



A Comparative Study of Scalarization Techniques on the Multi-Objective Single Machine-Scheduling Problem Under Sequence-Dependent Setup Time, Release Date and Due Date Constraints

Zeynep Idil ERZURUM CICEK^{*} , Zehra KAMISLI OZTURK 

Eskisehir Technical University, Department of Industrial Engineering, 26555, Eskisehir, Turkey

Highlights

- A multi-objective mixed-integer programming model is proposed for a single machine-scheduling problem.
- Test problems are generated for the considered problem.
- The multi-objective mixed-integer programming model is solved using scalarization techniques.
- The obtained results using three scalarization techniques are compared and evaluated.

Article Info

Received: 25/06/2019
Accepted: 09/01/2020

Keywords

Machine-scheduling
Scalarization
Multi-objective optimization
Sequence-dependent setup times

Abstract

In this paper, we considered a two-objective machine-scheduling problem under sequence-dependent setup time, release date and due date constraints. The problem is formulated as a multi-objective mixed-integer programming model. Two conflicting objectives are considered as minimization of maximum completion time (makespan) and total tardiness. Despite the most use of metaheuristics in this kind of multi-objective problems, here, we try to solve the problem by transforming the two-objectives as a single objective using scalarization techniques. Test instances are generated as proposed in the scheduling literature. The solutions are obtained using Weighted Sum Scalarization, Benson's Method and Pascoletti-Serafini Method. In addition, a comparison of scalarization techniques using Δ performance metric is given on the considered problem instances. The obtained results are evaluated and Δ values, which were obtained for Benson's method, are mostly better than other techniques for the generated test problems.

1. INTRODUCTION

Whether it is the production or service industry, many problems we face in real life are scheduling problems. According to Pinedo's [1] definition, scheduling is a decision-making process that is used in various industries. It deals with the allocation of resources and timeslots to events under single or multiple objectives. In the production side, the machine-scheduling problem is a combinatorial optimization problem. And the generally used objective of this problem is the minimization of jobs' completion times [2].

As Pinedo [1] mentioned, the basic single machine environment is easy to model. However, there can be special constraints about customers, machines, and jobs in manufacturing systems. With numerous kinds of constraints, the single machine-scheduling problem becomes harder to model and solve, but the solution will give more realistic schedules.

In some of the machine scheduling problems, there can be setup times, which must be considered enforcedly. The setup times are considered either sequence-independent or sequence-dependent [3]. When the setup time/cost depends solely on the task to be processed, regardless of its preceding task, the scheduling problem must be considered as sequence-independent. Moreover, in the sequence-dependent type, setup time/cost depends on both the task and its preceding task [4]. Sequence-dependent setup times can be originated from the activities to prepare the machine like changing or cleaning of mold, changing of

^{*}Corresponding author, e-mail: zierzurum@eskisehir.edu.tr

cutting tips, etc. The authors here want to emphasize an important point that, according to Sioud et al. [5], 70% of industrial activities include sequence-dependent setup times.

In this study, a multi-objective single machine-scheduling problem is considered under the minimization of both maximum completion time and total tardiness. The jobs in the considered problem have process times, sequence-dependent setup times, due dates and release dates. The problem $1|s_{ij}|C_{max}$ is strongly NP-hard [1]. The problem that becomes more difficult with the handling of multiple goals is further complicated by the addition of sequence-dependent setup times. In addition to these, applicability to real-life problems has increased. Three multi-objective scalarization techniques are used to propose efficient schedules for randomly generated test problems and the obtained results are compared based on a well-known metric.

The rest of this paper as follows: In Section 2, the studies related to single machine scheduling problems will be addressed. Brief information about multi-objective optimization, scalarization and the well-known scalarization techniques, which are used in the study, are reviewed in Section 3. The definition of the problem and the proposed mixed-integer mathematical model will be explained in Section 4. The generated test problems and computational experiments will be explained in Section 5 and the results will be discussed in Section 6.

2. LITERATURE REVIEW

There is a considerable amount of studies about single machine scheduling problems. In this section, the single machine scheduling studies with sequence-dependent setup times are studied.

To minimize total weighted tardiness with sequence-dependent setup times on a single machine, various algorithms have been proposed like an ant colony optimization algorithm [6], a discrete differential evolution algorithm [7], an Iterated Local Search (ILS) heuristic [8], and a novel memetic algorithm [9]. There are also studies that consider just the total tardiness for the related problem. Luo & Chu [10] developed an algorithm based on branch-and-bound permutation schemes. Sioud et al. [5] proposed a hybrid crossover into a genetic algorithm with concepts from constraint programming. Jacob & Arroyo [11] proposed three heuristics based on the ILS. Keshavarz et al. [12] developed a branch and bound algorithm to minimize total earliness and tardiness penalties. Hinder & Mason [13] proposed a novel integer programming formulation to minimize maximum lateness.

Beside these studies, one of the main objective values that has been tried to be minimized for single machine scheduling problems is the maximum completion time (C_{max}). For a dynamic manufacturing system, Kaplanoglu [14] developed a collaborative multi-agent-based optimization method to minimize C_{max} under both regular and irregular maintenance constraints. Velez-Gallego et al. [15] developed a beam search heuristic, which considers release dates and sequence-dependent setup times to minimize C_{max} .

Nevertheless, there are not too many studies about multi-objective single machine scheduling with sequence-dependent setup times. Choobineh et al. [2] developed a mixed-integer mathematical model and Tabu Search heuristic for a single machine-scheduling problem, which has three objectives. The objectives of the problem are minimizing the completion time, total number of tardy jobs and total tardiness. Duenas & Petrovic [16] proposed a hybrid algorithm with genetic algorithms and local search, in the presence of uncertainty to minimize the maximum and the average tardiness of the jobs. Sioud et al. [5] also declared that in the presence of sequence-dependent setup times, most of the research has focused on either minimizing the sum of setup times or minimizing the sum of job completion times. Wang [17] proposed multi-objective evolutionary algorithms as NSGA-II and SPEA2 to minimize the total expected completion time of jobs and to minimize the maximum of expected times of failure of the machine, simultaneously. Yue et al. [18] presented a hybrid Pareto Artificial Bee Colony algorithm to minimize the maximum completion time and the total weighted tardiness. Rubaiee and Yildirim [19] developed several new multi-objective optimization algorithms based on Ant Colony Optimization to solve a single machine-scheduling problem with minimization of both total completion time and total energy cost objectives. Similarly, Zhang et al. [20] proposed a bi-objective single-machine batch scheduling under time-of-use electricity prices to

minimize the total energy cost and the makespan and presented two heuristics for the solution of the proposed model.

As seen from the literature, there are few studies, which are about multi-objective single machine scheduling problems, so we aimed to contribute to this research area with this study. Moreover, metaheuristic algorithms are mostly proposed to obtain Pareto solutions of multi-objective single machine scheduling problems instead of exact solution methods. We also aimed to contribute to this research area using three well-known scalarization techniques. Weighted Sum scalarization, Benson's scalarization and Pascoletti-Serafini scalarization techniques are chosen for solving the considered multi-objective problem. The reason for selecting these methods is the ease of application and requirement of the smaller number of method parameters. The use of other scalarization technique has been determined as a future work, which is mentioned in the last section in detail. Finally, with this study, release date, and due date, which are not considered much in the literature, are also included in the scheduling process at the same time as the sequence-dependent setup time constraints.

3. MULTI-OBJECTIVE OPTIMIZATION

Many of the real-life problems can be handled as a multi-objective optimization problem (MOP), actually. As also Eicfelder [21] mentioned, these problems with conflicting objectives have many Pareto solutions instead of one, rather than a single best solution.

Traditional single objective optimization solution methods cannot be used for multi-objective optimization problems in which we aim to optimize multiple objectives. Scalarization techniques, evolutionary multi-objective optimization methods, and goal programming are the most preferred solution methods for these kinds of problems. As scalarization techniques are used to solve the multi-objective single machine-scheduling problem in this study, some of the scalarization techniques will be explained in the following section.

3.1. Scalarization Techniques

In general, scalarization means converting a MOP into a suitable single optimization problem. Weighted Sum Scalarization, ϵ -Constraint, Weighted Chebychev's, Conic Scalarization, Pascoletti-Serafini and Benson's are the main scalarization methods used in the solution of MOP.

Basic prerequisites

A multi-objective optimization problem (MOP) is defined as

$$\min_{x \in X} \{f_1(x), \dots, f_p(x)\} \quad (\text{MOP})$$

where $X \subset \mathbb{R}^n$ is a feasible set and $f: \mathbb{R}^n \rightarrow \mathbb{R}^p$ is a vector valued objective function. $Y = f(X) \subset \mathbb{R}^p$ represents the image of the feasible set in the criterion space. Optimal solutions of MOP are considered in the sense of Pareto optimality (efficiency).

A feasible solution $x \in X$ is called efficient if there doesn't exist $x' \in X$ such that $f_k(x') \leq f_k(x)$ for all $k = 1, \dots, p$ and $f_j(x') < f_j(x)$ for some j . In other words, no solution is at least as good as x for all objective, and strictly better for at least one [22].

Definition 1. The solution notion for (MOP) is defined with respect to an ordering cone which is used for ordering the criterion space \mathbb{R}^p [23]. We use the natural ordering cone defined as $\mathbb{R}_+^p = \{x \in \mathbb{R}^p: x_j \geq 0, \forall j = 1, 2, \dots, p\}$.

Definition 2. The set Y is called \mathbb{R}_+^p -closed if $Y + \mathbb{R}_+^p$ is closed. Similarly, Y is called \mathbb{R}_+^p -convex if $Y + \mathbb{R}_+^p$ is convex. If X is a convex set and f_1, f_2, \dots, f_p are convex functions, then $Y + \mathbb{R}_+^p$ -convex [23].

Definition 3. A feasible solution $x \in X$ is called efficient, weakly efficient or properly efficient solution of MOP if $y = f(x)$ is minimal, weakly minimal or properly minimal element of Y , respectively [24].

3.1.1. Weighted Sum Scalarization

It is not wrong to say that the Weighted Sum Scalarization (WSS) method, which was first introduced by Gass and Saaty [25], is the most preferred method for scalarization. If all the weights $(w_i, i = 1, \dots, p)$ are positive, the minimum of following WSS is Pareto optimal

$$\min_{x \in X} \sum_{i=1}^p w_i f_i(x) \quad (\text{WSS}(w)).$$

Although WSS is the commonly most preferred scalarization method because of its practical implementation, there are some limitations.

The w_i parameters are determined by decision makers. However, it's not always possible to obtain all proper efficient solutions of the scalarized problem by all weights. iv. The weights of objectives are taken, while the reference point information is not considered in this method. If the problem has the characteristic of being convex, this scalarization method can guarantee to obtain weakly and properly efficient solutions. In addition to these limitations, the method does not require additional restrictions, which provides ease of use.

3.1.2. Benson's scalarization

One of the other common used scalarization technique is Benson's Scalarization (BS) Method introduced by Benson [26]. Benson's method [23,26] gives an examination of the existence of efficient and properly efficient solutions.

The scalarized optimization problem (BS) can be defined as given:

$$\max \sum_{k=1}^p l_k \quad (\text{BS}),$$

$$\text{s.t. } f_k(x^0) - l_k - f_k(x) = 0, k = \overline{1, p},$$

$$l_k \geq 0, k = \overline{1, p} \quad x \in X,$$

where x^0 from X is a feasible solution, and the efficiency of x^0 is checked by l_1 -norm.

The background of BS method is given by [24]. The BS method can be applied only in the case when the ordering cone C equals R_+^p . When the set $(f(x^0) - \mathbb{R}_+^p) \cap f(X)$ is unbounded, BS cannot guarantee to give finite solution. BS doesn't provide conditions that guarantee the production of properly efficient solutions, while it provides the necessary and sufficient conditions. The problem size increases when we add p number of new decision variables l_i , p number of functional constraints $f_i(x^0) - l_i - f_i(x) = 0, i = 1, \dots, p$ and p nonnegativity constraints $l_i \geq 0$, for the new decision variables. Beside these disadvantages, BS doesn't require convexity assumptions. And also, there is no need to take any parameter from the decision maker like objective weights and reference points.

3.1.3. Pascoletti-Serafini Scalarization

Like WSS and BS methods, Pascoletti-Serafini Scalarization Method (PSS) introduced by Gerstewiz [27] is also a common used scalarization method where optimality is defined through convex cones. The scalarized problem using PSS is given as follows:

$$\min t \quad (\text{PSS}),$$

$$\text{s.t. } a + tr - f(x) \in \mathbb{C},$$

$x \in X, t \in \mathbb{R}$.

where $a \in \mathbb{R}$ and $r \in \mathbb{C}$ are the problem parameters of the (PSS). The main properties of PSS method are given by [24]. Like BS method, PSS also isn't limited by convexity. The solutions which is obtained using PSS scalarization is at least weakly efficient. Beside these advantages, there're some limitations of the method. The PSS method can be applied for arbitrary ordering cone \mathbb{C} . The boundedness from below is not an essential condition for the PSS method. The method uses additional functional constraints of the form $a + tr - f(x) \in \mathbb{C}$, which may an increase in the problem size. If for the given parameter $a \in \mathbb{R}^n$, the straight line $\{ a + tr : t \in \mathbb{R} \}$ does not intersect the objective space $f(X)$ for some $r \in \mathbb{C}$, then the same set of weakly minimal points will be obtained for all $r \in \text{int}(\mathbb{C})$. So, it becomes very important to select the parameter r from the interior of the ordering cone, to guarantee existence of an optimal solution to PSS.

4. PROBLEM DEFINITION AND MATHEMATICAL MODEL

In this study, the problem considers assigning a set of jobs to a single machine. These jobs have non-identical process times. A setup time occurs before a job is processed on a machine and differs based on the processed job. In addition, the jobs have also release dates and due dates. It is also assumed that there isn't any preemption for the jobs. The mixed-integer mathematical model of the problem was developed based on the model, which was proposed by Velez-Gallego [15]. The main difference of the studied model from the model of [15] is the second objective function, which minimizes total tardiness. Besides that, a new constraint group was added to calculate the tardiness of each job if there exists. The notation of the problem is given as follows

Sets

J : Set of jobs

J' : Extended set of jobs

Parameters

p_j : Process time of j^{th} job

r_j : Release date of j^{th} job

d_j : Due date of j^{th} job

s_{jk} : Setup time needed to start j^{th} job after job k

π : Sum of the process times of all jobs in J

M : A very large positive real number

Decision Variables

$x_{jk} = \begin{cases} 1, & \text{if job } k \text{ processed just after job } j \text{ in the sequence} \\ 0, & \text{otherwise} \end{cases}$

y_j : j^{th} job's setup starting time

T_j : Tardiness of job j .

In order to add the setup time of the first job in the schedule, a dummy job is added to the set of jobs. The dummy job doesn't have an effect on objective function values, because all parameters are equal to zero. The proposed multi-objective mixed-integer programming model of the problem is given as follows:

$$\min C_{max} = \pi + \sum_j \sum_k s_{jk} x_{jk} + \sigma \quad (1)$$

$$\min \sum T = \sum_j T_j \quad (2)$$

subject to

$$\sum_{i \in J'} x_{ij} = 1, \quad \forall j \in J' \quad (3)$$

$$\sum_{j \in J'} x_{ij} = 1, \quad \forall i \in J' \quad (4)$$

$$y_1 = 0 \quad (5)$$

$$y_i + p_i + \sum_{q \in J'} s_{qi} x_{qi} - \pi - \sum_{j \in J'} \sum_{k \in J'} s_{jk} x_{jk} - \sigma \leq 0, \quad \forall i \in J' \quad (6)$$

$$y_i + p_i + \sum_{q \in J'} s_{qi} x_{qi} + M(x_{ij} - 1) - y_j \leq 0, \quad \forall i \in J', j \in J' \quad (7)$$

$$y_i + p_i + \sum_{q \in J'} s_{qi} x_{qi} + p_i - d_i \leq T_i, \quad \forall i \in J' \quad (8)$$

$$r_j - y_j \leq 0, \quad \forall j \in J' \quad (9)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in J', j \in J', i \neq j \quad (10)$$

$$y_i, T_i \geq 0 \text{ and integer} \quad \forall i \in J'. \quad (11)$$

The problem has two objectives as minimizing the makespan (C_{max}) in Equation (1) and minimizing total tardiness in Equation (2). The 3th and 4th constraint groups provide that each job has only a predecessor and a successor. Constraint 5 guarantees that the dummy job is the first job in the schedule. The starting times of the remaining jobs in the schedule are calculated in constraint groups (6) and (7). The tardiness value of each job is computed using the constraint group (8). Constraint group (9) provides that a job can't be assigned to the machine before its release time. Constraint (10) and (11) corresponds to the domain of the decision variables.

To validate the mathematical model, eight test problems generated using rules mentioned by Velez-Gallego *et al.* [15], Tan *et al.* [28], and Ragatz [29]. The numbers of jobs in each problem are 15, 25, 35, 45, 55, 65, 75 and 85. The problems with 15, 25, 35 and 45 jobs refer to small-sized problems and the others refer to large-sized problems. The job processing times are normally distributed and the mean and standard deviation parameters of the normal distribution are 100 and 9.5 respectively. The setup times of each job are uniformly distributed with a mean of 9.5 and the range of set up times is taken as 5. The due dates are calculated using a parameter, which is called tardiness factor (TF). TF indicates the mean value of the distribution from which the due dates are generated. TF means approximately to the expected proportion of jobs that will be tardy in a random sequence of the jobs. The mean parameter of the distribution of due dates is calculated as $(1-TF)(N)(\text{mean processing time})$ where N refers to the number of jobs [29]. TF is taken as 0.2. Job release dates have a discrete uniform distribution with the interval $[1, L]$, where $L=N\rho R$. Here, n represents the number of jobs in the test problem, ρ is the expected processing time, and R is a range factor that is used for controlling the dispersion of the release dates over time [15]. The range factor R is also taken as 0.2. The parameters that are used to generate test problems are summarized in Table 1.

Table 1. Parameters of generated test problems

N	Processing Times	Setup Times	TF	R
{15, 25, 35, 45, 55, 65, 75, 85}	$\mu = 100$ $\sigma = 9.5$	$\mu = 9.5$ $r = 5$	0.2	0.2

The generation procedure of test problems is explained in Figure 1.

Input: Number of jobs N , TF , R

1: **Start**

2: Generate N normally distributed random numbers with mean 100 and standard deviation 9.5 for job processing times.

3: Generate N uniformly distributed random numbers with mean 9.5 and range 5 for setup times.

4: Generate N uniformly distributed random numbers with mean $(1-TF)(N)(\text{mean processing time})$ and range $(TF)(N)(\text{mean processing time})$ for due dates.

5: Calculate ρ as the total of the processing times.

6: Calculate $L=N \cdot \rho \cdot R$

7: Generate N uniformly distributed random numbers with the interval $[1, L]$ for release dates.

8: **Finish**

Figure 1. Test problem generation procedure

Afterwards, a dimensional analysis is conducted, and the related data are given in Table 2.

Table 2. Dimensional analysis

Problem No	# of jobs	# of decision variables	# of constraints
1	15	288	337
2	25	728	807
3	35	1368	1472
4	45	2208	2342
5	55	3248	3412
6	65	4488	4682
7	75	5928	6152
8	85	7568	7822

Before running test problems, the mathematical model of the problem is updated according to the scalarization techniques. Using WSS, the objective function of the model transforms into Equation (12) and Equation (13) is added to the model

$$\min z = (w_1 \times z_1) + (w_2 \times z_2) \quad (12)$$

$$w_1, w_2 \geq 0. \quad (13)$$

The equations between Equation (14) and Equation (17) provide Benson's scalarization.

$$\max l_1 + l_2 \quad (14)$$

$$\text{s.t. } f_1(x^0) - l_1 - f_1(x) = 0 \quad (15)$$

$$f_2(x^0) - l_2 - f_2(x) = 0 \quad (16)$$

$$l_1, l_2 \geq 0 \quad (17)$$

where l_1 and l_2 are decision variables of Benson's scalarization technique, f_1 and f_2 denote C_{\max} and total tardiness objectives, respectively.

Lastly, in order to apply PSS, the equations between Equation (18) and Equation (21) are added to the model.

$$\min t \quad (18)$$

$$\text{s.t. } a_1 + t * r_1 - f_1(x) = 0 \quad (19)$$

$$a_2 + t * r_2 - f_2(x) = 0 \quad (20)$$

$$t \in \mathbb{R} . \quad (21)$$

After the scalarization of the objective functions, the functional constraints, which are given between Equation (3) and Equation (11), are added to the model without any modification.

5. COMPUTATIONAL EXPERIMENTS

The scalarized multi-objective mathematical models and scalarization techniques are run using GAMS version 23.3. If only minimization of C_{\max} is considered as a single objective function, the problem itself is an NP-Hard problem. Therefore, the proposed model is allowed to run only 15 minutes for the small-sized problems and 30 minutes for the large-sized ones. Furthermore, the smallest problem was solved in order to show that the two objective functions are conflicting. The objective function weights were changed in the range of 0-1 and the objective function values were obtained by WSS. As a result, while one objective function value increased mostly, a decrease was detected on the other's value. The graph of the objective function values obtained of this experiment is given in Figure 2 and the z_1 and z_2 denote C_{\max} and total tardiness objective values respectively.

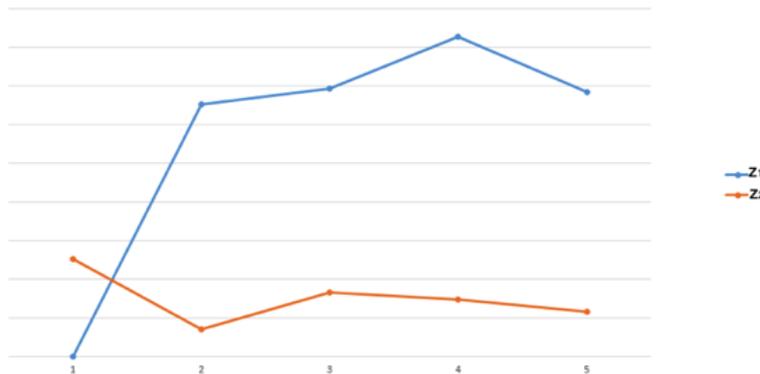


Figure 2. The values of objective functions for Problem 1

The parameters of the methods, which are tuned as (1,10) at the beginning, are changed by increasing and decreasing the values by one in each run. The obtained results for the small and large-sized test problems are given in Table 3 and Figure 3, respectively. The numbers from 1 to 8 represent the generated problem numbers. z_1 (C_{\max}) and z_2 ($\sum_j T_j$) refer to the objective function values which are given in Equations (1) and (2). The values of objective functions, which are obtained by solving models individually, are also given in Table 3. Individual problems are solved under the same time limit. When all problems are solved separately for each objective function, the objective function values obtained are better for some problems when compared with the Pareto results obtained by the scalarization techniques used. For instance, for problem 1, the better z_2 value is obtained by WSS than the problem with the only objective function as minimization of z_2 . However, we can't talk about the same situation for z_1 value. This is the result of the time limitation given by the decision makers.

Table 3. The objective function values

No	Individual objective function values	WSS		Benson		PSS	
		Z ₁	Z ₂	Z ₁	Z ₂	Z ₁	Z ₂
1	z ₁ =2365 z ₂ =6465	2396	6624	2644	10521	2850	6465
		2387	6491	2610	9306	2855	6465
		2389	6472	2582	9495	2860	6465
		2385	6507	2582	9495	2855	6442
		2377	6493	2468	7649	2860	6442
		2372	6444	2892	14084	2875	6464
		2379	6479	2594	9084	2905	6519
		2374	6442	2683	10545	2876	6442
		2379	6479	2694	10729	2894	6471
		2372	6472	2570	9319	2910	6496
2	z ₁ =3960 z ₂ =20439	4027	20215	3925	18652	5809	19917
		4009	20142	3924	18735	5932	20315
		4026	20135	3927	18679	5715	19547
		4025	20013	3924	18611	5934	20274
		4003	19953	3931	18774	5804	19807
		4010	20223	3925	18777	5767	19658
		4014	20062	3926	18733	5950	20257
		3981	19997	3924	18642	5778	19648
		3997	19895	3937	18847	5799	19698
		4005	20008	3927	18647	5958	20214
3	z ₁ =6123 z ₂ =38620	5728	36462	5755	37645	5880	45511
		5719	36147	5799	37578	5701	39482
		5757	36857	5768	37623	5656	37547
		5741	36121	5771	38295	5741	40335
		5743	36878	5768	37405	5744	40116
		5764	36600	5756	37576	5775	41128
		5725	36298	5730	36947	5716	38712
		5691	35824	5752	37256	5724	39601
		5725	36884	5750	37404	5773	41269
		5718	36349	5784	37574	5765	40968
4	z ₁ =7429 z ₂ =68077	7507	70279	7421	69136	9267	80743
		7461	71019	7392	68957	8835	76901
		7408	69453	7429	68138	9169	79719
		7442	69495	7389	68158	8690	75472
		7434	69565	7395	67940	9728	84394
		7449	69573	7404	67439	8446	73195
		7383	68960	7454	67149	10293	89104
		7410	69610	7395	68575	9267	80136
		7481	71364	7420	68047	9538	82392
7410	68147	7379	67939	10703	92357		

Table 3. The objective function values (continued)

No	Individual objective function values	WSS		Benson		PSS	
		z ₁	z ₂	z ₁	z ₂	z ₁	z ₂
5	z ₁ =9004 z ₂ =97402	9106	98643	9033	98498	20827	111762
		9180	99441	9130	103837	19010	101968
		9071	95736	9130	104739	20110	107821
		9068	97655	9093	101797	20194	108226
		9101	98007	9181	104894	20326	108885
		9072	98074	8992	98059	21881	117165
		9077	97867	9048	98935	19878	106390
		9051	97799	9168	104997	21796	116603
		9022	97380	9083	101654	22311	119310
6	z ₁ =10972 z ₂ =142778	9000	95433	9086	102296	19766	105654
		10692	137170	11233	171623	13497	153163
		10711	134900	11131	167881	13756	155988
		10723	137310	10912	156215	14464	163894
		10686	132250	11083	166887	15193	172029
		10732	137420	11177	170038	14863	168163
		10773	136860	11159	170125	14275	161384
		10749	138130	11136	168107	15055	170078
		10741	137900	11227	170097	15313	172869
7	z ₁ =12966 z ₂ =197984	10645	136120	10913	156327	15048	169751
		10721	138260	11089	168925	15427	173895
		12561	191040	14435	336893	33676	248410
		12608	190207	14367	325293	32015	236113
		12712	201499	14497	332644	35699	263235
		12640	199023	13826	296475	36959	272478
		12636	196967	13932	306178	33957	250295
		12609	196870	14368	331271	35078	258511
		12604	196425	13875	295087	35734	263295
8	z ₁ =15153 z ₂ =244719	12692	201461	14427	336067	34682	255494
		12803	212373	13932	305532	34568	254608
		12582	195176	14121	317358	36957	272151
		14092	227990	15706	382912	62716	365147
		14022	230660	15965	398790	50131	291814
		13998	227890	15899	388775	51607	300339
		14138	234170	15899	380155	49318	286959
		14133	234270	15612	364889	50723	295071
		14066	228870	15781	376740	57230	332855
		14089	232540	16079	407228	57390	333715
		14206	244550	15940	396387	54393	316222
		14023	231690	15982	401947	56581	328874
		14354	250380	15806	385642	53109	308627

In Table 3, some Pareto solutions obtained by WSS, Benson and PS scalarization methods are given. As explained in Section 5, the proposed scalarized models are allowed to run only 15 minutes and 30 minutes for the small and large sized problems, respectively. For instance, for problem 1, while z₁ value is 2387 in solution 2, it's increased to 2389 in solution 2. However, while z₂ value is 6491 in solution 2, it's decreased to 6472 in solution 3. It means that, solution 2 and solution 3 obtained by WSS method for problem 1, are not dominating each other. So, they're Pareto points. The same nondomination relation can be seen on the some solutions obtained by Benson's method and PSS for problem 3. For this problem, while z₁ value is

5768 in solution 5, it's decreased to 5756 in solution 6. However, while z_2 value is 37405 in solution 5, it's increased to 37576 in solution 6. It means that, solution 5 and solution 6 obtained by Benson's method for problem 3, are not dominating each other. Similarly, while z_1 value is 5741 in solution 4, it's increased to 5744 in solution 5. However, while z_2 value is 40355 in solution 4, it's decreased to 40116 in solution 5. It means that, solution 4 and solution 5 obtained by PSS for problem 3, are not dominating each other. So, some Pareto points are obtained by Benson's method and PSS, also. Beside these Pareto solutions, weakly efficient solutions are also obtained by PSS. For problem 1, while z_2 values of the first three solutions are the same, their z_1 values are changing.

Figure 3 shows the solution spaces obtained by all 3 techniques for each problem.

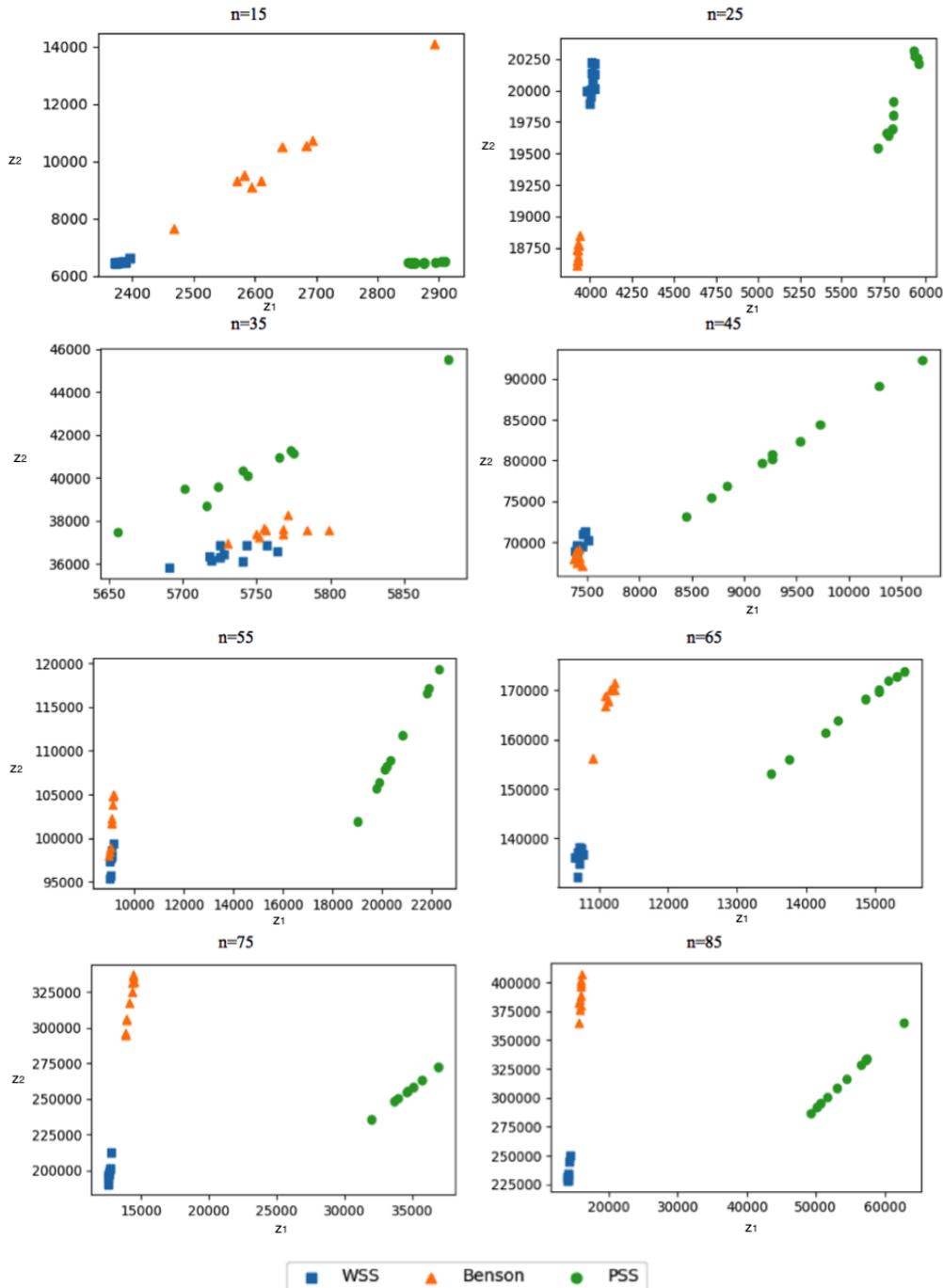


Figure 3. Obtained solution spaces

In a MOP study, there are two goals to be reached: convergence to the Pareto optimal set and maintenance of diversity in solutions of the Pareto optimal set [30]. Some metrics can be used to determine the quality of MOP problems' solutions. Jiang *et al.* [31] categorized these metrics into four groups as namely capacity, convergence, diversity, and convergence-diversity. Some of the metrics can be reviewed from the classification table, which is presented by Jiang *et al.* [31]. In this study, the spread indicator (Δ) as a diversity metric, is used for evaluating the performance of chosen scalarization techniques. The spread indicator given in Equation (22) measures the extent of spread achieved among the obtained solutions [32]:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}} . \quad (22)$$

The d_f and d_l parameters are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained solutions, which are in the nondominated set. The geometric representation of the distance parameters is given in Figure 4.

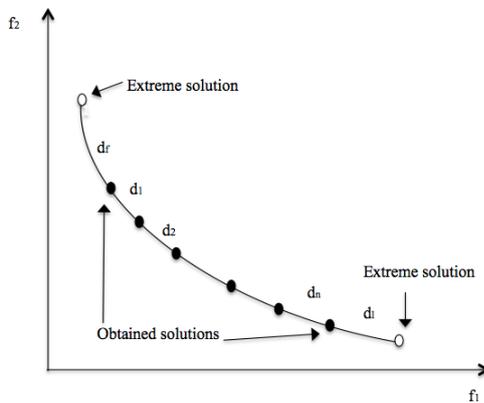


Figure 4. The distances of Δ metric

The figure illustrates all distances mentioned in equation (12) for two objectives as f_1 and f_2 . The parameter \bar{d} is the average of distances of N solutions that are on the best-nondominated front. With solutions, there are consecutive distances [30].

The Δ values calculated for each technique are given in Table 4. We apply this indicator after normalization of the objective functions [30]. As given in Equation (23), min-max normalization converts a value d from P to d' in the range $[0,1]$ [33]

$$d' = \frac{d - \min(P)}{\max(P) - \min(P)} . \quad (23)$$

Table 4. Δ values

Number of jobs	WSS	Benson	PS
15	0.580	0.871	0.661
25	0.339	0.603	0.418
35	0.520	0.434	0.751
45	0.622	0.453	0.567
55	0.729	0.595	0.615
65	0.575	0.652	0.625
75	0.762	0.545	0.633
85	0.721	0.510	0.560
Average	0.606	0.583	0.628
Standard deviation	0.138	0.139	0.107

As seen from Table 4, the Δ values for Benson's method are lower than the other techniques for the problems, which have 35, 45, 55, 75 and 85 jobs. In the rest of the problems, the Δ values of WSS method are better. Furthermore, the average Δ value of Benson method is better than Δ values of WSS and PSS techniques. Figure 5 also shows the distribution of Δ values. The reason for the better Δ values of the Benson method is that the objective function values obtained with different parameters of the method have less variation.

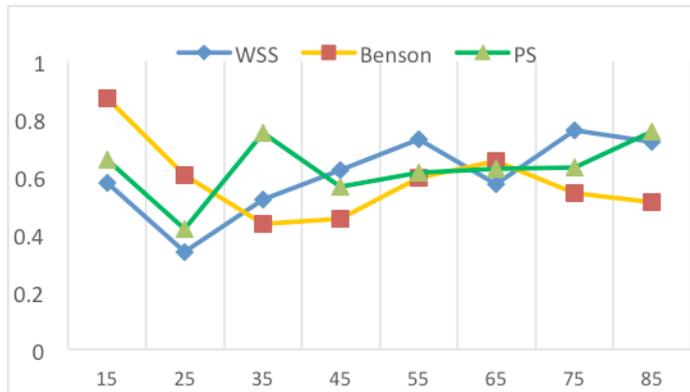


Figure 5. The Δ values

Due to the time constraint for problem solution, a limited number of Pareto solutions were obtained for eight problem types by the three scalarization methods. The spread refers to the range of values covered by the solutions.

6. CONCLUSIONS

It is known that any permutation of the jobs gives the same C_{\max} when the main purpose is the minimization of C_{\max} in the single machine problem. However, when the problem is extended by considering setup times and the release dates the problem gets more complicated. In this study, an NP-hard multi-objective scheduling problem with special constraints is considered and three scalarization techniques are coded for the solution of the test problems. In general, multi-objective metaheuristics are developed for multi-objective problems in the literature. The main contribution of this study is performing scalarization techniques in the solution of multi-objective problems instead of metaheuristics. Moreover, a time limit for the run of each model is applied in order to simulate a manufacturing firm that has to decide the production schedule in short periods. The time constraint allowed us to achieve the best possible solutions within the given time without achieving optimal results.

The obtained results are evaluated using Δ performance metric. As discussed in Section 5 the Δ values, which were obtained for Benson's method, are mostly better than other techniques for the generated test problems. This demonstrates the results obtained with Benson's method have a low degree of variation and the Pareto fronts can be obtained using this scalarization technique. To improve the solutions obtained, the running parameters of chosen scalarization techniques especially for Benson's method, should be optimized to obtain better Pareto fronts and the time limits can be increased.

The 15 and 30 minutes of running times are acceptable for this NP-hard problem, but while the problem size increases, it becomes compulsory to increase the running time for more than 30 minutes. Therefore, multi-objective metaheuristic algorithms can be used. Also, different scalarization techniques such as Conic, Weighted Chebychev's and ϵ -Constraint scalarization techniques can be experienced.

CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

REFERENCES

- [1] Pinedo, M., *Scheduling Theory, Algorithms, and Systems*. 4th ed. New York, NY, Springer, (2008).
- [2] Coobineh, F.F., Mohebbi, E., Khoo, H., "A multi-objective tabu search for a single machine scheduling problem with sequence-dependent setup times", *European Journal of Operations Research*, 175(1):318-337, (2006).
- [3] Kuo, W.H., Yang, D.L., "Single machine scheduling with past-sequence-dependent setup times and learning effects", *Information Processing Letters*, 102(1):22-26, (2007).
- [4] Allahverdi, A., Soroush, H., "The significance of reducing setup times/setup costs", *European Journal of Operations Research*, 187(3):978-984, (2008).
- [5] Sioud, A., Gravel, M., Gagn, C., "A hybrid genetic algorithm for single machine scheduling problem with sequence-dependent setup times", *Computers and Operation Research*, 39:2415-2424, (2012).
- [6] Liao C.J., Juan H.C., "An ant colony optimization for single-machine tardiness scheduling with sequence-dependent setups", *Computers and Operations Research*, 34(7):1899-1909, (2007).
- [7] Tasgetiren M., Pan, Q., Liang, Y., "A discrete differential evolution algorithm for the single machine total weighted tardiness problem with sequence-dependent setup times", *Computers and Operations Research*, 36(6):1900-1915, (2009).
- [8] Subramanian A., Battarra, M., Potts, C.N., "An Iterated Local Search heuristic for the single machine total weighted tardiness scheduling problem with sequence-dependent setup times", *International Journal of Production Research*, 52:2729–2742, (2014).
- [9] Gonzalez, M.A., Vela, C.R., "An efficient memetic algorithm for total weighted tardiness minimization in a single machine with setups", *Applied Soft Computing*, 37:506-518, (2015).
- [10] Luo, X., Chu, F., "A branch and bound algorithm of the single machine schedule with sequence-dependent setup times for minimizing total tardiness", *Applied Mathematics and Computation*, 183(1):575-588, (2006).
- [11] Jacob, V.V., Arroyo, J.E.C., "ILS heuristic for the single-machine scheduling problem with sequence-dependent family setup times to minimize total tardiness", *Journal of Applied Mathematics*, 2016:1-15, (2016).
- [12] Keshevarz T., Savelsbergh, M., Salmasi, N., "A branch-and-bound algorithm for the single machine sequence-dependent group scheduling problem with earliness and tardiness penaltie", *Applied Mathematical Modelling*, 39(20):6410-6424, (2015).
- [13] Hinder, O., Mason, A.J., "A novel integer programming formulation for scheduling with family setup times on a single machine to minimize maximum lateness", *European Journal of Operational Research*, 262(2):411-423, (2017).
- [14] Kaplanoglu V., "Multi-agent based approach for single machine scheduling with sequence-dependent setup times and machine maintenance", *Applied Soft Computing*, 23:165-179, (2014).

- [15] Velez-Gallego, M., Maya, J., Montoya-Torres, J., "A beam search heuristic for scheduling a single machine with release dates and sequence-dependent setup times to minimize the makespan", *Computers and Operations Research*, 73:132-140, (2016).
- [16] Duenas, A., Petrovic D., "Multi-objective genetic algorithm for single machine scheduling problem under fuzziness", *Fuzzy Optim. Decis. Making*, 7(1):87-104, (2006).
- [17] Wang S., "Bi-objective optimisation for integrated scheduling of single machine with setup times and preventive maintenance planning", *International Journal of Production Research*, 51(12): 3719–3733, (2013).
- [18] Yue, L., Guan, Z., Saif, U., Zhang, F., Wang, H., "Hybrid pareto artificial bee colony algorithm for multi-objective single machine group scheduling problem with sequence-dependent setup times and learning effects", *SpringerPlus*, 5(1):1593, (2016).
- [19] Rubaiee, S., Yildirim, M.B., "An energy-aware multiobjective ant colony algorithm to minimize total completion time and energy cost on a single-machine preemptive scheduling", *Computers & Industrial Engineering*, 127: 240-252, (2019).
- [20] Zhang, S., Che, A., Wu, X., Chu, C., "Improved mixed-integer linear programming model and heuristics for bi-objective single-machine batch scheduling with energy cost consideration", *Engineering Optimization*, 50:8, 1380-1394, (2018).
- [21] Eichfelder G., "An adaptive scalarization method in multiobjective optimization", *SIAM Journal on Optimization*, 19(4):1694-1718, (2009).
- [22] Ehrgott, M., Gandibleux, X., "Approximate solution methods for multiobjective combinatorial optimization", *TOP Journal*, 12(1):1-63, (2004).
- [23] Soleimani-damaneh, M., Zamani, M., "On Benson's scalarization in multi-objective optimization", *Optim. Lett.*, 10(8):1757-1762, (2016).
- [24] Kasimbeyli, R., Kamisli Ozturk, Z., Kasimbeyli, N., Yalcin, G., İcmen Erdem, B., "Comparison of some scalarization methods in multiobjective optimization", *Bulletin of the Malaysian Mathematical Sciences Society*, (2017).
- [25] Gass, S., Saaty, T., "The computational algorithm for the parametric objective function", *Naval Research Logistics*, 12(1):1-89, (1955).
- [26] Benson H.P., "Existence of efficient solutions for vector maximization problems", *Journal of Optimization Theory and Applications*, 26(4):569-580, (1978).
- [27] Gerstewitz C., "Nichtkonvexe dualität in der vektoroptimierung", *Wissensch Zeitschr*, 25(3):357-364, (1983).
- [28] Tan, K.C., Narasimhan, R., Rubin, P.A., Ragatz, G.L., "A comparison of four methods for minimizing total tardiness on a single processor with sequence-dependent setup times", *Omega*, 28(3):313-326, (2000).
- [29] Ragatz G., "A branch-and-bound method for minimum tardiness sequencing on a single processor with sequence-dependent setup times", in *Proceedings of the 24th Annual Meeting of the Decision Sciences Institute*, 1375-1377, (1993).

- [30] Deb, K., Pratap, A., Agrawal, S., Meyarivan, T., "A fast and elitist mul-tiobjective genetic algorithm: Nsga-II", *IEEE Transactions on Evolutionary Computation*, 6(2):182-197, (2002).
- [31] Jiang, S., Ong, Y., Zhang, J., Feng, L. "Consistencies and contradictions of performance metrics in multiobjective optimization", *IEEE Transactions on Cybernetics*, 44(12):2391–2404, (2014).
- [32] Durillo, J., Nebro, A., Garcia-Nieto, J., Alba, E., On the Velocity Update in Multi-Objective Particle Swarm Optimizers, Editors: Coello, C.A., Dhaenens, C., Jourdan, L. in *Multi-Objective Nature Inspired Computing, Studies in Computational Intelligence, Vol 272*, Springer, Berlin, Heidelberg, (2010).
- [33] Jain, Y., Bahandre, S., "Min max normalization based data perturbation method for privacy protection", *International Journal of Computer and Communication Technology*, 3(4):45-50, (2014).