A TWO-STAGE SOLUTION APPROACH FOR PLASTIC INJECTION MACHINES SCHEDULING PROBLEM

TUGBA SARAC
Eskisehir Osmangazi University,
Department of Industrial Engineering
Meselik Campus, 26040, Eskisehir, Turkey

AYDIN SIPAHIOGLU AND EMINE AKYOL OZER*
Eskisehir Osmangazi University,
Department of Industrial Engineering
Meselik Campus, 26040, Eskisehir, Turkey

Eskisehir Technical University,
Department of Industrial Engineering
Iki Eylul Campus, 26555, Eskisehir, Turkey

(Communicated by Gerhard-Wilhelm Weber)

Abstract. One of the most common plastic manufacturing methods is injection molding. In injection molding process, scheduling of plastic injection machines is very difficult because of the complex nature of the problem. For example, similar plastic parts should be produced sequentially to prevent long setup times. On the other hand, to produce a plastic part, its mold should be fixed on an injection machine. Machine eligibility restrictions should be considered because a mold can be usually fixed on a subset of the injection machines. Some plastic parts which have same shapes but different colors are used same mold so these parts can only be scheduled simultaneously if their mold has copies, otherwise resource constraints should be considered. In this study, a multi-objective mathematical model is proposed for parallel machine scheduling problem to minimize makespan, total tardiness, and total waiting time. Since NP-hard nature of problem, this paper presents a two-stage mathematical model and a two-stage solution approach. In the first stage of mathematical model, jobs are assigned to the machines and each machine is scheduled separately in the second stage. The integrated model and two-stage mathematical model are scalarized by using goal programming, compromise programming and Lexicographic Weighted Tchebycheff programming methods. To solve large-scale problems in a short time, a two-stage solution approach is also proposed. In the first stage of this approach, jobs are assigned to machines and scheduled by using proposed simulated annealing algorithm. In the second stage of the approach, starting time, completion time and waiting time of the jobs are calculated by using a mathematical model. The performance of the methods is demonstrated on randomly generated test problems.

2010 Mathematics Subject Classification. Primary: 90C29, 90C90; Secondary: 90B35.
Key words and phrases. Plastic injection, identical parallel machine scheduling, multi-objective programming, integer programming.

* Corresponding author: Emine Akyol Ozer.
1. Introduction. Production by plastic injection is carried out by melting and shaping plastic raw materials in the injection-molding machine. Since plastic parts manufacturers, generally working in the subsidiary industry, are required to strictly adhere to the production schedule of a key company, production must be carried out in such a way that they do not delay the delivery time of the products in question. Otherwise, subsidiary industries may be faced with crucial financial enforcements. This necessity puts a great deal of time pressure on subsidiary industries. On the other hand, there are situations that should be taken into consideration during production. For instance, it is preferred that manufacturing sequence should be followed from light-colored parts to dark-colored parts. The reason for this is to minimize the setup time required for machine cleaning. In addition, for a similar reason, it is requested that raw materials in the machinery should not be changed as much as possible. This is because every raw material change requires a setup time in the machine and, therefore, causes a loss in the makespan. In short, the setup times depend on the manufacturing sequence. Besides, the mold of a product must be able to be fixed on to the appropriate machine in order to be able to manufacture that product. Although injection machines are identical in terms of processing times, there are some machine eligibility constraints because each machine may not be fixed to all machines with different tonnage and dimensions. In addition, in some plastic injection molding companies, there are some specific constraints arising from the nature of the process. For example, jobs that use the same mold can be scheduled simultaneously only if there are copy molds. In other words, parallel machine scheduling is possible if machines use a common resource (mold). Such special circumstances require an efficient scheduling in a manufacturing process where there are a large number of injection machines. Otherwise, it will take a lot of time to prepare machines for manufacturing, and the working time will be wasted. This also leads to inefficiency and even failure to meet the due dates. What is more, the lack of a well-prepared plan causes chaos within the facility. It is therefore evident that there is a need for an efficient method to instantly determine a work schedule that can minimize setup times and consider all the particular conditions mentioned above.

The scheduling problem of plastic injection machines is a special case of the parallel machine scheduling problem. In scheduling literature, it is represented as $P_m / s_{ij}, M_j, \text{resources} / C_{\text{max}}, \sum T_j, \sum w_j$. This representation implies that the problem is a parallel machine scheduling problem and that the objectives are to minimize makespan, total tardiness and total waiting time. There are also three types of constraints in the problem: presence of sequence-based setup time differences between jobs, machine eligibility restrictions because of technical constraints, and using common resource (mold) for jobs.

Parallel machine scheduling problems can have different process characteristics and objective functions. Process characteristics are about the flexibility of resources, whether machines are identical, whether machine eligibility restrictions exist, the presence of setup times, and whether tasks are sequence-dependent or not. Alagoz and Azizoglu [3], Edis and Oguz [9], Edis and Ozkarahan [10], [11], Gacias et al. [13] and Su et al. [23] considered different process characteristics of parallel machine scheduling problems. Since the problem is NP-hard, studies in the literature often used heuristic and meta-heuristic algorithms. Unlu and Mason [26] used a mixed integer programming model for the parallel machine scheduling problem. Li et al. [16] proposed a simulated annealing algorithm to minimize the completion times in
identical parallel machines. Tanev et al. [24] considered the problem of sequencing of customer orders in the plastic injection plant as a flexible jobshop scheduling ($F_{J_L}$) problem. The aim of problem was scheduling of jobs to reduce the number of tardy jobs, the variability of the operation time and the mold changes, but to increase the machine efficiency. Dastidar and Nagi [7] considered a scheduling problem in a plastic injection plant as a parallel machine scheduling problem in which resources with multiple parts, multiple and different capacities exist, and preparation times and costs are sequence-dependent. Their aim was to meet customer demands in a way that minimizes the cost of inventory holding, backlogging and setup time. They developed a two-stage and work center-based decomposition approach for this problem. Sarac and Sipahioglu [22] proposed a mathematical model based on knapsack problem for the scheduling of plastic injection machines and demonstrated the success of the model on small-scale test problems. Keskinturk et al. [15] proposed a mathematical model for the parallel machine problem with the presence of sequence-dependent setup times. They also developed an ant colony algorithm and a genetic algorithm (GA). Chaudhry and Drake [6] proposed a genetic algorithm approach for minimizing the total tardiness and worker assignment problems in identical parallel machines. Liu and Wu [19] developed a genetic algorithm for minimizing the number of tardy jobs in an identical parallel machine scheduling problem, and they achieved successful results for large-scale problems in industry. Ruiz and Romano [21], considered an unrelated parallel machine scheduling problem ($R_{jm}$) and job sequence-dependent setup times. The duration of the setup times depended on assignable resources. They developed a mixed integer programming model to minimize total completion time and the total amount of resources assigned. Afzalirad and Shafipour [1] investigated a resource-constrained unrelated parallel machine scheduling problem with machine eligibility restrictions. They proposed an integer programming model for minimization makespan ($C_{max}$). Due to NP-hardness nature of this problem, two new genetic algorithms including a pure genetic algorithm and a genetic algorithm along with a heuristic procedure are presented. Ezygwu and Akutsah [12] developed a firefly algorithm which is refined with a local search solution improvement mechanism to minimize makespan like Afzalirad and Shafipour [1]. Three different popular metaheuristic algorithms are presented in parallel to verify and measure the effectiveness of the firefly algorithm. They demonstrated that improved firefly algorithm provides good quality solutions for both small and large instances. Baykasoglu and Ozsoydan [4] focused on online and dynamic scheduling of parallel heat treatment furnaces at a real manufacturing company. The problem has significant characteristics: release times ($r_j$), eligibility constraints, due dates, sequence-dependent setup times due to heating up or cooling down the furnaces, breakdowns and maintenance periods. As a solution approach, a multi-start and constructive search algorithm is developed for this dynamic problem. Akyol Ozer and Sarac [2] considered an identical parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and multiple copies of shared resources. Objective function of the problem is to minimize the total weighted completion time and two mixed-integer programming models and a matheuristic algorithm are proposed. In addition to these studies, it can be given different examples of studies using the heuristic approach such that Driessel and Monch [8], Lin et al. [18], Turker and Sel [25], Gokhale and Mathirajan [14], Li et al. [17], Park et al. [20] and Bektur and Sarac [5]. The studies of related literature are also summarized in Table 1.
### Table 1. Literature Review

| Ref. | α | β | γ | Solution Methods |
|------|---|---|---|------------------|
| [24] | FJ, | Resource, | Number of Tardy Jobs | Hybrid evolutionary algorithm |
| [8]  | Pm, | rj, prec, sij | $\sum w_jT_j$ | Variable neighborhood search |
| [7]  | Pm | Resource, sij | Inventory holding, backlogging and setup time | Work center based decomposition approach |
| [15] | Pm | sij | Average relative percentage of imbalance | GA and ant colony optimization |
| [10] | Pm | rj, sij | $C_{max}, \sum T_j$ | 0-1 MIP model and GA with a fuzzy logic controller |
| [18] | Pm | rj, sij | Maximum tardiness($T_{max}$) | Greedy Algorithm |
| [25] | Pm | sij | $C_{max}$ | A hybrid GA and tabu search approach |
| [3]  | Pm | Mj | $\sum C_j$ | Optimizing Algorithm |
| [23] | Pm | Mj | $T_{max}$ | A network flow mathematical programming model and a heuristic approach |
| [17] | Pm | Mj | $C_{max}$ | Simulated annealing algorithm |
| [11] | Pm | Resource, Mj | $C_{max}$ | Integer programming and constraint programming model |
| [21] | Rm | Resource sij | $\sum C_j$, total amount of resources assigned | Mixed integer programming model |
| [11] | Pm | Resource, Mj | $C_{max}$ | Integer programming, constraint programming model |
| [10] | Pm | Resource, Mj | $C_{max}$ | A Combined Integer/Constraint Programming Approach |
| [9]  | Pm | Resource | $C_{max}$ | Integer programming, constraint programming model |
| [1]  | Rm | Mj | $C_{max}$ | An integer programming model, GA |
| [12] | Rm | sij | $C_{max}$ | Firefly algorithm |
| [4]  | Pm | rj, sij, Mj | Energy consumption and annual income | A multi-start and constructive search algorithm |
| [2]  | Pm | Resource, sij, Mj | $\sum w_jC_j$ | MIP models and a matheuristic algorithm |

As seen from Table 1, there are a few studies considering resources ([9], [10], [11], [24], [7], [21], [2]). There is only one comparable study [2] that handle same three processing characteristics together like our paper. However, in real life, if products are not delivered on time, tardiness cost may be huge and waits are not desired. So, unlike Akyol Ozer and Sarac [2], we considered our problem as multi-objective. A two-stage mathematical model and a two-stage solution approach that can be used to solve large-scale problems are also developed. Three multi-objective techniques (goal programming, compromise programming and Lexicographic Weighted Tchebycheff programming) are applied to proposed mathematical models and the performances of these techniques are compared. To the best of our knowledge, there is no another study that deals with the same constraints and objective functions as our study in the literature. Our solution method is also original. The following sections of this
study are as follows. The proposed goal programming, compromise programming, Lexicographic Weighted Tchebycheff programming versions of integrated and two-stage mathematical models and also two stage solution approach are explained in Section 2. The computational results are presented in tables and interpreted in Section 3. Finally, conclusions are presented in Section 4.

2. The developed solution approach. The developed approach in this study is to solve the problem in two stages. It is also possible to address the problem in an integrated manner, and, as a matter of fact, the first relevant study in this regard was developed by Akyol Ozer and Sarac [2]. The objective of the model is minimization of total weighted completion time. The integrated model proposed by Akyol Ozer and Sarac [2] carries out the assignment of jobs to machines and sequencing jobs for each machine together and guarantees the optimal solution. However, the model may not find a solution in large-scale problems within a time limit due to the NP-hard nature of the problem. Also, in real life, problems often have multiple objectives. For this reason, in our study, by dividing the solution process into two stages, we developed a two-stage mathematical model and a two-stage solution approach, of course these approaches do not ensure the optimal solution, but can find a good solution in a time limit. The multi-objective integrated model (M0) and the models (M1 and M2) in proposed two-stage mathematical model are solved goal programming, compromise programming and Lexicographic Weighted Tchebycheff programming methods. These approaches are presented in the following sections.

The set of jobs and machines are denoted by \( N = \{1, 2, \ldots, n\} \) and \( L = \{1, 2, \ldots, m\} \), respectively. Subscripts \( i, j, q \in N \) refers to a job. Taking into account sequence-dependent setup times and eligibility restrictions, these models determine which job \((j \in N)\) on which machine \((l \in L)\) and in which position \((k \in N)\) in sequence to be processed. It also considers the need for non-overlapping common resources. The set of resources is defined by \( R = \{1, 2, \ldots, g\} \) and subscript \( r \in R \) refers to a resource. The parameters and decision variables are set out in Appendix 1-2.

2.1. Multi-objective integrated model (M0). The multi-objective integrated model is given below.

\[
\begin{align*}
\text{min } f_1 &= C_{\text{max}} \\
\text{min } f_2 &= \sum_{j=1}^{n} T_j \\
\text{min } f_3 &= \sum_{j=1}^{n} w_j
\end{align*}
\]

The objective functions of the M0 are set to find the optimal schedule which minimizes the makespan \((C_{\text{max}})\), total tardiness \((\sum T_j)\), and total waiting time \((\sum w_j)\). These are indicated in Eq. (1), (2) and (3), respectively.

\[
\begin{align*}
C_j + M(1 - x_{jkl}) &\geq w_j + h_j + p_j \quad \forall j, k, l \quad k = 1 \\
C_j - M(1 - x_{jkl}) &\leq w_j + h_j + p_j \quad \forall j, k, l \quad k = 1 \\
C_j + M(2 - x_{jkl} - x_{i(k-1)l}) &\geq C_i + w_j + s_{ij} + p_j \quad \forall i, j, k, l \quad i \neq j, k > 1
\end{align*}
\]
\[
C_j - M(2 - x_{jkl} - x_{i(k-1)l}) \leq C_i + s_{ij} + p_j \quad \forall i, j, k, l \ i \neq j, k > 1 \quad (7)
\]

\[
C_{\max} \geq C_j \quad \forall j \quad (8)
\]

\[
\sum_j x_{jkl} \leq 1 \quad \forall k, l \quad (9)
\]

\[
\sum_k \sum_l x_{jkl} = 1 \quad \forall j \quad (10)
\]

\[
\beta_{jl} \geq x_{jkl} \quad \forall j, k, l \quad (11)
\]

\[
\sum_j x_{jkl} - \sum_i x_{i(k-1)l} \leq 0 \quad \forall k, l \ k > 1 \quad (12)
\]

\[
a_j \geq C_i + w_j - M(2 - x_{jkl} - x_{i(k-1)l}) \quad \forall i, j, k, l \ i \neq j, k > 1 \quad (13)
\]

\[
a_j \leq C_i + w_j + M(2 - x_{jkl} - x_{i(k-1)l}) \quad \forall i, j, k, l \ i \neq j, k > 1 \quad (14)
\]

\[
a_j \leq w_j + M(1 - x_{jkl}) \quad \forall j, k, l \ k = 1 \quad (15)
\]

\[
a_j \geq w_j - M(1 - x_{jkl}) \quad \forall j, k, l \ k = 1 \quad (16)
\]

\[
a_q \leq a_j + M \ e_{1jq} \quad \forall j, q, r \ j < q, res_{jr} = 1, res_{qr} = 1 \quad (17)
\]

\[
a_j \leq C_q + M \ e_{1jq} \quad \forall j, q, r \ j < q, res_{jr} = 1, res_{qr} = 1 \quad (18)
\]

\[
a_j \leq a_q + M \ e_{2jq} \quad \forall j, q, r \ j < q, res_{jr} = 1, res_{qr} = 1 \quad (19)
\]

\[
a_q \leq C_j + M \ e_{2jq} \quad \forall j, q, r \ j < q, res_{jr} = 1, res_{qr} = 1 \quad (20)
\]

\[
C_q \leq a_j + M \ e_{3jq} \quad \forall j, q, r \ j < q, res_{jr} = 1, res_{qr} = 1 \quad (21)
\]

\[
C_j \leq a_q + M \ e_{4jq} \quad \forall j, q, r \ j < q, res_{jr} = 1, res_{qr} = 1 \quad (22)
\]

\[
e_{1jq} + e_{2jq} + e_{3jq} + e_{4jq} = 3 \quad \forall j, q, r \ j < q, res_{jr} = 1, res_{qr} = 1 \quad (23)
\]

\[
\sum_{j \mid j < q, res_{jr} = 1, res_{qr} = 1} (1 - e_{1jq}) + (1 - e_{2jq}) \leq f_r - 1 \quad \forall q, r \quad (24)
\]

\[
T_j \geq C_j - d_j \quad \forall j \quad (25)
\]

\[
x_{jkl} \in \{0,1\} \quad \forall j, k, l \quad (26)
\]

\[
e_{1jq}, e_{2jq}, e_{3jq}, e_{4jq} \in \{0,1\} \quad \forall j, q \quad (27)
\]
Constraints (4) and (5) ensure that the completion time of a job in the first sequence of any machine is equal to the sum of the waiting time, setup time and processing time of that job. Constraints (6) and (7) determine the completion times of non-first jobs. Constraint (8) ensures that $C_{\text{max}}$ value is equal to or greater than the greatest completion time. Constraints (9) and (10), respectively, ensure that at most one job is assigned to each position in sequence on each machine and each job is assigned to only one position in sequence of one machine. Constraint (11) ensures that jobs are assigned to only technically suitable machines. Constraint (12) ensures that jobs are sorted without skipping any position in sequence. Constraints (13) and (14) calculates the starting times for the jobs that are not assigned to the first position in sequence on a machine. Constraints (15) and (16) determines the starting times for the jobs that are assigned in the first position for processing. Constraints (17) - (24) prevent overlapping of the resource $r$. If job $j$ and job $q$ use resource $r$, there are four possible situations for these jobs. When $e_{1jq}$ value is equal to 0, job $q$ starts to process before job $j$ and they are overlapped. Similarly, if $e_{2jq}$ gets 0 value, job $j$ process before and job $j$ and $q$ are also overlapped. If $e_{3jq}$ or $e_{4jq}$ is zero, this means that the jobs do not overlap. Constraint (23) guarantees exactly one among four situations should be happened. Constraint (24) ensures that number of overlapping should be equal or less than number of copy resource ($f_r - 1$). For instance, if resource $r$ has one copy, number of resource $r$ is equal to two, the jobs that are used resource $r$ can be overlapped once. Tardiness of jobs is calculated by Constraint (25). Finally, constraints (26) - (32) are the sign constraints of the decision variables. To understand better of mathematical model, a toy problem that consists of three jobs (job $i$, job $j$ and job $q$) and two machines (M1 and M2) are presented. Gantt Chart of the problem is shown in Figure 1.

The job $i$ and $j$ using the same resource are shown in the same color. As seen in the figure, job $i$ and $j$ cannot be produced at the same time because their resource has no copy and job $j$ must wait for completion of job $i$. If this resource had one copy, job $j$ could start at the same time with job $i$.

### Integrated Weighted Goal Programming Model (M0-WGP)

In the goal programming model, decision makers are required to set a target level for each goal. Then, an achievement function is formulated for each goal and a solution that minimizes deviations from this achievement function is searched. An important advantage of goal programming models is that they enable a single objective function to be defined by minimizing the sum of deviation variables from target levels of these goals, although there are many goals in the decision-making process. Another advantage of goal programming is that always a feasible solution is obtained even if
no target level of goal can be achieved. The objective function of M0-WGP model is given in Eq. (33). It is weighted sum of over-achievements from target values of $C_{\text{max}}, \sum T_j, \sum w_j$, respectively. Over-achievements are normalized using ideal and anti-ideal values of each objective function.

$$\min z = \alpha_1\left(\frac{dc^+}{I_{C_{\text{max}}} - I_{C_{\max}}}\right) + \alpha_2\left(\frac{dt^+}{I_{T} - I_{t}}\right) + \alpha_3\left(\frac{dw^+}{I_{w} - I_{w}}\right)$$ (33)

Constraints of integrated weighted goal programming model can be stated as:

$$C_{\text{max}} + dc^- - dc^+ = I_{C_{\text{max}}}$$ (34)

$$\sum_{j=1}^{n} T_j + dt^- - dt^+ = I_{T}$$ (35)

$$\sum_{j=1}^{n} w_j + dw^- - dw^+ = I_{w}$$ (36)

Constraints (34), (35) and (36) attempt to achieve the prioritized targets as much as possible. The target value for each objective function is set to ideal value and right-hand sides of these constraints are equal to ideal values of each objective function. In weighted goal programming version of M0, constraints (4)-(32) should be also included.

(4)-(32)

$$dc^- , dc^+ \geq 0$$ (37)

$$dt^- , dt^+ \geq 0$$ (38)

$$dw^- , dw^+ \geq 0$$ (39)

Finally, Constraints (37), (38) and (39) are sign constraints.

**Integrated Compromise Programming Model (M0-CP)**

Compromise programming assumes that decision makers seek a solution as close as possible to the ideal point. To achieve this closeness, a distance (deviation) function $(\theta)$ is used. In this paper, in addition to deviation, sum of the normalized objective
functions, multiplied by a very small number (\( \gamma \)), is also added to objective function of integrated compromise programming model based on Augmented Tchebycheff metric. The objective function of M0-CP model is given in Eq. (40).

\[
\min z = \theta + \gamma \left( \frac{C_{\text{max}} - I_{\text{cmax}}}{AI_{\text{cmax}} - I_{\text{cmax}}} + \frac{\sum_{j=1}^{n} T_j - I^{tt}}{AI^{tt} - I^{tt}} + \frac{\sum_{j=1}^{n} w_j - I^{w}}{AI^{w} - I^{w}} \right) \tag{40}
\]

Constraints of integrated weighted compromise programming model are given below:

\[\alpha_1 \left( \frac{C_{\text{max}} - I_{\text{cmax}}}{AI_{\text{cmax}} - I_{\text{cmax}}} \right) \leq \theta \tag{41}\]
\[\alpha_2 \left( \frac{\sum_{j=1}^{n} T_j - I^{tt}}{AI^{tt} - I^{tt}} \right) \leq \theta \tag{42}\]
\[\alpha_3 \left( \frac{\sum_{j=1}^{n} w_j - I^{w}}{AI^{w} - I^{w}} \right) \leq \theta \tag{43}\]

Constraints (41), (42) and (43) are used for calculation of deviation value. In compromise programming version of M0, constraints (4) - (32) should be also included.

(4) - (32)
\[\theta \geq 0 \tag{44}\]

Finally, Constraints (44) is related with nature of decision variable \( \theta \).

### Integrated Lexicographic Weighted Tchebycheff Programming Model (M0-LWT)

To solve the multi-objective mathematical model presented in this article, the LWT method is also implemented to obtain efficient solutions to the problem. The formulation of the LWT method for three-objective mathematical model consists of four mathematical models. The constraints of all these mathematical models of M0-LWT include same constraints of integrated weighted compromise programming model. Constraints (41), (42) and (43) are also used for calculation of deviation value in this version. In lexicographic weighted Tchebycheff programming version of M0, constraints (4) - (32) should be included. Constraint (44) is sign constraint of decision variable \( \theta \) in first model of M0-LWT. The objective function of first model of M0-LWT is presented in Eq. (45).

\[
\min z = \theta \tag{45}
\]

The best value of \( \theta \) which is called \( \theta^* \) is obtained by solving first model. To keep \( \theta^* \) value, Constraint (46) are added to second, third and fourth models.

\[\theta \leq \theta^* \tag{46}\]

The objective function of second model of M0-LWT is minimization of total tardiness (2). After solving second model, Constraint (47) are added to third and fourth models to keep obtained total tardiness value \( TT^* \).

\[\sum_{j=1}^{n} T_j \leq TT^* \tag{47}\]

The objective function of third model of M0-LWT is minimization of makespan (1). After solving third model, Constraint (48) are added to fourth models to keep obtained makespan value \( C_{\text{max}}^* \).
The objective function of last model of M0-LWT is minimization of total waiting time (3).

2.2. Two stage mathematical model. The problem is addressed in two stages in the proposed approach. In the first stage, the problem of assigning jobs to machines is solved then in the second stage; the jobs assigned to each machine are sorted. In the first stage, the jobs are assigned to the machines by taking into account the machine eligibility constraints. However, at this stage, total sequence dependent setup times and completion time of the jobs cannot be calculated because sequence of the jobs is not yet known. For this reason, it is tried to assign the similar jobs to the same machines as possible and the number of jobs assigned to the machines and the workloads of machines are balanced. In fact, the success of the two-stage solution approach depends on the success of assigning jobs to machines in the first stage. In the second stage, dedicated parallel machine scheduling problem with sequence dependent setup times and common resources. The name of the model used in the first stage is M1 and second model developed for stage 2 is named as M2. M1:

\[
\min f_4 = \sum_{l=1}^{m} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} s_{ij} y_{jl} y_{il} \quad (49)
\]

\[
\min f_5 = \sum_{l=1}^{m} \sum_{j=1}^{n} (y_{jl} - I^n)^2 \quad (50)
\]

Since a long total setup time is undesirable, minimization of total all possible set up times for each machine is the first objective of M1 and it is defined in Eq. (49). Another consideration is related with assignment of jobs to the machines in a balance manner. So, second objective is minimization of deviations from ideal value of number of jobs assigned a machine and it is expressed in Eq. (50).

\[
\sum_{j=1}^{n} p_j y_{jl} \leq \text{cap} \quad \forall l \quad (51)
\]

\[
\sum_{l=1}^{m} y_{jl} = 1 \quad \forall j \quad (52)
\]

\[
y_{jl} \leq \beta_{jl} \quad \forall j, l \quad (53)
\]

\[
y_{jl} \in \{0, 1\} \quad \forall j, l \quad (54)
\]

Constraint (51) ensures that the total processing time of the jobs assigned to a machine do not exceed the representative capacity (cap) of that machine. Constraint (52) ensures that each job is assigned exactly to a machine. Constraint (53) ensures a job is assigned only to a machine on which the job can be processed. Finally, (54) is the sign constraint for \(y_{jl}\) decision variable.

M2:

As expected, objective functions of M2 are same as integrated model (M0) and these are minimization of makespan, total tardiness and total waiting time. The formulations are given in Eq. (55), (56) and (57).
\[
\begin{align*}
\min f_1 &= C_{\text{max}} \\
\min f_2 &= \sum_{j=1}^{n} T_j \\
\min f_3 &= \sum_{j=1}^{n} w_j
\end{align*}
\]

Constraints (58) and (59) ensure that the completion time of the first job assigned to any machine is equal to the sum of the processing time, setup time, and job waiting time. Constraints (60) and (61) ensure that the complete on times of jobs not assigned to the first position in sequence are equal to the sum of the preceding job completion time, processing time, the setup time between two jobs and the waiting time.

\[
C_j + M(1 - \lambda_{jk}) \geq w_j + h_j + p_j \quad \forall j, k, l \quad b_j = l, k = 1
\] (58)

\[
C_j - M(1 - \lambda_{jk}) \leq w_j + h_j + p_j \quad \forall j, k, l \quad b_j = l, k = 1
\] (59)

\[
C_j + M(2 - \lambda_{jk} - \lambda_{i(k-1)}) \geq C_i + w_j + s_{ij} + p_j \quad \forall i, j, k, l \quad b_i = l, b_j = l, i \neq j, k > 1
\] (60)

\[
C_j - M(2 - \lambda_{jk} - \lambda_{i(k-1)}) \leq C_i + w_j + s_{ij} + p_j \quad \forall i, j, k, l \quad b_i = l, b_j = l, i \neq j, k > 1
\] (61)

Constraint (62) and constraint (63) ensure that only one job is assigned to each position in the sequence on each machine and each one job should be assigned to a position in sequence. Constraint (64) ensures that \(C_{\text{max}}\) value is greater than or equal to the largest completion time. Constraint (65) ensures that jobs are sequenced without skipping any position in sequence. Constraints (66) - (69) calculates the starting times for the jobs.

\[
\sum_{j|b_j=l} \lambda_{jk} \leq 1 \quad \forall k, l
\] (62)

\[
\sum_{k} \lambda_{jk} = 1 \quad \forall j \quad b_j = l
\] (63)

\[
C_{\text{max}} \geq C_j \quad \forall j
\] (64)

\[
\sum_{j|b_j=l} \lambda_{jk} - \sum_{i|b_i=l} \lambda_{i(k-1)} \leq 0 \quad \forall k, l \quad k > 1
\] (65)

\[
a_j \geq C_i + w_j - M(2 - \lambda_{jk} - \lambda_{i(k-1)}) \quad \forall i, j, k, l \quad b_i = l, b_j = l, i \neq j, k > 1
\] (66)

\[
a_j \leq C_i + w_j + M(2 - \lambda_{jk} - \lambda_{i(k-1)}) \quad \forall i, j, k, l \quad b_i = l, b_j = l, i \neq j, k > 1
\] (67)

\[
a_j \leq w_j + M(1 - \lambda_{jk}) \quad \forall j, k, l \quad b_j = l, k = 1
\] (68)
\begin{align*}
    a_j & \geq w_j - M(1 - \lambda_{jk}) \quad \forall j, k, l \quad b_j = l, k = 1 \quad (69) \\
    \lambda_{jk} & \in \{0, 1\} \quad \forall j, k \quad (70)
\end{align*}

Overlapping constraints (17)-(24) and tardiness constraint (25) also take part in the M2 model of two-stage solution approach. Constraint (70) is the sign constraint for \( \lambda_{jk} \) decision variable. Lastly, sign constraints (27)-(32) which are given in M0 are also valid for M2.

Two-Stage Weighted Goal Programming Model (M1-M2-WGP)

In the first stage of proposed goal programming model, the objective function is expressed in Eq. (71). First part of objective function is sum of over-achievement from target value of total all possible setup time. Second part is equal to sum of over-achievement and also under-achievement from target value of number of assigned jobs to each machine. Each objective function is normalized using ideal and anti-ideal values of its objective function.

\[
    \min z_1 = \frac{\sum_{l=1}^{m} ds_l^+}{m(I^s - I^n)} + \frac{\sum_{l=1}^{m} dn_l^+ + \sum_{l=1}^{m} dn_l^-}{m(I^s - I^n)} \quad (71)
\]

\[
    \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} s_{ij}y_{jl}y_{il} + ds_l^- - ds_l^+ = I^s \quad \forall l \quad (72)
\]

\[
    \sum_{j=1}^{n} y_{jl} + dn_l^- - dn_l^+ = I^n \quad \forall l \quad (73)
\]

\[
    ds_l^+, ds_l^- \geq 0 \quad \forall l \quad (74)
\]

\[
    dn_l^-, dn_l^+ \geq 0 \quad \forall l \quad (75)
\]

Constraint (72) and (73) are soft constraints formulated, respectively, for target values of total sequence-dependent setup time and number of assigned jobs to a machine. The target values are equal to ideal values of its objective function as in the integrated weighted goal programming model. In goal programming version of M1, constraints (51) - (54) should be also included. Finally, (74) and (75) are the sign constraints.

M2-WGP

In the second stage, objective is to find optimal sequence of jobs assigned to a particular machine for minimization over-achievement of the makespan, total tardiness and total waiting time. Normalized M2-WGP formulation as in M0-WGP model is represented in Eq. (76). It is weighted sum of over-achievements from target values of \( C_{max} \), \( \sum T_j \) and \( \sum w_j \), respectively.

\[
    \min z_2 = \alpha_1 \left( \frac{dc^+}{AI^{C_{max}} - I^{C_{max}}} \right) + \alpha_2 \left( \frac{dt^+}{AI^t - I^t} \right) + \alpha_3 \left( \frac{dw^+}{AI^w - I^w} \right) \quad (76)
\]

Constraints of weighted goal programming version of M2 model are as follows:
\begin{align*}
C_{\text{max}} + dc^{-} - dc^{+} &= I_{c}^{\text{max}} \quad (77) \\
\sum_{j=1}^{n} T_j + dt^{-} - dt^{+} &= I_{tt} \quad (78) \\
\sum_{j=1}^{n} w_j + dw^{-} - dw^{+} &= I_{w} \quad (79)
\end{align*}

(17) - (25), (27) - (32), (58) - (70)

In goal programming version of M2, constraints (17) - (25), (27) - (32) and (58) - (70) should be also included. Finally, (80) - (82) are the sign constraints.

\begin{align*}
dc^{-}, dc^{+} &\geq 0 \quad (80) \\
dt^{-}, dt^{+} &\geq 0 \quad (81) \\
dw^{-}, dw^{+} &\geq 0 \quad (82)
\end{align*}

**Two-Stage Compromise Programming Model (M1-M2-CP)**

In the first stage of M1-M2-CP, compromise programming technique is also applied to M1 model. In addition to \( \theta_1 \), normalized objectives are also included in objective function based on Augmented Chebyshev metric. The objective function of M1-CP model is represented in Eq. (83).

\[
\min \ z_1 = \theta_1 + \gamma \left( \frac{\sum_{i=1}^{m} \sum_{j=i+1}^{n} s_{ij} y_{jl} y_{il} - I^s}{m(AI^s - I^s)} + \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} (y_{jl} - I^n)^2}{m(AI^n - I^n)} \right) \quad (83)
\]

Constraints of compromise programming version of M1 model can be stated as:

\[
\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} s_{ij} y_{jl} y_{il} - I^s \leq \theta_1 \forall l \quad (84)
\]

\[
\sum_{j=1}^{n} y_{jl} - I^n \leq \theta_1 \forall l \quad (85)
\]

Constraints (84) and (85) are used for calculation of deviation value for each objective function. In compromise programming version of M1, constraints (51)-(54) should be also included. Lastly, sign constraint of \( \theta_1 \) which is represented in constraints (86) should be added to M1-CP.

\[
(51)-(54) \quad \theta_1 \geq 0 \quad (86)
\]

**M2-CP**

As expected, normalized objective functions of M2-CP are same as compromise version of integrated model (M0-CP). First part of objective function is equal to minimization of maximum deviation (\( \theta_2 \)) among the individual deviations from objective functions. The second part is defined as sum of normalized objective
function which is multiplied by a small positive constant ($\gamma$) based on Augmented Tchebycheff metric.

$$\min z_2 = \theta_2 + \gamma \left( \frac{C_{\text{max}} - I_{c_{\text{max}}}}{AI_{c_{\text{max}}}} - \frac{C_{\text{max}} - I_{c_{\text{max}}}}{AI_{c_{\text{max}}}} + \sum_{j=1}^{n} T_j - I_{tt} + \sum_{j=1}^{n} w_j - I_{w} \right)$$  \hspace{1cm} (87)

Constraints of M2-CP model are given below:

$$\alpha_1 \left( \frac{C_{\text{max}} - I_{c_{\text{max}}}}{AI_{c_{\text{max}}}} - \frac{C_{\text{max}} - I_{c_{\text{max}}}}{AI_{c_{\text{max}}}} \right) \leq \theta_2$$  \hspace{1cm} (88)

$$\alpha_2 \left( \frac{\sum_{j=1}^{n} T_j - I_{tt}}{AI_{tt}} - I_{tt} \right) \leq \theta_2$$  \hspace{1cm} (89)

$$\alpha_3 \left( \frac{\sum_{j=1}^{n} w_j - I_{w}}{AI_{w}} - I_{w} \right) \leq \theta_2$$  \hspace{1cm} (90)

Constraints (88), (89) and (90) calculates individual deviations for each objective function associated with its weight. Constraints (58) -(70), (17) -(25), (27) -(32) and (91) are also included in all four models of M2-LWT for calculations of starting and completion time, ordering of jobs, prevention of overlapping and declaration of nature of decision variables.

$$\theta_2 \geq 0$$  \hspace{1cm} (91)

Finally, Constraint (91) is sign constraint of decision variable $\theta_2$.

**Two-Stage Lexicographic Weighted Tchebycheff Model (M1-M2-LWT)**

Mathematical model of the first stage of M1-M2-LWT is same with M1-CP. The second stage (M2-LWT) consist of four mathematical models as M0-LWT. The constraints of all these mathematical models include Constraints (88), (89) and (90) which calculate individual deviations for each objective function associated with its weight. Constraints (58) -(70), (17) -(25), (27) -(32) and (91) are also included in all four models of M2-LWT for calculations of starting and completion time, ordering of jobs, prevention of overlapping and declaration of nature of decision variables.

The objective function of first model of M2-LWT is presented in Equations (92).

$$\min z = \theta_2$$  \hspace{1cm} (92)

The best value of $\theta_2$ which is called $\theta_2^*$ is obtained by solving first model. To keep value $\theta_2^*$, Constraint (93) are added to second, third and fourth models.

$$\theta_2 \leq \theta_2^*$$  \hspace{1cm} (93)

The objective function of second model of M2-LWT is minimization of total tardiness (2). After solving second model, Constraint (47) are added to third and fourth models to keep obtained total tardiness value($TT^*$). The objective function of third model of M2-LWT is minimization of makespan (1). After solving third model, Constraint (48) are added to fourth models to keep obtained makespan value ($C_{\text{max}}^*$). Finally, the objective function of last model of M2-LWT is minimization of total waiting time (3).
2.3. **Two stage solution approach (SA-M3).** Proposed solution approach for considered problem has two stages. In the first stage, considered parallel machine scheduling problem without resource constraints is solved by simulated annealing algorithm. The objective is minimization of total tardiness. In the second stage, the model determines which jobs should wait for common resources in case of overlapping resources. The stages of proposed heuristic solution approach are explained in detail below:

*First stage*

In the first stage, jobs are assigned to machines and scheduled in these machines by using proposed simulated annealing algorithm (SA). Initial solutions are generated randomly considering machine eligibility restrictions.

Given a current solution \( s \), SA proceeds generating at each iteration a neighbor solution \( s' \) with a swap move. If \( s' \) improves total tardiness of \( s \), then the current solution is updated. The algorithm accepts some worsening solutions to avoid getting stuck in a local optimum solution. It is provided by a probability of allowance in relation with a temperature \( T \). \( \Delta \) is the difference in the total tardiness between \( s' \) and \( s \). A random number is generated in range \([0,1]\). If it is lower than \( e^{-\Delta/T} \), the worse solution is accepted. The algorithm runs until stopping criteria satisfied.

*Representation Scheme and Generation of a Solution*

Representation scheme is the way of representing a candidate solution. Representation scheme in proposed solution approach is very similar to permutation encoding. The length of array is \((n + m - 1)\). Each element in the array takes values from “0” to “n”. The \( n \) elements of the array represent jobs. The value of \( m - 1 \) element is equal to “0”. Elements with a value of “0”, means that the current machine part has finished and the next machine part has started. The order of jobs on the part of the array that corresponds to each machine specifies the order of job on that machine. For example, the following is an example solution for the 6-job, 3-machine problem:

\[
\begin{array}{cccccccc}
4 & 3 & 5 & 0 & 1 & 0 & 6 & 2 \\
\end{array}
\]

This representation scheme means that jobs 4, 3 and 5 are assigned to machine 1 in this sequence, job 1 is assigned to machine 2, and jobs 6 and 2 are assigned to machine 3.

The initial solution is generated randomly taking into account machine eligibility constraints. It is not allowed to deteriorate feasibility while neighboring solutions are generated. Swap operator is used to produce neighboring solutions considering machine eligibility restrictions.

*Cooling Schedule and Stopping Criteria*

Cooling schedule defines the way in which the temperature is going to be decreased. One common cooling schedule, and used in this approach is the geometric schedule:

\[ T_k = \alpha T_{k-1} \]

where \( T_k \) is the value of the temperature at iteration \( k \) and \( \alpha \) is the cooling speed parameter. In this type of schedule, \( \alpha \) must be smaller but close to 1. The most typical values of \( \alpha \) are between 0.8 and 0.99; smaller values can result in excessively fast cooling.

The stopping criteria of proposed two stage solution approach is to reach a certain predefined temperature value.

*Second stage*

In the first stage, machine assignment of jobs and their order in these machines are determined. Starting time, completion time and waiting time of the jobs are
calculated considering resource constraints in the second stage for minimization of total tardiness, makespan and total waiting time. So, a sub-model (M3) is proposed to decide which job will wait for common resource in case of job overlapping. Objective function of M3 are same as M2-WGP and its formulation is given in Eq. (76). Constraints of M3 are given below:

\[ C_j = w_j + h_j + p_j \quad \forall j, k, l \quad \delta_{jkl} = 1, k = 1 \quad (94) \]

\[ C_j = C_i + w_j + s_{ij} + p_j \quad \forall i, j, k, l \quad i \neq j, k > 1, \delta_{jkl} = 1, \delta_{i(k-1)l} = 1 \quad (95) \]

Constraints (94) and (95) were formed for calculation of the completion time of the jobs.

\[ a_j = C_i + w_j \quad \forall i, j, k, l \quad i \neq j, k > 1, \delta_{jkl} = 1, \delta_{i(k-1)l} = 1 \quad (96) \]

\[ a_j = w_j \quad \forall j, k, l \quad \delta_{jkl} = 1, k = 1 \quad (97) \]

Constraints (96) and (97) calculates the starting times for the jobs. In M3, constraints (17)- (25), (64), (77)- (79) should be included. Lastly, sign constraints (27)- (32) which are given in M0 are also valid for M3.

3. **Computational results.** Test problems formulated by Akyol Ozer and Sarac [2] are used to test the developed solution approaches in this study. In terms of the number of jobs, these test problems are classified into three groups according to number of jobs (n): 8, 40 and 100. Each group has problems with different characteristics according to various parameters such as number of mold types, processing time and setup time. The rest of parameters are generated as follows:

- **Representative capacity of each machine (\(cap\)):** In the first stage of the two-stage approach, \(cap\) parameter is used and calculated as the following formula:

\[ cap = \psi \sum_{j=1}^{n} P_j \]

To determine the value of \(\psi\), six problems (S1-1, S9-1, M1-1, M9-1, L1-1 and L9-1) are solved with different \(\psi\) values (0.95,1.05,1.15,1.25,1.35,1.45 and 1.55). Mean of \(z\) values for each \(\psi\) are obtained 0.33, 0.22, 0.32, 0.34, 0.32, 0.31, 0.33, respectively. So, the value of \(\psi\) is chosen as 1.05 since it has the smallest mean.

- **Weights of objectives (\(\alpha_1, \alpha_2, \alpha_3\)):** Weights of makespan, total tardiness and total waiting time are set as 0.2, 0.6 and 0.2, respectively according to expert opinion in plastic injection factory.

- **Ideal values of objectives (\(I^{C_{max}}, I^{tt}, I^{w}, I^{s}, I^{n}\)):** Ideal values are used for target values in goal programming models, reference points in compromise programming models and normalization of objective functions. They are calculated as follows:

\[ I^{C_{max}} = cap \]

\[ I^{tt} = 0 \]

\[ I^{w} = 0 \]

\[ I^{s} = \frac{\sum_{i}^{n} \sum_{j}^{n} s_{ij}}{2nm} \]

\[ I^{n} = \frac{n}{m} \]
○ Anti-ideal values of objectives ($AI_{C_{\text{max}}}$, $AI_{tt}$, $AI_{w}$, $AI_{s}$, $AI_{n}$): Anti-ideal values are used for normalization of objective functions. It is assumed that all jobs are assigned to a single machine for calculating anti-ideal values. The jobs are scheduled in descending order of their due dates to determine $AI_{tt}$ value. Formulations for the rest of anti-ideal values are represented as follows:

\[
AI_{C_{\text{max}}} = \sum_{i}^{n} \text{max}_{j}s_{ij} + \sum_{j}^{n} p_{j}
\]

\[
AI_{w} = \sum_{r}^{g} \left( \sum_{j}^{n} p_{j|\text{res}_{j,r} = 1} - \min_{j} p_{j|\text{res}_{j,r} = 1} \right)
\]

\[
AI_{s} = \sum_{i}^{n} \text{max}_{j}s_{ij}
\]

\[
AI_{n} = n
\]

○ A small positive constant ($\gamma$): $\gamma$ value is used in Augmented Chebyshev metric for compromise programming version of models. In the literature, it is generally assumed that $\gamma$ value is equal to 0.006. Therefore, it is taken as 0.006.

○ The parameters of simulated annealing algorithm are as follows: initial temperature = 100, cooling speed = 0.99, number of neighbors in each temperature = 100 and termination temperature = 0.1.

For the solution of the test problems, an Intel (R) Xeon (R) (2.67 Ghz, 16 GB Ram) computer is used. Also, GAMS DICOPT and GAMS CPLEX solvers are used, respectively, in the first stage (M1) and in the second stage (M2) of the approach. According to experts’ opinion, the maximum time they can allocate for scheduling is 8000. Therefore, solution time for all the tests is limited to 8000 seconds.

3.1. Test results. We solved different scale test problems with goal programming, compromise programming and lexicographic weighted Tchebycheff programming versions of integrated models and two-stage mathematical models. In addition to mathematical models, two stage solution approach is examined with different scale test problems. The results are given in Tables 2 to 7. These tables are composed of eight sections. The first section consists of a row and shows the number of the test problem. The solution reports of M0-WGP, M0-CP, M0-LWT, M1-M2-WGP, M1-M2-CP, M1-M2-LWT and SA-M3 are given in the rest of the sections, respectively. The solution report of each section for small-scale test problems consists of five rows: objective function values of makespan ($f_{1}$), total tardiness ($f_{2}$), total waiting time ($f_{3}$), inconvenience value ($z$) and solution time in seconds ($t$). In the tables of medium-scale and large-scale test problems, solution time column is not included in M0-WGP, M0-CP, M0-LWT, M1-M2-WGP, M1-M2-CP, M1-M2-LWT since running time limit (8000 sec.) is exceed for all test problems. Inconvenience value can be described as weighted normalized distance from ideal points and it takes values between 0 and 1. When the obtained solution is getting close to ideal point, $z$ value is getting close to 0. The $z$ value is calculated by Eq. (98):

\[
\min z = \alpha_{1} \frac{C_{\text{max}} - I_{C_{\text{max}}}}{AI_{C_{\text{max}}} - I_{C_{\text{max}}}} + \alpha_{2} \frac{\sum_{j=1}^{n} T_{j} - I_{tt}}{AI_{tt} - I_{tt}} + \alpha_{3} \frac{\sum_{j=1}^{n} w_{j} - I_{w}}{AI_{w} - I_{w}}
\]

The obtained best ($z$) value for each test problem is signed as bold in all the tables in this section.
Table 2. Results of Small-scale ($n=8, m=2$) Problems

| Test No | 1-1 | 2-1 | 3-1 | 4-1 | 5-1 | 6-1 | 7-1 | 8-1 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| $f_1$   | 384 | 273 | 326 | 394 | 346 | 335 | 318 | 280 |
| $f_2$   | 299 | 99  | 296 | 564 | 116 | 295 | 232 | 102 |
| $f_3$   | 4   | 7   | 3   | 0   | 0   | 0   | 0   | 0   |
| $z$     | 0.1 | 0.07| 0.09| 0.24| 0.07| 0.08| 0.11| 0.05|
| $t$ (sec.) | 196 | 181 | 299 | 880 | 153 | 175 | 82  | 95  |

Solutions of small-scale test problems with ($m=2$) and ($m=3$) are given in Table 2 and Table 3, respectively. As seen these tables, best $z$ values are achieved by integrated models (M0-WGP, M0-CP and M0-LWT) for small-scale test problems. However, M1-M2-WGP, M1-M2-CP, M1-M2-LWT and SA-M3 are more advantageous in terms of solution time. Furthermore, it is noteworthy that M0-WGP has achieved better $z$ values in all problems.

In Tables 4 and 5, results of medium-scale test problems with ($m=2$) and ($m=6$) are presented, respectively. As can be seen in these tables, M0-WGP could not find a feasible solution for three of test problems (7-1, 12-1, 15-1) in time limit. Best ($z$) values are obtained by SA-M3 for all medium-scale test problems with

---

| Test No | 1-1 | 2-1 | 3-1 | 4-1 | 5-1 | 6-1 | 7-1 | 8-1 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| $f_1$   | 423 | 313 | 345 | 433 | 331 | 335 | 310 | 266 |
| $f_2$   | 465 | 130 | 333 | 796 | 291 | 295 | 485 | 141 |
| $f_3$   | 4   | 7   | 3   | 87  | 0   | 0   | 0   | 0   |
| $z$     | 0.14| 0.09| 0.10| 0.42| 0.11| 0.08| 0.19| 0.05|
| $t$ (sec.) | 1   | 1   | 1   | 4   | 1   | 1   | 1   | 1   |

| Test No | 1-1 | 2-1 | 3-1 | 4-1 | 5-1 | 6-1 | 7-1 | 8-1 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| $f_1$   | 450 | 320 | 345 | 393 | 390 | 335 | 350 | 351 |
| $f_2$   | 636 | 144 | 333 | 701 | 237 | 295 | 479 | 208 |
| $f_3$   | 0   | 7   | 0   | 167 | 0   | 0   | 0   | 13  |
| $z$     | 0.19| 0.1 | 0.1 | 0.42| 0.11| 0.08| 0.19| 0.1  |
| $t$ (sec.) | 2   | 1   | 0   | 2   | 1   | 1   | 1   | 1   |

| Test No | 1-1 | 2-1 | 3-1 | 4-1 | 5-1 | 6-1 | 7-1 | 8-1 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| $f_1$   | 450 | 320 | 345 | 393 | 390 | 335 | 350 | 351 |
| $f_2$   | 636 | 144 | 333 | 701 | 237 | 295 | 479 | 208 |
| $f_3$   | 0   | 7   | 0   | 167 | 0   | 0   | 0   | 13  |
| $z$     | 0.19| 0.1 | 0.1 | 0.42| 0.12| 0.08| 0.19| 0.1  |
| $t$ (sec.) | 8   | 3   | 6   | 14  | 4   | 3   | 5   | 3   |

| Test No | 1-1 | 2-1 | 3-1 | 4-1 | 5-1 | 6-1 | 7-1 | 8-1 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| $f_1$   | 473 | 320 | 411 | 366 | 402 | 383 | 328 | 295 |
| $f_2$   | 456 | 82  | 349 | 717 | 335 | 258 | 222 | 99  |
| $f_3$   | 90  | 16  | 37  | 112 | 127 | 0   | 16  | 0   |
| $z$     | 0.19| 0.08| 0.13| 0.37| 0.25| 0.08| 0.15| 0.05|
| $t$ (sec.) | 13  | 12  | 13  | 12  | 12  | 12  | 13  | 13  |
Table 3. Results of Small-scale \((n=8, m=3)\) Problems

| Test No | 9-1 | 10-1 | 11-1 | 12-1 | 13-1 | 14-1 | 15-1 | 16-1 |
|---------|-----|------|------|------|------|------|------|------|
| M0-WGP  |     |      |      |      |      |      |      |      |
| \(f_1\) | 296 | 331  | 391  | 239  | 154  | 283  | 183  | 262  |
| \(f_2\) | 311 | 325  | 393  | 135  | 191  | 331  | 124  | 117  |
| \(f_3\) | 2   | 0    | 2    | 0    | 0    | 0    | 0    | 0    |
| \(z\)   | 0.08| 0.13 | 0.14 | 0.06 | 0.06 | 0.08 | 0.07 | 0.04 |
| \(t(\text{sec.})\) | 618 | 2211 | 1774 | 337  | 415  | 497  | 439  | 246  |
| M0-CP   |     |      |      |      |      |      |      |      |
| \(f_1\) | 296 | 331  | 370  | 198  | 154  | 283  | 143  | 262  |
| \(f_2\) | 311 | 325  | 326  | 122  | 191  | 331  | 94   | 117  |
| \(f_3\) | 2   | 0    | 70   | 0    | 0    | 17   | 0    | 0    |
| \(z\)   | 0.08| 0.13 | 0.15 | 0.06 | 0.06 | 0.08 | 0.08 | 0.04 |
| \(t(\text{sec.})\) | 691 | 1227 | 1414 | 401  | 403  | 539  | 252  | 169  |
| M0-LWT  |     |      |      |      |      |      |      |      |
| \(f_1\) | 341 | 492  | 409  | 286  | 223  | 282  | 243  | 295  |
| \(f_2\) | 495 | 870  | 428  | 344  | 431  | 500  | 295  | 342  |
| \(f_3\) | 14  | 327  | 155  | 8    | 0    | 0    | 0    | 0    |
| \(z\)   | 0.12| 0.58 | 0.23 | 0.13 | 0.14 | 0.12 | 0.15 | 0.09 |
| \(t(\text{sec.})\) | 1045| 1231 | 1980 | 354  | 402  | 876  | 298  | 108  |
| M1-M2-WGP|    |      |      |      |      |      |      |      |
| \(f_1\) | 334 | 411  | 470  | 297  | 230  | 345  | 189  | 530  |
| \(f_2\) | 488 | 791  | 567  | 263  | 430  | 600  | 273  | 1073 |
| \(f_3\) | 14  | 274  | 305  | 89   | 0    | 7    | 21   | 0    |
| \(z\)   | 0.11| 0.5  | 0.34 | 0.16 | 0.14 | 0.15 | 0.16 | 0.26 |
| \(t(\text{sec.})\) | 1   | 2    | 0    | 1    | 1    | 1    | 1    | 1    |
| M1-M2-CP|    |      |      |      |      |      |      |      |
| \(f_1\) | 341 | 411  | 470  | 297  | 230  | 338  | 189  | 295  |
| \(f_2\) | 495 | 791  | 567  | 263  | 430  | 600  | 273  | 344  |
| \(f_3\) | 14  | 274  | 305  | 89   | 0    | 0    | 21   | 0    |
| \(z\)   | 0.12| 0.5  | 0.34 | 0.16 | 0.14 | 0.15 | 0.16 | 0.09 |
| \(t(\text{sec.})\) | 2   | 6    | 3    | 3    | 2    | 2    | 2    | 1    |
| M1-M2-LWT|   |      |      |      |      |      |      |      |
| \(f_1\) | 369 | 416  | 469  | 253  | 185  | 336  | 143  | 309  |
| \(f_2\) | 317 | 571  | 639  | 128  | 186  | 407  | 94   | 116  |
| \(f_3\) | 50  | 282  | 304  | 80   | 44   | 85   | 17   | 0    |
| \(z\)   | 0.11| 0.46 | 0.36 | 0.11 | 0.16 | 0.18 | 0.08 | 0.05 |
| \(t(\text{sec.})\) | 12  | 14   | 12   | 12   | 14   | 14   | 13   |      |

\((m = 2)\). On the contrary, M1-M2-WGP are achieved best \((z)\) values the rest of test problems.

Solutions of large-scale test problems with \((m = 2)\) and \((m = 8)\) are given in Table 6 and Table 7, respectively. As seen these tables, none of the test problems could not solve by M0-WGP and M0-CP in time limit. However, M0-LWT could find a feasible solution for four test problems. Best \((z)\) values are obtained by SA-M3 for all problems. SA-M3 also indicates significantly high performance compared to other approaches in terms of solution times. The comparison of the results reveals which method shows best performance in terms of \((z)\) value. However, only the value of \((z)\) cannot be sufficient to indicate which method has exceptional performance. For
### Table 4. Results of Medium-scale \((n=40, m=2)\) Problems

| Test No | 1-1  | 2-1  | 3-1  | 4-1  | 5-1  | 6-1  | 7-1  | 8-1  |
|---------|------|------|------|------|------|------|------|------|
| M0-GP   |      |      |      |      |      |      |      |      |
| \(f_1\) | 2365 | 2135 | 1928 | 2115 | 1574 | 1805 | -    | -    |
| \(f_2\) | 22190| 20719| 15318| 19944| 4590 | 8989 | -    | -    |
| \(f_3\) | 396  | 94   | 130  | 124  | 0    | 0    |      |      |
| \(z\)   | 0.27 | 0.23 | 0.19 | 0.28 | 0.07 | 0.11 | -    | -    |
| M0-CP   |      |      |      |      |      |      |      |      |
| \(f_1\) | 2358 | 2274 | 1965 | 1625 | 1491 | 1772 | 1656 | 1821 |
| \(f_2\) | 18290| 17933| 15682| 14654| 4590 | 8989 | -    | -    |
| \(f_3\) | 280  | 416  | 364  | 301  | 0    | 0    |      |      |
| \(z\)   | 0.23 | 0.25 | 0.22 | 0.22 | 0.07 | 0.08 | 0.10 | 0.12 |
| M0-LWT  |      |      |      |      |      |      |      |      |
| \(f_1\) | 5776 | 8558 | 7284 | 2547 | 2764 | 5648 | 2013 | 4304 |
| \(f_2\) | 21808| 33944| 26740| 56966| 7363 | 16720| 19845| 14963|
| \(f_3\) | 577  | 1458 | 657  | 1070 | 91   | 240  |      |      |
| \(z\)   | 0.43 | 0.74 | 0.56 | 0.80 | 0.16 | 0.37 | 0.22 | 0.33 |
| M1-M2-GWP|     |      |      |      |      |      |      |      |
| \(f_1\) | 2077 | 2922 | 1779 | 1771 | 1447 | 1789 | 1747 | 1463 |
| \(f_2\) | 21191| 19565| 17681| 16175| 4779 | 5232 | 5191 | 3661 |
| \(f_3\) | 76   | 178  | 4    | 198  | 0    | 0    | 0    | 12   |
| \(z\)   | 0.22 | 0.23 | 0.19 | 0.23 | 0.06 | 0.16 | 0.07 | 0.14 |
| M1-M2-CP|     |      |      |      |      |      |      |      |
| \(f_1\) | 1979 | 1992 | 1865 | 1546 | 1467 | 1635 | 1440 | 1663 |
| \(f_2\) | 15322| 20602| 14408| 13934| 3348 | 6396 | 4255 | 7751 |
| \(f_3\) | 467  | 128  | 185  | 306  | 2    | 0    | 0    | 186  |
| \(z\)   | 0.21 | 0.23 | 0.18 | 0.21 | 0.05 | 0.05 | 0.05 | 0.12 |
| M1-M2-LWT|    |      |      |      |      |      |      |      |
| \(f_1\) | 5232 | 5076 | 6301 | 5395 | 2624 | 3838 | 3360 | 2862 |
| \(f_2\) | 19227| 18025| 22470| 18168| 6768 | 11926| 10643| 8291 |
| \(f_3\) | 593  | 489  | 349  | 382  | 83   | 13   | 0    | 41   |
| \(z\)   | 0.38 | 0.37 | 0.45 | 0.43 | 0.14 | 0.22 | 0.19 | 0.15 |
| SA-M3   |      |      |      |      |      |      |      |      |
| \(f_1\) | 1819 | 1561 | 1623 | 1346 | 1473 | 1621 | 1399 | 1776 |
| \(f_2\) | 8153 | 5314 | 3941 | 4380 | 2699 | 1347 | 2687 | 3086 |
| \(f_3\) | 479  | 82   | 137  | 290  | 0    | 0    | 0    | 357  |
| \(z\)   | 0.14 | 0.07 | 0.07 | 0.1  | 0.04 | 0.03 | 0.04 | 0.08 |
| \(t\)(sec.) | 36  | 32  | 38  | 40  | 30  | 31  | 36  | 36  |

For example, in Table 3.1, the best \((z)\) value \((0.11)\) for 7-1 is obtained with M0-WGP. For the same problem, the \((z)\) value is obtained \(0.15\) with SA-M3. If only the value of \((z)\) is considered, it can be said that MO-WGP is more successful than SA-M3 for this problem. However, when these solutions are compared with each other in terms of dominance, it will be seen that they do not dominate each other. In other words, the success of these two methods is the same according to dominance. Therefore, in addition to \((z)\) values, the number of non-dominant solutions obtained by each method should be examined.

In Figure 2, the non-dominant solution numbers obtained by M0-WGP, M0-CP, M0-LWT, M1-M2-WGP, M1-M2-CP, M1-M2-LWT and SA-M3 methods are given for small, medium and large size problems, respectively. As can be seen from Figure 2, non-dominant solutions for all 16 problems were obtained with all three of the integrated models scalarized with GP, CP and LWT for small problems. In almost all of the medium and large scale problems, non-dominant solutions were obtained by SA-M3 method.

In summary, when both the success of \((z)\) values and dominance are evaluated together, the success of integrated models stands out for small scale test problems. Especially, goal programming version of integrated model has achieved better \(z\) values.
### Table 5. Results of Medium-scale ($n=40$, $m=6$) Problems

| Test No | 9-1 | 10-1 | 11-1 | 12-1 | 13-1 | 14-1 | 15-1 | 16-1 |
|---------|-----|------|------|------|------|------|------|------|
| $f_1$   | 1539| 1896 | 1515 | -    | 645  | 709  | -    | 1336 |
| $f_2$   | 17499| 25024| 19640| -    | 3198 | 4627 | -    | 14430|
| $f_3$   | 1092 | 1396 | 3426 | -    | 0    | 3    | -    | 1094 |
| $z$     | 0.27 | 0.43 | 0.51 | -    | 0.03 | 0.05 | -    | 0.29 |

**M0-WGP**

| $f_1$   | 1487 | 2179 | 1515 | 1798 | 756  | 784  | 815  | 1219 |
| $f_2$   | 17499| 25024| 19640| 20895| 3262 | 3674 | 14430| 6330 |
| $f_3$   | 1530 | 2531 | 3777 | 1862 | 66   | 7    | 94   | 869  |
| $z$     | 0.31 | 0.55 | 0.54 | 0.43 | 0.04 | 0.05 | 0.08 | 0.23 |

**M0-CP**

| $f_1$   | 4093 | 6059 | 10466| 16743| 1002 | 1947 | 1726 | 3677 |
| $f_2$   | 17052| 23743| 44256| 70611| 3124 | 7355 | 6522 | 14022|
| $f_3$   | 1434 | 2531 | 3777 | 1862 | 66   | 7    | 94   | 869  |
| $z$     | 0.38 | 0.62 | 1.0  | 1.0  | 0.06 | 0.16 | 0.12 | 0.33 |

**M0-LWT**

| $f_1$   | 1261 | 1608 | 1205 | 1136 | 598  | 578  | 528  | 633  |
| $f_2$   | 11303| 15041| 12840| 10871| 2202 | 4029 | 2336 | 3878 |
| $f_3$   | 1350 | 2117 | 1138 | 1645 | 0    | 3    | 2    | 6    |
| $z$     | 0.23 | 0.41 | 0.23 | 0.29 | 0.02 | 0.04 | 0.02 | 0.04 |

**M1-M2-WGP**

| $f_1$   | 1278 | 1674 | 1217 | 1069 | 627  | 619  | 531  | 712  |
| $f_2$   | 11048| 16595| 12358| 10871| 2202 | 4029 | 2336 | 3878 |
| $f_3$   | 1486 | 2512 | 1140 | 1795 | 92   | 50   | 5    | 190  |
| $z$     | 0.25 | 0.47 | 0.23 | 0.3  | 0.04 | 0.04 | 0.02 | 0.06 |

**M1-M2-CP**

| $f_1$   | 4806 | 8617 | 4168 | 5018 | 841  | 906  | 727  | 1253 |
| $f_2$   | 20346| 34350| 16584| 20148| 2316 | 2578 | 1798 | 3606 |
| $f_3$   | 1711 | 2793 | 1318 | 1619 | 11   | 10   | 56   | 79   |
| $z$     | 0.46 | 0.91 | 0.39 | 0.51 | 0.03 | 0.04 | 0.04 | 0.07 |

**M1-M2-LWT**

| $f_1$   | 1249 | 1807 | 1047 | 977  | 876  | 651  | 639  | 738  |
| $f_2$   | 9711 | 14617| 6886 | 5216 | 2845 | 1635 | 1121 | 1539 |
| $f_3$   | 1941 | 2752 | 1318 | 1327 | 315  | 28   | 39   | 457  |
| $z$     | 0.27 | 0.48 | 0.20 | 0.21 | 0.08 | 0.03 | 0.03 | 0.08 |

**SA-M3**

| $f_1$   | 231  | 232  | 198  | 235  | 34   | 33   | 37   | 38   |

**Figure 2. Number of Non-Dominated Solutions**

values than other approaches. In addition to this, Lexicographic Weighted Tchebycheff versions of integrated and two stage mathematical model have shown poor performance for almost all medium and large-scale test problems. Since integrated models are not adequate to find feasible solutions as problem size increase we can suggest two-stage mathematical model and two-stage solution approach for these test
problems. Especially, two-stage solution approach (SA-M3) is capable of obtaining non-dominated solutions of almost all problems within a reasonable amount of CPU time (maximum 277 seconds). SA-M3 is very promising, particularly, for large scale problems.

4. Conclusion. In this study, a mathematical model was developed for the problem of scheduling of plastic injection machines. This problem is known as a multi-objective parallel machine scheduling problem with sequence dependent setup times, eligibility constraints and common resources in scheduling literature. Objectives functions are determined as the minimization of the makespan, total tardiness, and total waiting time in accordance with the nature of the problem. Goal programming, compromise programming and Lexicographic Weighted Tchebycheff programming techniques were used for solving this multi-objective problem. However, because of the complexity of the problem, there were no feasible solution for large-scale problems. So, a two-stage mathematical model and a two-stage solution approach were developed for solving industrial-scale problems. In the first stage of mathematical model, each job is assigned to a machine under machine eligibility constraint by using a multi-objective integer programming model, and in the second stage a dedicated parallel machine scheduling problem with sequence dependent setup times and common resources is solved by using another multi-objective integer programming model. Goal programming, compromise programming and Lexicographic Weighted Tchebycheff

| Test No | 1-1 | 2-1 | 3-1 | 4-1 | 5-1 | 6-1 | 7-1 | 8-1 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| f1      | -   | -   | -   | -   | -   | -   | -   | -   |
| f2      | -   | -   | -   | -   | -   | -   | -   | -   |
| f3      | -   | -   | -   | -   | -   | -   | -   | -   |
| z       | -   | -   | -   | -   | -   | -   | -   | -   |
| M0-WGP  | f1  | 5499| 7943| -   | -   | 5224| 5683| -   |
| f2      | 3718| 10  | 3946| 679 | 5073| 5495| 5430| 5873|
| f3      | 1299| 5206| -   | -   | 942 | 1450| -   | -   |
| z       | 0.71| 0.89| -   | -   | 0.72| 0.76| -   | -   |

| Test No | 1-1 | 2-1 | 3-1 | 4-1 | 5-1 | 6-1 | 7-1 | 8-1 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| f1      | -   | -   | -   | -   | -   | -   | -   | -   |
| f2      | -   | -   | -   | -   | -   | -   | -   | -   |
| f3      | -   | -   | -   | -   | -   | -   | -   | -   |
| z       | -   | -   | -   | -   | -   | -   | -   | -   |
| M0-CP   | f1  | 5499| 7943| -   | -   | 5224| 5683| -   |
| f2      | 3718| 10  | 3946| 679 | 5073| 5495| 5430| 5873|
| f3      | 1299| 5206| -   | -   | 942 | 1450| -   | -   |
| z       | 0.71| 0.89| -   | -   | 0.72| 0.76| -   | -   |
| M0-LWT  | f1  | 5499| 7943| -   | -   | 5224| 5683| -   |
| f2      | 3718| 10  | 3946| 679 | 5073| 5495| 5430| 5873|
| f3      | 1299| 5206| -   | -   | 942 | 1450| -   | -   |
| z       | 0.71| 0.89| -   | -   | 0.72| 0.76| -   | -   |

| Test No | 1-1 | 2-1 | 3-1 | 4-1 | 5-1 | 6-1 | 7-1 | 8-1 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| f1      | -   | -   | -   | -   | -   | -   | -   | -   |
| f2      | -   | -   | -   | -   | -   | -   | -   | -   |
| f3      | -   | -   | -   | -   | -   | -   | -   | -   |
| z       | -   | -   | -   | -   | -   | -   | -   | -   |
| M1-M2-WGP | f1  | 5499| 7943| -   | -   | 5224| 5683| -   |
| f2      | 3718| 10  | 3946| 679 | 5073| 5495| 5430| 5873|
| f3      | 1299| 5206| -   | -   | 942 | 1450| -   | -   |
| z       | 0.71| 0.89| -   | -   | 0.72| 0.76| -   | -   |
| M1-M2-CP | f1  | -   | -   | -   | -   | -   | -   | -   |
| f2      | -   | -   | -   | -   | -   | -   | -   | -   |
| f3      | -   | -   | -   | -   | -   | -   | -   | -   |
| z       | -   | -   | -   | -   | -   | -   | -   | -   |
| M1-M2-LWT | f1  | 5499| 7943| -   | -   | 5224| 5683| -   |
| f2      | 3718| 10  | 3946| 679 | 5073| 5495| 5430| 5873|
| f3      | 1299| 5206| -   | -   | 942 | 1450| -   | -   |
| z       | 0.71| 0.89| -   | -   | 0.72| 0.76| -   | -   |
| SA-M3   | f1  | 3536| 4158| -   | -   | 3849| 3124| 3530|
| f2      | 1840| 2474| 2699| 2359| 1347| 2065| 9298| 26957|
| f3      | 122  | 158  | 241 | 86  | 0   | 173  | 0   | 108 |
| z       | 0.05 | 0.07 | 0.07 | 0.06 | 0.04 | 0.06 | 0.04 | 0.07 |
programming multi-objective techniques are applied to the mathematical models in each stage. To solve industrial-scale problems in reasonable time, a two-stage solution approach was proposed. In the first stage, jobs are assigned to machines and scheduled in these machines by using proposed simulated annealing algorithm. In the second stage, starting time, completion time and waiting time of the jobs are calculated considering resource constraints. The performance of the approaches (M0-WGP, M0-CP, M0-LWT, M1-M2-WGP, M1-M2-CP, M1-M2-LWT and SA-M3) are compared with randomly derived instances, similar to real-life problems. Numerical results are demonstrated that goal programming versions of mathematical models are superior compared to other multi-objective techniques. Two stage solution approach (SA-M3) has shown remarkable performance for industrial-scale problems within short time. As for future research, it may be desirable to develop a multi-objective metaheuristic approach to compare proposed two-stage solution approach.

Acknowledgments. This work is supported by the Scientific Research Project Fund of Eskisehir Osmangazi University under the project number 2012-5.

Appendix-1: Parameters. \( n \): number of jobs 
\( m \): number of machines 
\( g \): number of resource groups 
\( M \): a very large positive number

| Test No | 9-1 | 10-1 | 11-1 | 12-1 | 13-1 | 14-1 | 15-1 | 16-1 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| M0-WGP  | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| \( f_1 \) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| \( f_2 \) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| \( f_3 \) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| \( z \)  | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| M0-CP   | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| \( f_1 \) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| \( f_2 \) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| \( f_3 \) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| \( z \)  | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| M0-LWT  | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| \( f_1 \) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| \( f_2 \) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| \( f_3 \) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| \( z \)  | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) | \(-\) |
| M1-M2-WGP | \(1865\) | \(4259\) | \(1637\) | \(4430\) | \(1048\) | \(578\) | \(1093\) | \(1744\) |
| \( f_1 \) | \(49222\) | \(144672\) | \(38766\) | \(156241\) | \(15984\) | \(4029\) | \(17236\) | \(399726\) |
| \( f_2 \) | \(3562\) | \(17320\) | \(2274\) | \(22013\) | \(1\) | \(3\) | \(0\) | \(2472\) |
| \( f_3 \) | \(0.24\) | \(0.95\) | \(0.15\) | \(1.0\) | \(0.03\) | \(0.04\) | \(0.03\) | \(0.25\) |
| \( z \)  | \(0.25\) | \(1.0\) | \(0.27\) | \(1.0\) | \(0.12\) | \(0.20\) | \(0.04\) | \(0.22\) |
| M1-M2-CP | \(56691\) | \(158564\) | \(59000\) | \(186331\) | \(25063\) | \(41366\) | \(21170\) | \(42777\) |
| \( f_1 \) | \(3674\) | \(19700\) | \(4637\) | \(22460\) | \(1388\) | \(125\) | \(1954\) |
| \( f_2 \) | \(0.24\) | \(0.95\) | \(0.15\) | \(1.0\) | \(0.03\) | \(0.04\) | \(0.03\) | \(0.25\) |
| \( f_3 \) | \(0.25\) | \(1.0\) | \(0.27\) | \(1.0\) | \(0.12\) | \(0.20\) | \(0.04\) | \(0.22\) |
| \( z \)  | \(0.25\) | \(1.0\) | \(0.27\) | \(1.0\) | \(0.12\) | \(0.20\) | \(0.04\) | \(0.22\) |
| M1-M2-LWT | \(11668\) | \(-\) | \(13751\) | \(-\) | \(2141\) | \(16990\) | \(2203\) | \(14119\) |
| \( f_1 \) | \(123224\) | \(-\) | \(142194\) | \(-\) | \(15990\) | \(178534\) | \(17515\) | \(140421\) |
| \( f_2 \) | \(3611\) | \(-\) | \(4665\) | \(-\) | \(0\) | \(3370\) | \(71\) | \(2534\) |
| \( f_3 \) | \(0.48\) | \(0.54\) | \(0.54\) | \(0.54\) | \(0.54\) | \(0.54\) | \(0.54\) | \(0.54\) |
| \( z \)  | \(0.25\) | \(1.0\) | \(0.27\) | \(1.0\) | \(0.12\) | \(0.20\) | \(0.04\) | \(0.22\) |
| SA-M3   | \(1617\) | \(4096\) | \(1443\) | \(3695\) | \(1113\) | \(1746\) | \(1095\) | \(1350\) |
| \( f_1 \) | \(25013\) | \(109062\) | \(18002\) | \(93653\) | \(3156\) | \(11361\) | \(5052\) | \(9681\) |
| \( f_2 \) | \(3611\) | \(-\) | \(4665\) | \(-\) | \(0\) | \(3370\) | \(71\) | \(2534\) |
| \( f_3 \) | \(0.48\) | \(0.54\) | \(0.54\) | \(0.54\) | \(0.54\) | \(0.54\) | \(0.54\) | \(0.54\) |
| \( z \)  | \(0.14\) | \(0.84\) | \(0.11\) | \(0.70\) | \(0.01\) | \(0.13\) | \(0.02\) | \(0.09\) |
| \( t(\text{sec.}) \) | \(90\) | \(276\) | \(275\) | \(267\) | \(72\) | \(173\) | \(69\) | \(277\) |
\( p_j \): processing time of job \( j \)
\( h_j \): setup time if job \( j \) is first processing job
\( s_{ij} \): sequence-dependent setup time between jobs \( i \) and \( j \)
\( d_j \): delivery time of job \( j \)
\( f_r \): number of type \( r \) resources
\( \text{res}_{ir} \): 1, if job \( j \) uses resource \( r \); 0, otherwise.
\( \beta_{jl} \): 1, if job \( j \) can be processed on machine \( l \), otherwise 0.
\( b_j \): \( l \) (if job \( j \) is assigned to machine \( l \))
\( \delta_{ijkl} \): if job \( j \) is assigned to machine \( l \) in sequence \( k \), 0; otherwise
\( cap \): representative capacity of any machine.
\( \alpha_1, \alpha_2, \alpha_3 \): weights of objectives
\( \gamma \): a small positive constant
\( I_{C_{\text{max}}}^c \): Ideal value for makespan
\( I_{\text{tt}}^t \): Ideal value for total tardiness
\( I_{w}^w \): Ideal value for the total waiting time
\( I_{s}^s \): ideal value for total setup time of each machine
\( I_{n}^n \): ideal value for number of assigned jobs to each machine
\( AI_{C_{\text{max}}}^c \): anti-ideal value for makespan
\( AI_{\text{tt}}^t \): anti-ideal value for total tardiness
\( AI_{w}^w \): anti-ideal value for total waiting time
\( AI_{s}^s \): anti-ideal value for total setup time of each machine
\( AI_{n}^n \): anti-ideal value for number of assigned jobs to each machine

Appendix-2: Decision Variables.

\( C_j \): completion time of job \( j \)
\( T_j \): tardiness of job \( j \)
\( C_{\text{max}} \): completion time of the last job
\( a_j \): starting time of job \( j \)
\( y_{jl} \): 1, if job \( j \) is assigned to machine \( l \); otherwise 0.
\( x_{ijkl} \): 1, if job \( j \) is assigned to machine \( l \) in sequence \( k \), 0; otherwise.
\( \lambda_{ijkl} \): 1, if job \( j \) is assigned to \( k \)th sequence, 0; otherwise.
\( \epsilon_{1j}, \epsilon_{2j}, \epsilon_{3j}, \epsilon_{4j} \): 0-1 decision variables that are used to ensure the constraints for the priority among the jobs
\( w_j \): waiting time for job \( j \)
\( d_{s_j}^+ \): over-achievement of \( I_s^s \) for machine \( l \)
\( d_{s_j}^- \): under-achievement of \( I_s^s \) for machine \( l \)
\( d_{n_j}^+ \): over-achievement of \( I_n^n \) for machine \( l \)
\( d_{n_j}^- \): under-achievement of \( I_n^n \) for machine \( l \)
\( d_{c}^+ \): over-achievement of \( I_{C_{\text{max}}}^c \)
\( d_{c}^- \): under-achievement of \( I_{C_{\text{max}}}^c \)
\( d_{t}^+ \): over-achievement of \( I_{\text{tt}}^t \)
\( d_{t}^- \): under-achievement of \( I_{\text{tt}}^t \)
\( d_{w}^+ \): over-achievement of \( I_w^w \)
\( d_{w}^- \): under-achievement of \( I_w^w \)
\( \theta \): largest deviation for M0
\( \theta_1 \): largest deviation for M1
\( \theta_2 \): largest deviation for M2
REFERENCES

[1] M. Afzalirad and M. Shafipour, Design of an efficient genetic algorithm for resource-constrained unrelated parallel machine scheduling problem with machine eligibility restrictions, *Journal Of Intelligent Manufacturing*, 29 (2018), 423–437.

[2] E. Akyol Ozer and T. Sarac, MIP models and a matheuristic algorithm for an identical parallel machine scheduling problem under multiple copies of shared resources constraints, *TOP*, 27 (2019), 94–124.

[3] O. Alagoz and M. Azizoglu, Rescheduling of identical parallel machines under machine eligibility constraints, *European Journal of Operational Research*, 149 (2003), 523–532.

[4] A. Baykasoglu and F. Ozsoydan, Dynamic scheduling of parallel heat treatment furnaces: A case study at a manufacturing system, *Journal of Manufacturing Systems*, 46 (2018), 152–162.

[5] G. Bektur and T. Sarac, A Mathematical Model and Heuristic Algorithms for an Unrelated Parallel Machine Scheduling Problem with Sequence-Dependent Setup Times, Machine Eligibility Restrictions and a Common Server, *Computers & Operations Research*, 103 (2019), 46–63.

[6] I. A. Chaudhry and P. R. Drake, Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms, *International Journal of Advanced Manufacturing Technology*, 42 (2009), 581–594.

[7] S. G. Dastidar and R. Nagi, Scheduling injection molding operations with multiple resource constraints and sequence dependent setup times and costs, *Computers & Operations Research*, 32 (2005), 2987–3005.

[8] R. Driessel and L. Moench, Scheduling jobs on parallel machines with sequence dependent setup times precedence constraints and ready times using variable neighborhood search, *International Conference on Computers and Industrial Engineering*, (2009), 273–278.

[9] E. B. Edis and C. Oguz, Parallel machine scheduling with flexible resources, *Computers & Industrial Engineering*, 63 (2012), 433–447.

[10] E.B. Edis and I. Ozkarahan, A combined integer/constraint programming approach to a resource constrained parallel machine scheduling problem with machine eligibility restrictions, *Engineering Optimization*, 43 (2011), 135–157.

[11] E. B. Edis and I. Ozkarahan, Solution approaches for a real-life resource-constrained parallel machine scheduling problem, *International Journal of Advanced Manufacturing Technology*, 58 (2012), 1141–1153.

[12] A. Ezugwu and F. Akutsah, An Improved Firefly Algorithm for the Unrelated Parallel Machines Scheduling Problem With Sequence-Dependent Setup Times, *IEEE ACCESS*, 6 (2018), 54459–54478.

[13] B. Gacias, C. Artigues and P. Lopez, Parallel machine scheduling with precedence constraints and setup times, *Computers & Operations Research*, 37 (2010), 2141–2151.

[14] R. Gokhale and M. Mathirajan, Scheduling identical parallel machines with machine eligibility restrictions to minimize total weighted flow time in automobile gear manufacturing, *International Journal of Advanced Manufacturing Technology*, 60 (2012), 1099–1110.

[15] T. Keskinturk, M. B. Yildirim and M. Barut, An ant colony optimization algorithm for load balancing in parallel machines with sequence-dependent setup times, *Computers & Operations Research*, 39 (2012), 1225–1235.

[16] K. Li, Y. Shia, S. Yanga and B. Cheng, Parallel machine scheduling problem to minimize the makespan with resource dependent processing times, *Applied Soft Computing*, 11 (2011), 5551–5557.

[17] X. Li, H. Chehade, F. Yalaoui and L. Amodeo, Fuzzy logic controller based multi-objective meta-heuristics to solve parallel machines scheduling problem, *Journal of Multiple-Valued Logic and Soft Computing*, 18 (2012), 617–636.

[18] S. W. Lin, Z. J. Lee, K. C. Ying and C. C. Lu, Minimization of maximum lateness on parallel machines with sequence-dependent setup times and job release dates, *Computers & Operations Research*, 38 (2011), 809–815.

[19] M. Liu and C. Wu, Scheduling algorithm based on evolutionary computing in identical parallel machine production line, *Robotics and Computer Integrated Manufacturing*, 19 (2003), 401–407.

[20] T. Park, T. Lee and C. O. Kim, Due-date scheduling on parallel machines with job splitting and sequence-dependent major/minor setup times, *International Journal of Advanced Manufacturing Technology*, 59 (2012), 325–333.
[21] R. Ruiz and C. A. Romano, Scheduling unrelated parallel machines with resource-assignable sequence-dependent setup times, *International Journal of Advanced Manufacturing Technology*, 57 (2011), 777–794.

[22] T. Sarac and A. Sipahioglu, Plastik enjeksiyon makinalarının Çizelgelenmesi problemi, (Turkish) [Plastic injection machines scheduling problem], *Journal of Industrial Engineering*, 20 (2009), 2–14.

[23] L. Su, W. Y. Chang and F. D. Chou, Minimizing maximum lateness on identical parallel machines with flexible resources and machine eligibility constraints, *International Journal of Advanced Manufacturing Technology*, 56 (2011), 1195.

[24] I. T. Tanev, T. Uozumi and Y. Morotome, Hybrid evolutionary algorithm-based real-world flexible job shop scheduling problem: Application service provider approach, *Applied Soft Computing*, 5 (2004), 87–100.

[25] A. K. Turker and C. Sel, A Hybrid approach on single server parallel machines scheduling problem with sequence dependent setup times, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 26 (2011), 731–740.

[26] Y. Unlu and S. J. Mason, Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems, *Computers & Industrial Engineering*, 58 (2010), 785–800.

Received August 2018; revised March 2019.

E-mail address: tsarac@ogu.edu.tr
E-mail address: asipahi@ogu.edu.tr
E-mail address: emineakyol@eskisehir.edu.tr