_3	202532	227325	233715	202532.2	229733.3	235125.1	0	1183.45	1325.79
D100_4	199740	231803	234707	200336.3	233474.6	243167.9	241.22	1279.52	6413.55
D100_5	182999	211992	217814	182959.1	215738.6	224136.7	52643.75	1686.75	3401.54

Dataset	No. of Iterations = 50											
	Min value of Tardiness for 10 runs					Average Tardiness					Std. Deviation	
	DRCEC	GA	PSO	DRCEC	GA	PSO	DRCEC	GA	PSO	DRCEC	GA	PSO
D005_1	188	188	188	188	188	188	188	188	188	0	0	0
D005_2	160	160	160	160	160	160	160	160	160	0	0	0
D005_3	180	180	180	180	180	180	180	180	180	0	0	0
D005_4	275	275	275	278	275	275	275	275	275	0	0	0
D005_5	107	107	107	107	107	107	107	107	107	0	0	0
D010_1	401	401	401	401	401	401	401	401	439.3	0	0	59.09
D010_2	448	448	448	448	448	448	448	448	469.6	0	0	36.64
D010_3	425	425	425	425	425	425	425	425	445.2	0	0	15.4
D010_4	315	315	318	315	315	315	315	315	358.2	0	0	28.73
D010_5	309	309	309	309	309	309	309	309	333.5	0	0	19.29
D015_1	780	780	789	780	780.8	870.3	780	780.8	870.3	0	2.28	78.4
D015_2	845	845	845	845	847.8	960.1	845	847.8	960.1	0	2.78	60.1
D015_3	796	796	860	796	796.5	921.2	796	796.5	921.2	0	0.97	55.2
D015_4	842	842	894	842	846.4	952.2	842	846.4	952.2	0	5.91	40.34
D015_5	735	735	774	735	735	852.3	735	735	852.3	0	0	52.61

Optimal job scheduling to minimize total tardiness by dispatching rules and community ...

D025_1	1766	1813	2083	1776.1	1867.3	2292.5	14.94	29.23	88.76
D025_2	1862	1813	2189	1865.1	1867.3	2593	5.75	29.23	225.79
D025_3	1975	2034	2288	1979.5	2083.2	2429.5	7.98	30.16	90.38
D025_4	1516	1539	1927	1516	1578.9	2117	0	23.55	111.79
D025_5	1760	1768	1945	1762.5	1824	2070.4	3.22	26.79	86.07
D035_1	884	937	2765	888.9	1019.1	3480.9	3.05	50.8	446.59
D035_2	1456	1540	3345	1469	1630.6	4371.8	13.43	54.48	675.61
D035_3	930	1157	3912	951.9	1309.1	4460.3	22.61	135.38	326.38
D035_4	1755	1838	2832	1755	1986.9	4009.4	0	86.46	492.96
D035_5	783	789	2718	816.7	891.7	3451	18.38	61.82	352.52
D045_1	2334	3035	7540	2334	3712.2	8519.4	0	381.7	586.88
D045_2	456	456	4820	456	541.6	6261.3	0	89.17	773.07
D045_3	1571	1634	4527	1571	1910	5663.1	0	131.76	585.2
D045_4	536	611	4034	536	805	6274.3	0	120.85	990.64
D045_5	2204	2334	5297	2222	2536.9	6156.3	5.72	122.63	631.8
D100_1	172879	204661	213216	173909	208255.1	218023.3	579.4	1808.97	2894.34
D100_2	170521	199461	208234	172287.2	203094.4	212211.3	927.04	1949.9	2074.88
D100_3	202532	225441	228564	202532.2	228642.6	235768.6	0	1555.68	1734.68
D100_4	198819	229699	235293	199927.8	233147.6	239032.4	433.53	1694.38	2287.28
D100_5	182006	212938	220216	182959.1	215603.4	222784.9	487.2	1775.38	1702.06

### 4.3. Execution Time of GA and DRCEC

The time of each iteration for the D045\_3 dataset as a sample is generated and shown in Figure 4 to compare the execution durations of the algorithms. Algorithms are terminated after they have completed 40 iterations. Figure 4 displays the runtimes of GA and DRCEC for each iteration of the D045\_3 dataset, demonstrating that DRCEC has a faster runtime than GA. In addition, because the priority rule sequences are included in the initial population, the DRCEC technique increases convergence and achieves the near-optimal solution in a significantly shorter period.

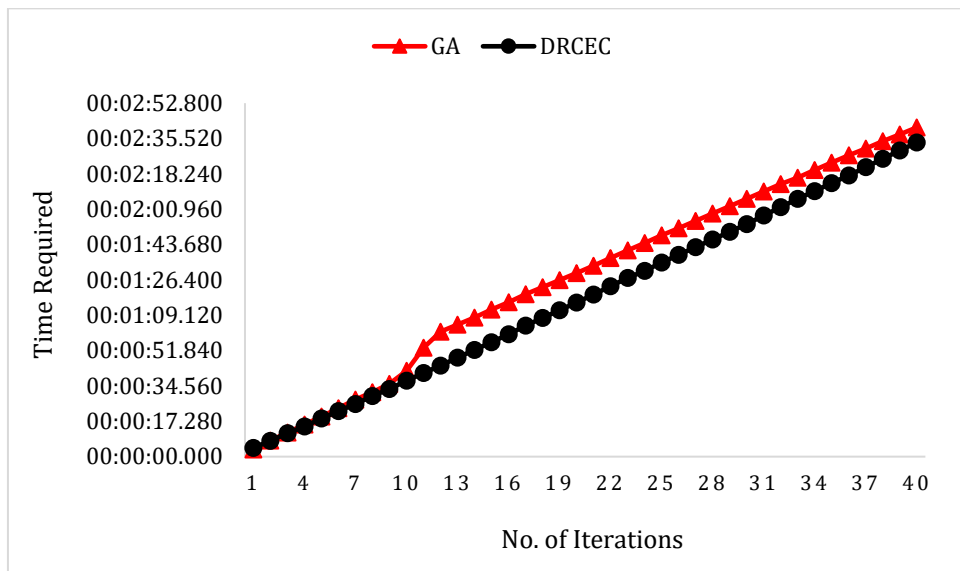


Figure 4. Execution time for GA and DRCEC

### 4.4. Accuracy of PSO, GA and DRCEC

The average total tardiness for all datasets with 10 executions is computed and presented in Table 6. The termination condition is to reach 50 iterations. The optimal value of each dataset is found using the DRCEC algorithm. The reason behind using DRCEC to find near-optimal values is as it performs better than PSO and GA. By reaching around 100 iterations, the first 30 datasets (D005\_1 to D045\_5) with 5 to 45 jobs obtain near-optimal values for the dataset. The datasets D005\_1 to D045\_5 were tested for all possible iterations, and it was revealed that after 100 iterations, it received the best value for total tardiness. At the same time, the dataset with 100 jobs, that means D100\_1 to D100\_5, is tested with all possible iterations and observed that after 500 iterations, it obtains a near-optimal value for total tardiness. The optimal value of each dataset from D005\_1 to D100\_5 is computed and recorded in Table 6.

The relative error in the algorithm is calculated concerning the optimal value obtained by the algorithm after 100 iterations for datasets with 5, 10, 15, 25, 35 and 45 jobs and 500 iterations for the dataset with 100 jobs. The relative error is represented by Eq. (10).

$$Relative\ error = \frac{Average\ Tardiness\ of\ 10\ runs\ for\ the\ dataset - Optimal\ value\ for\ the\ dataset}{Optimal\ value\ for\ the\ Dataset} \quad (10)$$



Optimal job scheduling to minimize total tardiness by dispatching rules and community...

The algorithm's accuracy is how close the calculated tardiness value is to the optimal value of the dataset. Thus the accuracy of the algorithm can be represented by Eq. (11).

$$Accuracy = (1 - Relative\ Error\ for\ algorithm) \quad (11)$$

$$\% Accuracy = (1 - Relative\ Error\ for\ algorithm) * 100$$

The relative error and percentage accuracy for all the datasets are figured out and expressed in Table 6. The average percentage accuracy of algorithms is presented in Table 7. It is found that for generating an optimal sequence of jobs, DRCEC achieves an average accuracy of 99.59%. The range of relative error for 35 datasets with GA is 0.59, while for DRCEC it is 0.043. Using DRCEC, 22 out of 35 datasets attain near-optimal solution after 50 iterations, while only 11 out of 35 datasets could reach a near-optimal solution with an error of 0.0006 to 0.5904 using GA. It is observed that the PSO method performs poorly because only small datasets with five jobs may reach the best solution. Figure 5 represents the graphical representation of accuracy calculated for PSO, GA and DRCEC approaches. Experimental data in Figure 5 shows that the accuracy of GA and DRCEC is almost the same for small and medium datasets, but for large datasets and datasets with 100 jobs, the accuracy differs. When the number of jobs rises, the accuracy of the DRCEC approach is superior to GA. When compared to GA and DRCEC, the PSO algorithm is less accurate.

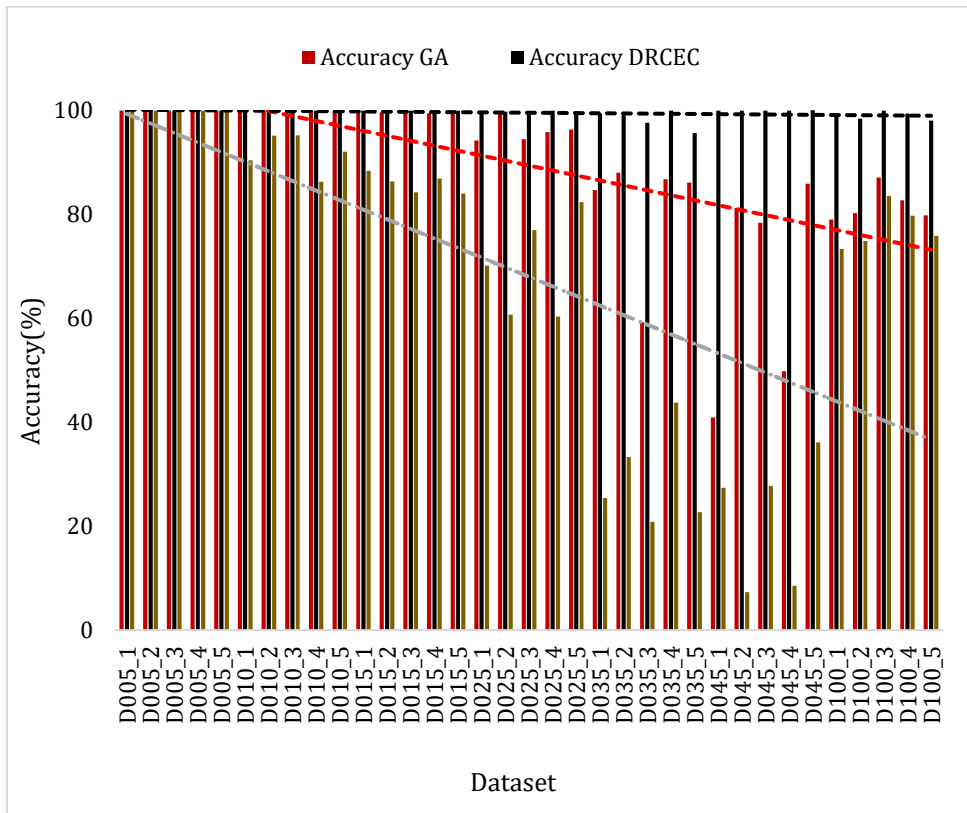


Figure 5. Visual accuracy of PSO, GA and DRCEC

**Table 6.** Relative error and Accuracy of GA and DRCEC approach

Dataset	Average Tardiness			Optimal Tardiness	Relative error			Accuracy		
	DRCEC	GA	PSO		DRCEC	GA	PSO	DRCEC	GA	PSO
D005_1	188	188	188	188	0	0	0	100	100	100
D005_2	160	160	160	160	0	0	0	100	100	100
D005_3	180	180	180	180	0	0	0	100	100	100
D005_4	275	275	275	275	0	0	0	100	100	100
D005_5	107	107	107	107	0	0	0	100	100	100
D010_1	401	401	439.3	401	0	0	0.0955	100	100	90.45
D010_2	448	448	469.6	448	0	0	0.0482	100	100	95.18
D010_3	425	425	445.2	425	0	0	0.0475	100	100	95.25
D010_4	315	315	358.2	315	0	0	0.1371	100	100	86.29
D010_5	309	309	333.5	309	0	0	0.0792	100	100	92.08
D015_1	780	780.8	870.3	780	0	0.001	0.1157	100	99.9	88.43
D015_2	845	847.8	960.1	845	0	0.0033	0.1362	100	99.67	86.38
D015_3	796	796.5	921.2	796	0	0.0006	0.1572	100	99.94	84.28
D015_4	842	846.4	952.2	842	0	0.0052	0.1308	100	99.48	86.92
D015_5	735	735	852.3	735	0	0	0.1595	100	100	84.05

D025_1	1776.1	1867.3	2292.5	1766	0	0.0573	0.2981	100	94.27	70.19
D025_2	1865.1	1867.3	2593	1862	0	0.0028	0.3925	100	99.72	60.75
D025_3	1979.5	2083.2	2429.5	1975	0	0.0547	0.2301	100	94.53	76.99
D025_4	1516	1578.9	2117	1516	0	0.0414	0.3964	100	95.86	60.36
D025_5	1762.5	1824	2070.4	1760	0	0.0363	0.1763	100	96.37	82.37
D035_1	888.9	1019.1	3480.9	884	0	0.1528	0.746	100	84.72	25.4
D035_2	1469	1630.6	4371.8	1456	0	0.1199	0.6669	100	88.01	33.31
D035_3	951.9	1309.1	4460.3	930	0.02	0.4076	0.7914	98	59.24	20.86
D035_4	1755	1986.9	4009.4	1755	0	0.1321	0.5622	100	86.79	43.78
D035_5	816.7	891.7	3451	783	0.04	0.1388	0.7731	96	86.12	22.69
D045_1	2334	3712.2	8519.4	2334	0	0.5904	0.726	100	40.96	27.4
D045_2	456	541.6	6261.3	456	0	0.1877	0.9271	100	81.23	7.29
D045_3	1571	1910	5663.1	1571	0	0.2157	0.7225	100	78.43	27.75
D045_4	536	805	6274.3	536	0	0.5018	0.9145	100	49.82	8.55
D045_5	2222	2536.9	6156.3	2224	0	0.1406	0.6387	100	85.94	36.13
D100_1	173909	208255	218023.3	172159	0.01	0.2096	0.2664	99	79.04	73.36
D100_2	172287	203094	212211.3	169652	0.01	0.1971	0.2508	99	80.29	74.92
D100_3	202532	228643	235768.6	202532	0	0.1289	0.1641	100	87.11	83.59
D100_4	199928	233148	239032.4	198851	0	0.1724	0.202	100	82.76	79.8
D100_5	182959	215603	222784.9	179474	0.01	0.2013	0.2413	99	79.87	75.87

**Table 7.** Average Accuracy of algorithms

Algorithm	Average Accuracy
PSO	68.01
GA	89.43
DRCEC	99.59

## 5. Case Study: Application of DRCEC in Pipe Fittings and Flanges Industry

A case study was conducted with a pipe fittings and flanges manufacturing company to verify methods and compare results to real-world scenarios. Table 8 shows information from a company datasheet. In this dataset, there are 90 jobs processed on a lathe machine and 31 jobs done on a vertical milling centre (VMC) machine. For every 90 and 31 jobs in the dataset, there are two scenarios with different processing times. The model developed for the case study uses the stochastic scenarios scheduling method explained in section 3.4. The model is created for multiple jobs and one machine scenario, which means scheduling will be done for any number of jobs as long as they are processed on one machine. With respect to Table 8, the simulation model assigns job numbers based on the sizes of the individual components; that is, job numbers are assigned based on the 'Size (NB)' column. The model also calculates the processing time of the respective job, in minutes, by taking the product of the values from the columns 'Qty.' and 'Processing Time'. The processing time is converted to hours because the jobs' due dates are also expressed in hours. The data required for scheduling operations are prepared and recorded for lathe and VMC machines. The reformed sample dataset for the VMC machine is shown in Table 9. Then it is restructured into the dataset using procedure 11 and shown in Table 10. It displays the dataset as a data frame; each row represents a job with a job number, job name and processing time.

**Table 8.** Dataset of pipe fittings and flanges industry

Item	Specification	Size (NB)	Qty.	Processing Time	Process Used
Couplings (Half)	Half couplings, dimensions as per ANSI B16.11, screwed to BSPT (F) end, 3000#, material as per ASTM A 105 duly galvanized	15	31	10 min / 1 no	Vertical Drilling On 3 Axis VMC
		20	32	8 min / 1 no	
		25	25	6 min / 1 no	
		32	14	11 min / 1 no	
		40	12	10 min / 1 no	
		50	10	8 min / 1 no	
Couplings (Full)	Full couplings, dimensions as per	15	20	22 min / 1 no	
		20	11	18 min / 1 no	

Item	Specification	Size (NB)	Qty.	Processing Time	Process Used
Flanges	ANSI B16.11, screwed to BSPT (F) end, 3000#, material as per ASTM A 105 duly galvanized	25	10	16 min / 1 no	Vertical Drilling On 3 Axis VMC
		40	12	20 min / 1 no	
	Slip on, raised face, serrated finish flange, dimensions as per ANSI B 16.5, 150#, Material to IS 2062 Gr. A. duly galvanized	15	2	14 min / 1 no	Vertical Boring
		20	2	12 min / 1 no	
		25	46	16 min / 1 no	Vertical Drilling 3 Axis VMC
		32	35	13 min / 1 no	
		40	25	18 min / 1 no	
		50	22	12 min / 1 no	
		65	21	10 min / 1 no	
		80	11	14 min / 1 no	
		100	10	18 min / 1 no	
		125	7	14 min / 1 no	
		150	8	15 min / 1 no	
		200	9	13 min / 1 no	
		250	12	14 min / 1 no	
		300	10	16 min / 1 no	
		350	13	17 min / 1 no	
		400	16	15 min / 1 no	
		450	14	16 min / 1 no	
		500	11	10 min / 1 no	
550	12	14 min / 1 no			
600	10	13 min / 1 no			

\*A dataset of 90 jobs processed on a lathe machine is excluded in the article due to space constraints.

**Table 9.** Reformed dataset of VMC for pipe fittings and flanges industry

Job Number	Job Name	Scenario 1	Scenario 2
		Processing Time	Processing Time
1	Couplings (Half) (15)	5.166	4.2
2	Couplings (Half) (20)	4.266	4.65
3	Couplings (Half) (25)	2.5	3.45
4	Couplings (Half) (32)	2.566	4.2
5	Couplings (Half) (40)	2	4.35
6	Couplings (Half) (50)	1.333	6.6
7	Couplings (Full) (15)	7.333	4.65
8	Couplings (Full) (20)	3.3	5.4

Job Number	Job Name	Scenario 1	Scenario 2
		Processing Time	Processing Time
9	Couplings (Full) (25)	2.666	6.6
10	Couplings (Full) (40)	4	5.7
11	Couplings (Full) (50)	4.333	4.8
12	Flanges (15)	0.466	19.133
13	Flanges (20)	0.4	22.4
14	Flanges (25)	12.266	166.133
15	Flanges (32)	7.583	74.2
16	Flanges (40)	7.5	118.066
17	Flanges (50)	4.4	79.333
18	Flanges (65)	3.5	77.7
19	Flanges (80)	2.566	112.233
20	Flanges (100)	3	100.333
21	Flanges (125)	1.633	98.466
22	Flanges (150)	2	46.2
23	Flanges (200)	1.95	99.4
24	Flanges (250)	2.8	98.233
25	Flanges (300)	2.666	98.466
26	Flanges (350)	3.683	70
27	Flanges (400)	4	64.4
28	Flanges (450)	3.733	65.333
29	Flanges (500)	1.833	46.9
30	Flanges (550)	2.8	46.2
31	Flanges (600)	2.166	28

**Table 10.** Restructured dataset for pipe fittings and flanges industry after applying Procedure 11

Job Number	Job Name	Average Processing Time
1	Couplings (Half) (15)	4.68
2	Couplings (Half) (20)	4.45
3	Couplings (Half) (25)	2.97
4	Couplings (Half) (32)	3.38
5	Couplings (Half) (40)	3.17
6	Couplings (Half) (50)	3.96
7	Couplings (Full) (15)	5.99
8	Couplings (Full) (20)	4.35
9	Couplings (Full) (25)	4.63
10	Couplings (Full) (40)	4.85
11	Couplings (Full) (50)	4.56
12	Flanges (15)	9.79
13	Flanges (20)	11.4
14	Flanges (25)	89.19
15	Flanges (32)	40.89
16	Flanges (40)	62.78
17	Flanges (50)	41.86

Job Number	Job Name	Average Processing Time
18	Flanges (65)	40.6
19	Flanges (80)	57.39
20	Flanges (100)	51.66
21	Flanges (125)	50.04
22	Flanges (150)	24.1
23	Flanges (200)	50.67
24	Flanges (250)	50.51
25	Flanges (300)	50.56
26	Flanges (350)	36.84
27	Flanges (400)	34.2
28	Flanges (450)	34.53
29	Flanges (500)	24.36
30	Flanges (550)	24.5
31	Flanges (600)	15.08

The dataset displayed in Table 10 and a due date provided by the client of four months, or 700 hours (assuming 25 working days per month and seven working hours per day), are used as input for the DRCEC algorithm explained in section 3.3. When tardiness was calculated using the due date of 700 hours for the job processed on the lathe machine, the total tardiness was found to be zero. Further testing with different due dates reveals that the tardiness for the lathe machine is zero for due dates more than or equal to 548. Thus the dataset for the lathe machine is tested for different due dates such as 400, 450, 500, 548, 600, and 700 hours, while the dataset for the VMC machine is tested for a due date of 700 hours. Finally, the DRCEC, GA and PSO algorithms are applied to the pipe fittings and flanges industry data for lathe and VMC machines considering the termination conditions of 10, 20, 30, 40, and 50 iterations with 10 runs each. Table 11 (a) and (b) show the average tardiness with a standard deviation of 10 runs for the VMC and lathe machine datasets, respectively. From the standard deviation, it is observed that DRCEC has no spread in tardiness values for all 10 runs. Thus convergence of the DRCEC algorithm is more significant than the GA and PSO algorithms. The convergence of the algorithm for varied iterations is shown in Figure 6. Figure 6 (a) presents the convergence of DRCEC, while Figure 6 (b) and Figure 6 (c) describe the convergence of GA and PSO algorithms respectively for the VMC machine. One can see from Figure 6 (c) that the PSO needs additional iterations to reach convergence. It is observed that in GA the value of tardiness is inconsistent for iterations and convergence of tardiness is after 20 iterations; on the other hand, in DRCEC, the value is steady from iteration one. Thus, convergence is more significant than GA.

Error and accuracy of DRCEC, GA and PSO are computed by applying Eq. (10) and Eq. (11). These are displayed in Table 12 (a) and (b) for VMC and lathe machines, respectively. It is experimental that the DRCEC algorithm's performance is remarkable compared to GA and PSO. The error is zero for a varied number of iterations; thus, the accuracy is 100 % in DRCEC. The accuracy is visually represented in Figure 7 and Figure 8 for VMC and lathe machines, respectively. Results of the FCFS utilized in a manufacturing company for job scheduling are shown in Table 13. It has been noted that the DRCEC results in less tardiness than the conventional method employed in the company. As a result, the manufacturing company reduce costs while increasing production efficiency.

**Table 11 (a).** Average tardiness and std. deviation of the VMC machine

Machine	VMC Machine					
Algorithm	DRCEC		GA		PSO	
Due Date	700		700		700	
Tardiness, Std. Deviation / Iterations	T	$\sigma$	T	$\sigma$	T	$\sigma$
10	206.96	0	218.75	21.14	206.690	0
20	206.96	0	207.55	0.72	232.646	36.63
30	206.96	0	207.68	2.16	219.014	14.67
40	206.96	0	206.96	0	216.569	8.76
50	206.96	0	207.26	0.59	239.811	59

T - Tardiness     $\sigma$  - Standard Deviation

**Table 11 (b).** Average tardiness and std. deviation of lathe machine

DRCEC							
548-700		500		450		400	
T	$\sigma$	T	$\sigma$	T	$\sigma$	T	$\sigma$
0	0	47.6	0	129.95	0	254.18	0
0	0	47.6	0	129.95	0	254.18	0
0	0	47.6	0	129.95	0	254.18	0
0	0	47.6	0	129.95	0	254.18	0
0	0	47.6	0	129.95	0	254.18	0
GA							
548-700		500		450		400	
T	$\sigma$	T	$\sigma$	T	$\sigma$	T	$\sigma$
0	0	47.6	0.14	133.45	3.87	293.48	39.55
0	0	47.6	0	130.66	2.13	263.37	11.49
0	0	47.6	0	131.45	4.5	260.47	6.99
0	0	47.6	0	130.66	2.13	262.41	14.18
0	0	47.6	0	130.66	2.13	262.1	7.56
PSO							
548-700		500		450		400	
T	$\sigma$	T	$\sigma$	T	$\sigma$	T	$\sigma$
0	0	47.6	0.14	221.711	82.49	553.627	168.84
0	0	47.6	0	177.013	38.89	511.977	89.48
0	0	47.6	0	205.155	87.85	661.152	317.07
0	0	47.6	0	129.94	0	545.064	398.27
0	0	47.6	0	226.537	87.23	592.07	224.49

T - Tardiness     $\sigma$  - Standard Deviation

**Table 12 (a).** Error and accuracy of DRCEC, GA and PSO algorithm for VMC machine

Machine	VMC Machine		
Algorithm	DRCEC	GA	PSO



Optimal job scheduling to minimize total tardiness by dispatching rules and community...

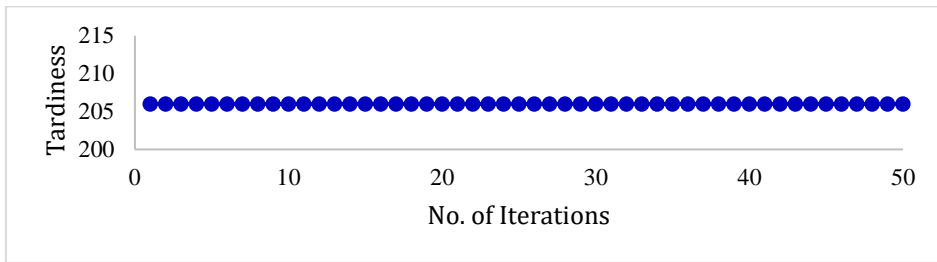
Due Date	700		700		700	
Error, Accuracy / Iterations	E	A	E	A	E	A
10	0	100	0.0569	94.31	0	100
20	0	100	0.0028	99.72	0.1241	87.59
30	0	100	0.0034	99.66	0.0582	94.18
40	0	100	0	100	0.0464	95.36
50	0	100	0.0014	99.86	0.1587	84.13

E - Error A - Accuracy in %

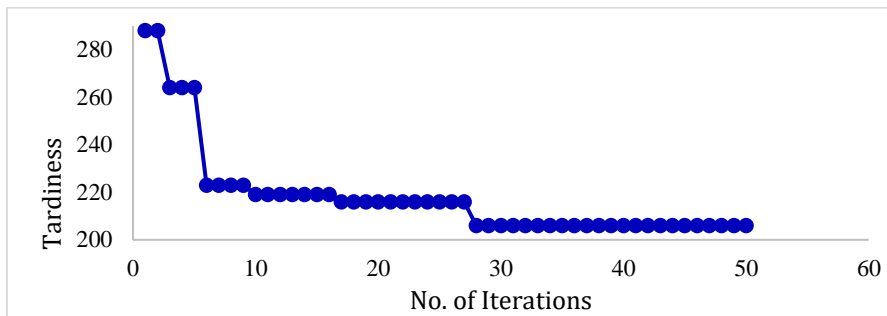
**Table 12 (b).** Error and accuracy of DRCEC, GA and PSO algorithm for lathe machine

DRCEC							
548-700		500		450		400	
E	A	E	A	E	A	E	A
0	100	0	100	0	100	0	100
0	100	0	100	0	100	0	100
0	100	0	100	0	100	0	100
0	100	0	100	0	100	0	100
0	100	0	100	0	100	0	100
GA							
548-700		500		450		400	
E	A	E	A	E	A	E	A
0	100	0.001	99.89	0.0269	97.3	0.1546	84.53
0	100	0	100	0.0055	99.45	0.0361	96.38
0	100	0	100	0.0115	98.84	0.0247	97.52
0	100	0	100	0.0055	99.45	0.0324	96.76
0	100	0	100	0.0055	99.45	0.0311	96.88
PSO							
548-700		500		450		400	
E	A	E	A	E	A	E	A
0	100	0	100	0.4138	58.62	0.5408	45.92
0	100	0	100	0.2658	73.42	0.5035	49.65
0	100	0	100	0.3665	63.35	0.6155	38.45
0	100	0	100	0	100	0.5336	46.64
0	100	0	100	0.4263	57.37	0.5706	42.94

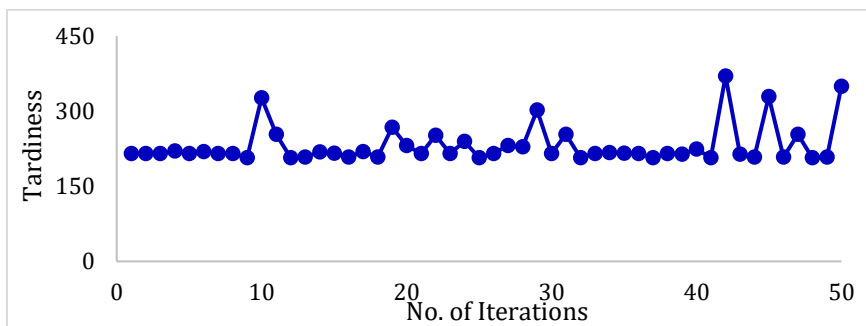
E - Error A- Accuracy in %



(a) DRCEC

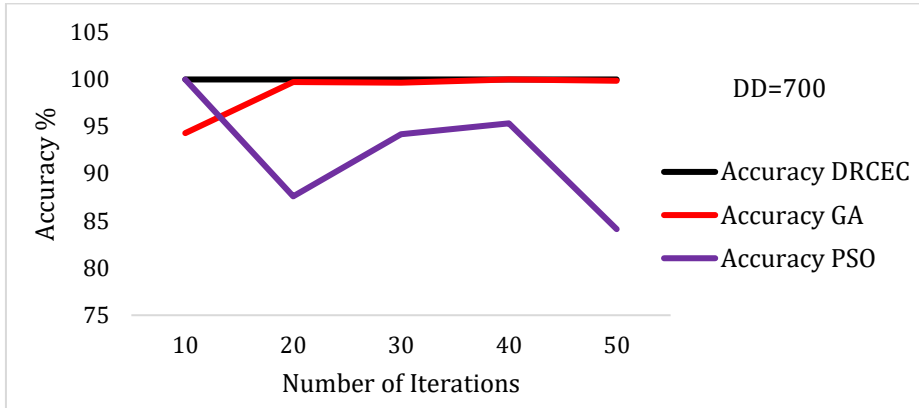


(b) GA

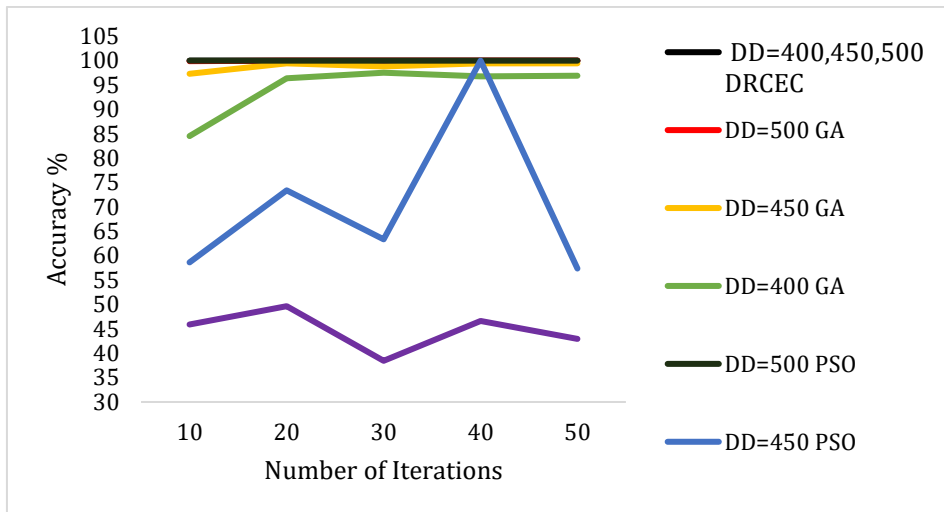


(c) PSO

**Figure 6.** Convergence of the algorithm



**Figure 7.** Accuracy of DRCEC, GA and PSO algorithm for VMC machine



**Figure 8.** Accuracy of DRCEC, GA and PSO algorithm for lathe machine

**Table 13.** Validation of case study

Machine Name	LATHE Machine			VMC Machine	
The method used in the company for scheduling jobs			FCFS		FCFS
Due Date	548-700	500	450	400	700
Total Tardiness	0	1200.55	4206.48	7691.41	494.07

## 6. Statistical test validation with ANOVA

This section presents statistical testing with Analysis of variance (ANOVA) which compares the means of PSO, GA and DRCEC algorithms. The absence of a discernible

difference between the means of the techniques used in this study is the null hypothesis that is investigated. ANOVA just reports that there is a significant difference between the approaches used but does not specify where those differences occur if the null hypothesis is rejected. Thus Post hoc tests are performed to pinpoint the locations of the techniques differences. The ANOVA test is performed on the average taken from 10 runs for two samples each from small, medium and large datasets in JASP open source statistical software. For datasets with five jobs, the standard deviation is 0, making the ANOVA test insignificant. The results of the significance test for other datasets are presented in Table 14 at a significance level of ( $\alpha=0.05$ ). It is observed that the main ANOVA table shows that the F-statistic is significant as the p-value is less than  $\alpha = 0.05$ . Later, post Hoc that means “after the fact” comparisons are also performed and the  $p_{tukey}$  value shows there is a difference between each pair of approaches, except in dataset D010\_5 and D015\_5 where there is less difference in DRCEC and GA. Figure 9 shows the descriptive plot with error bars for sample datasets which reveals that the performance of DRCEC is better than GA and PSO.

**Table 14.** Validation of results with ANOVA

ANOVA - D010_5-Tardiness					
Cases	Sum of Squares	df	Mean Square	F	p
Algorithm	5228.169	2	2614.085	12.104	0.001
Residuals	2591.6	12	215.967		
Post Hoc Comparisons - Algorithm					
		Mean Difference	SE	t	$p_{tukey}$
DRCEC	GA	-2.38	9.294	-0.256	0.965
	PSO	-40.74	9.294	-4.383	0.002
GA	PSO	-38.36	9.294	-4.127	0.004
ANOVA - D015_5-Tardiness					
Cases	Sum of Squares	df	Mean Square	F	p
Algorithm	39924.784	2	19962.392	26.35	<.001
Residuals	9090.86	12	757.572		
Post Hoc Comparisons - Algorithm					
		Mean Difference	SE	t	$p_{tukey}$
DRCEC	GA	-11.08	17.408	-0.636	0.803
	PSO	-114.56	17.408	-6.581	<.001
GA	PSO	-103.48	17.408	-5.944	<.001

ANOVA - D025_5-Tardiness					
Cases	Sum of Squares	df	Mean Square	F	p
Algorithm	271456.121	2	135728.061	175.822	< .001
Residuals	9263.568	12	771.964		

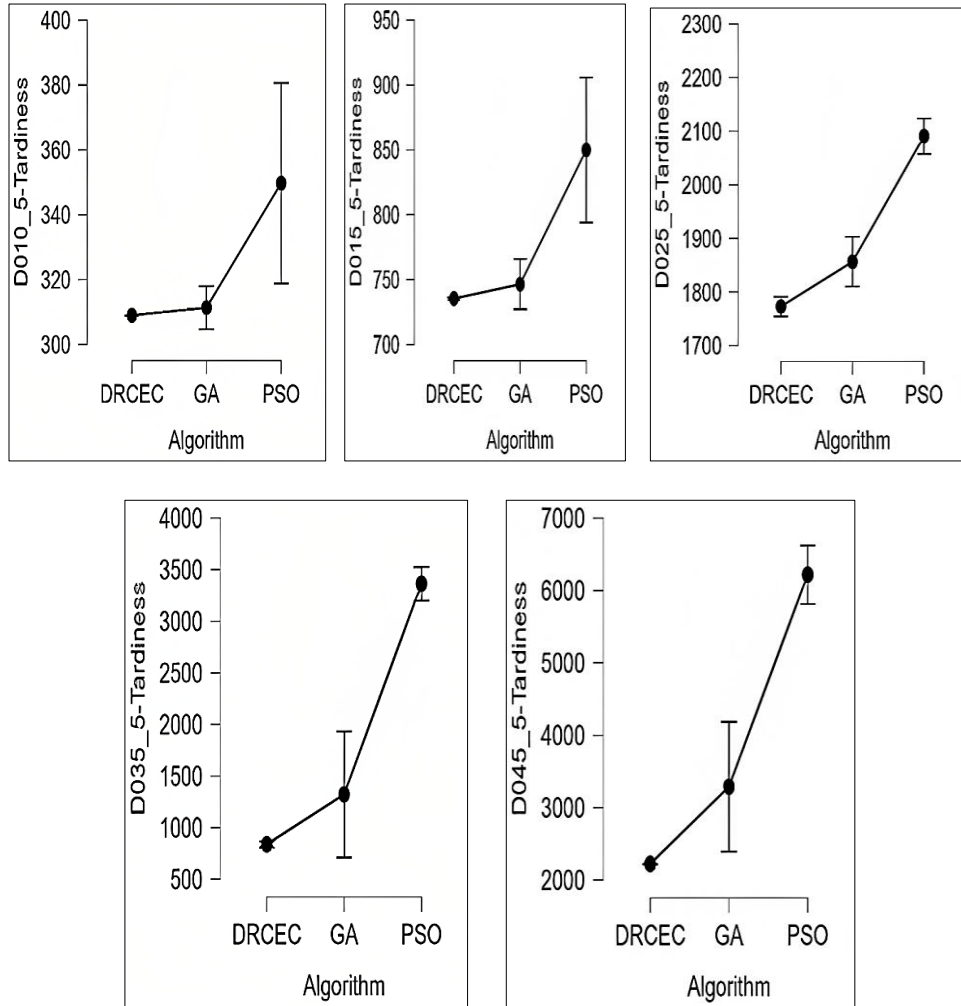
Post Hoc Comparisons - Algorithm					
		Mean Difference	SE	t	p <sub>Tukey</sub>
DRCEC	GA	-83.4	17.572	-4.746	0.001
	PSO	-317.78	17.572	-18.084	< .001
GA	PSO	-234.38	17.572	-13.338	< .001

ANOVA - D035_5-Tardiness					
Cases	Sum of Squares	df	Mean Square	F	p
Algorithm	1.795×10 <sup>7</sup>	2	8.973×10 <sup>6</sup>	103.836	< .001
Residuals	1.037×10 <sup>6</sup>	12	86419.041		

Post Hoc Comparisons - Algorithm					
		Mean Difference	SE	t	p <sub>Tukey</sub>
DRCEC	GA	-483.36	185.924	-2.6	0.056
	PSO	-2523.96	185.924	-13.575	< .001
GA	PSO	-2040.6	185.924	-10.975	< .001

ANOVA - D045_5-Tardiness					
Cases	Sum of Squares	df	Mean Square	F	p
Algorithm	4.279×10 <sup>7</sup>	2	2.140×10 <sup>7</sup>	102.374	< .001
Residuals	2.508×10 <sup>6</sup>	12	208991.186		

Post Hoc Comparisons - Algorithm					
		Mean Difference	SE	t	p <sub>Tukey</sub>
DRCEC	GA	-1065.84	289.131	-3.686	0.008
	PSO	-3994.88	289.131	-13.817	< .001
GA	PSO	-2929.04	289.131	-10.131	< .001



**Figure 9.** Descriptive plots- Error bars for DRCEC, GA, and PSO

## 7. Conclusions

Dispatching rules alone are ineffective in sequencing the jobs to reduce tardiness, but when combined with the evolutionary approach, they produce good results. This research combines dispatching rules with community evaluation chromosomes to create an optimal sequence with a lower tardiness performance measure. A total of 35 datasets were tested for 10, 20, 30, 40, and 50 iterations to compute the tardiness with DRCEC. The results were compared with PSO and GA, revealing that the total tardiness with the DRCEC approach is less than PSO and GA for all the datasets, especially with a large number of jobs. The DRCEC algorithm was used to create optimal sequences for three stochastic scheduling models that were further examined in the study. It is also witnessed that the convergence process of the DRCEC is much faster than GA and PSO. In the DRCEC method, the best sequence of each iteration is inserted into the population of the next iteration, enhancing the fitness function of all chromosomes and promising its convergence. Using DRCEC, 22 out of 35 datasets attain optimal solution

Optimal job scheduling to minimize total tardiness by dispatching rules and community ...

after 50 iterations, while only 11 out of 35 datasets could reach an optimal solution using GA. According to the average execution time of the algorithms, the proposed algorithm is faster. The algorithm has been tested for a case study in a manufacturing company that specializes in pipe fittings and flanges using a lathe and a VMC machine, and it outperforms. Finally, based on the results of the manufacturing company, the strong performance of DRCEC versus GA and PSO is proved. The statistical test validation with ANOVA showed that there are substantial differences between the methods employed and that the DRCEC outperforms GA and PSO. It has been noted that the traditional FCFS method gives tardiness values of 494.07 hours considering a due date of 700 hours for the VMC machine and 1200.55, 4206.48, and 7691.41 hours considering the due date of 500, 450 and 400 hours respectively for lathe machine, whereas DRCEC gives values of 206.96 hours for VMC machine and 47.6, 129.95, and 254.18 hours for lathe, which are significantly less. This helps the company increase production efficiency.

The study has a limitation which includes fixed parameters, like population size, mutation rate, and crossover rate. Parameter adjustments are not incorporated to elude recordings of numerous experimental tests. A total of 5250 experimental tests were performed with the fixed parameters.

Work on this research in the future could include designing environments for several machines, giving the company a wide array of options. Combinatorial approaches can be used by researchers to improve the accuracy of scheduling issues when combined with other different metaheuristic techniques. Furthermore, changing the stand-alone design parameters for web apps allows the operator to use it remotely on any device.

**Author Contributions:** Conceptualization, methodology, software, validation, formal analysis, investigation, writing original draft preparation, P. Bari.; writing, review, editing, visualization, supervision, P. Karande; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The datasets generated for this study are available on request to the corresponding author.

**Acknowledgements:** The authors gratefully acknowledge the editors' and anonymous referees' helpful remarks and constructive ideas.

**Conflicts of 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.

## References

- Ahmadian, M. M., Salehipour, A., & Cheng, T. C. E. (2021). A meta-heuristic to solve the just-in-time job-shop scheduling problem. *European Journal of Operational Research*, 288(1), 14–29. <https://doi.org/10.1016/j.ejor.2020.04.017>
- Anjana, V., Sridharan, R., & Ram Kumar, P. N. (2020). Metaheuristics for solving a multi-objective flow shop scheduling problem with sequence-dependent setup times. *Journal of Scheduling*, 23(1), 49–69. <https://doi.org/10.1007/s10951-019-00610-0>

Ardakani, A., Fei, J., & Beldar, P. (2020). Truck-to-door sequencing in multi-door cross-docking system with dock repeat truck holding pattern. *International Journal of Industrial Engineering Computations*, 11(2), 201–220. <https://doi.org/10.5267/j.ijiec.2019.10.001>

Baker, K. R., & Trietsch, D. (2009). Principles of Sequencing and Scheduling. In *Principles of Sequencing and Scheduling*. <https://doi.org/10.1002/9780470451793>

Bank, M., Fatemi Ghomi, S. M. T., Jolai, F., & Behnamian, J. (2012). Two-machine flow shop total tardiness scheduling problem with deteriorating jobs. *Applied Mathematical Modelling*, 36(11), 5418–5426. <https://doi.org/10.1016/j.apm.2011.12.010>

Bari, P., & Karande, P. (2021). Application of PROMETHEE-GAIA method to priority sequencing rules in a dynamic job shop for single machine. *Materials Today: Proceedings*, 46(17), 7258–7264. <https://doi.org/10.1016/j.matpr.2020.12.854>

Bari, P., Karande, P., & Menezes, J. (2022). Simulation of job sequencing for stochastic scheduling with a genetic algorithm. *Operational Research in Engineering Sciences: Theory and Applications*, 5(3), 17–39. <https://doi.org/10.31181/oresta060722075b>

Carlier, J., & Pinson, E. (1989). An algorithm for solving the job-shop problem. *Management Science*, 35(2), 164–176. <https://doi.org/10.1287/mnsc.35.2.164>

Cayo, P., & Onal, S. (2020). A shifting bottleneck procedure with multiple objectives in a complex manufacturing environment. *Production Engineering*, 14(2), 177–190. <https://doi.org/10.1007/s11740-019-00947-7>

Czibula, O. G., Gu, H., Hwang, F. J., Kovalyov, M. Y., & Zinder, Y. (2016). Bi-criteria sequencing of courses and formation of classes for a bottleneck classroom. *Computers and Operations Research*, 65, 53–63. <https://doi.org/10.1016/j.cor.2015.06.010>

Dao, T. K., Pan, T. S., Nguyen, T. T., & Pan, J. S. (2018). Parallel bat algorithm for optimizing makespan in job shop scheduling problems. *Journal of Intelligent Manufacturing*, 29(2), 451–462. <https://doi.org/10.1007/s10845-015-1121-x>

Defersha, F. M., & Rooyani, D. (2020). An efficient two-stage genetic algorithm for a flexible job-shop scheduling problem with sequence dependent attached/detached setup, machine release date and lag-time. *Computers and Industrial Engineering*, 147, 106605. <https://doi.org/10.1016/j.cie.2020.106605>

French, S. (1982). *Sequencing and scheduling: An introduction to the mathematics of the job-shop (First)*. Wiley.

Garey, M. R., Johnson, D. S., & Sethi, R. (1976). Complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, 1(2), 117–129. <https://doi.org/10.1287/moor.1.2.117>

Gil-Gala, F. J., Đurasević, M., Varela, R., & Jakobović, D. (2023). Ensembles of priority rules to solve one machine scheduling problem in real-time. *Information Sciences*, 634, 340–358. <https://doi.org/10.1016/j.ins.2023.03.114>

Gong, G., Deng, Q., Chiong, R., Gong, X., & Huang, H. (2019). An effective memetic algorithm for multi-objective job-shop scheduling. *Knowledge-Based Systems*, 182, 104840. <https://doi.org/10.1016/j.knosys.2019.07.011>

Gupta, A., & Chauhan, S. R. (2015). A heuristic algorithm for scheduling in a flow shop environment to minimize makespan. *International Journal of Industrial Engineering Computations*, 6(2), 173–184. <https://doi.org/10.5267/j.ijiec.2014.12.002>

Habib Zahmani, M., & Atmani, B. (2021). Multiple dispatching rules allocation in real



time using data mining, genetic algorithms, and simulation. *Journal of Scheduling*, 24(2), 175–196. <https://doi.org/10.1007/s10951-020-00664-5>

Hasan, S. M. K., Sarker, R., Essam, D., & Cornforth, D. (2009). Memetic algorithms for solving job-shop scheduling problems. *Memetic Computing*, 1(1), 69–83. <https://doi.org/10.1007/s12293-008-0004-5>

He, L., Li, W., Chiong, R., Abedi, M., Cao, Y., & Zhang, Y. (2021). Optimising the job-shop scheduling problem using a multi-objective Jaya algorithm. *Applied Soft Computing*, 111, 107654. <https://doi.org/10.1016/j.asoc.2021.107654>

Kumar, A., Garg, P., Pant, S., Ram, M., & Kumar, A. (2022). Multi-criteria decision-making techniques for complex decision making problems. *Mathematics in Engineering, Science & Aerospace*, 13(2), 791–803.

Kumar, K. K., Nagaraju, D., Gayathri, S., & Narayanan, S. (2017). Evaluation and Selection of Best Priority Sequencing Rule in Job Shop Scheduling using Hybrid MCDM Technique. *IOP Conference Series: Materials Science and Engineering*, 197(1). <https://doi.org/10.1088/1757-899X/197/1/012059>

Lee, W. C., Yeh, W. C., & Chung, Y. H. (2014). Total tardiness minimization in permutation flowshop with deterioration consideration. *Applied Mathematical Modelling*, 38(13), 3081–3092. <https://doi.org/10.1016/j.apm.2013.11.031>

Li, X., Chen, L., Xu, H., & Gupta, J. N. D. (2015). Trajectory scheduling methods for minimizing total tardiness in a flowshop. *Operations Research Perspectives*, 2, 13–23. <https://doi.org/10.1016/j.orp.2014.12.001>

M'Hallah, R. (2014). Minimizing total earliness and tardiness on a permutation flow shop using VNS and MIP. In *Computers and Industrial Engineering* 75(1). Elsevier Ltd. <https://doi.org/10.1016/j.cie.2014.06.011>

Martínez, K. P., Adulyasak, Y., Jans, R., Morabito, R., & Toso, E. A. V. (2019). An exact optimization approach for an integrated process configuration, lot-sizing, and scheduling problem. *Computers and Operations Research*, 103, 310–323. <https://doi.org/10.1016/j.cor.2018.10.005>

Meloni, C., Pacciarelli, D., & Pranzo, M. (2004). A rollout metaheuristic for job shop scheduling problems. *Annals of Operations Research*, 131(1–4), 215–235. <https://doi.org/10.1023/B:ANOR.0000039520.24932.4b>

Negi, G., Kumar, A., Pant, S., & Ram, M. (2021). GWO: A review and applications. *International Journal of System Assurance Engineering and Management*, 12(1), 1–8. <https://doi.org/10.1007/s13198-020-00995-8>

Pinedo, M. L. (2004). *Planning and scheduling in manufacturing and services*. Springer Series in Operations Research and Financial Engineering.

Pranzo, M., & Pacciarelli, D. (2016). An iterated greedy metaheuristic for the blocking job shop scheduling problem. *Journal of Heuristics*, 22(4), 587–611. <https://doi.org/10.1007/s10732-014-9279-5>

Raman, N., Rachamadugu, R. V., & Talbot, F. B. (1989). Real-time scheduling of an automated manufacturing center. *European Journal of Operational Research*, 40(2), 222–242. [https://doi.org/10.1016/0377-2217\(89\)90332-9](https://doi.org/10.1016/0377-2217(89)90332-9)

Sayadi, M. K., Ramezani, R., & Ghaffari-Nasab, N. (2010). A discrete firefly metaheuristic with local search for makespan minimization in permutation flow shop scheduling problems. *International Journal of Industrial Engineering Computations*, 1(1), 1–10. <https://doi.org/10.5267/j.ijiec.2010.01.001>

Schaller, J., & Valente, J. M. S. (2013). A comparison of metaheuristic procedures to schedule jobs in a permutation flow shop to minimise total earliness and tardiness. *International Journal of Production Research*, 51(3), 772–779. <https://doi.org/10.1080/00207543.2012.663945>

Sergienko, I. V., Huliannytskyi, L. F., & Sirenko, S. I. (2009). Classification of applied methods of combinatorial optimization. *Cybernetics and Systems Analysis*, 45(5), 732–741. <https://doi.org/10.1007/s10559-009-9134-0>

T'Kindt, V., & Billaut, J.-C. (2005). *Multicriteria Scheduling - Theory, Models and Algorithms*. Springer-Verlag.

Uniyal, N., Pant, S., Kumar, A., & Pant, P. (2022). Nature-inspired metaheuristic algorithms for optimization. In: Kumar A, Pant S, Ram M, Yadav O (Ed.) *Meta-Heuristic Optimization Techniques: Applications in Engineering*. Berlin, Boston: De Gruyter, 1–10. <https://doi.org/10.1515/9783110716214-001>

Valledor, P., Gomez, A., Puente, J., & Fernandez, I. (2022). Solving rescheduling problems in dynamic permutation flow shop environments with multiple objectives using the hybrid dynamic non-dominated sorting genetic II algorithm. *Mathematics*, 10(14). <https://doi.org/10.3390/math10142395>

Vinod, V., & Sridharan, R. (2011). Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system. *International Journal of Production Economics*, 129(1), 127–146. <https://doi.org/10.1016/j.ijpe.2010.08.017>

Volgenant, A., & Teerhuis, E. (1999). Improved heuristics for the n-job single-machine weighted tardiness problem. *Computers and Operations Research*, 26(1), 35–44. [https://doi.org/10.1016/S0305-0548\(98\)00048-3](https://doi.org/10.1016/S0305-0548(98)00048-3)

Yu, J. M., & Lee, D. H. (2018). Solution algorithms to minimise the total family tardiness for job shop scheduling with job families. *European Journal of Industrial Engineering*, 12(1), 1–23. <https://doi.org/10.1504/EJIE.2018.089876>

Zammori, F., Braglia, M., & Castellano, D. (2014). Harmony search algorithm for single-machine scheduling problem with planned maintenance. *Computers and Industrial Engineering*, 76, 333–346. <https://doi.org/10.1016/j.cie.2014.08.001>

Zhao, Z., Chen, X., An, Y., Li, Y., & Gao, K. (2023). A property-based hybrid genetic algorithm and tabu search for solving order acceptance and scheduling problem with trapezoidal penalty membership function. *Expert Systems with Applications*, 218. <https://doi.org/10.1016/j.eswa.2023.119598>



© 2023 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).