

# Preferred Solutions of the Ground Station Scheduling Problem using NSGA-III with Weighted Reference Points Selection

Margarita Antoniou

Computer Systems Department

Jožef Stefan Institute

Jožef Stefan Intl. Postgraduate School

Ljubljana, Slovenia

margarita.antoniou@ijs.si

Gašper Petelin

Computer Systems Department

Jožef Stefan Institute

Jožef Stefan Intl. Postgraduate School

Ljubljana, Slovenia

gasper.petelin@ijs.si

Gregor Papa

Computer Systems Department

Jožef Stefan Institute

Jožef Stefan Intl. Postgraduate School

Ljubljana, Slovenia

gregor.papa@ijs.si

**Abstract**—The Ground station Scheduling Problem refers to the allocation of the communication tasks, among the ground stations and satellites. In general, the problem is formulated as a many-objective problem. The NSGA-III is an algorithm developed to solve such problems. Due to its selection operator that uses a number of reference points, the NSGA-III gives users the option to specify their own reference points. In this paper, we use this opportunity by generating distributed reference points, shifted depending on weights. A specific weight is assigned to each objective that corresponds to reference points for preferred solutions. The generation of these reference points and their effect on the final objective function values of the Pareto front are first tested on the DTLZ2 test function with 4 objectives and for a different combination of weights. Finally, various weights are applied to an instance of the Ground station Scheduling Problem, leading to Pareto fronts that favor specific objectives and feasible schedules.

**Index Terms**—many-objective, satellite scheduling problem, NSGA-III, preferred solutions

## I. INTRODUCTION

The development of space science and technology has increased the number of satellites orbiting the earth. At the same time, the network of ground stations available to communicate with these satellites remains rather limited. The service of satellites relies on this communication between ground stations and satellites. Therefore, an appropriate allocation of the time for satellites communicating ground stations is very important for the space industry. This gives rise to particularly challenging scheduling problems as the resources between space and ground entities are limited.

The Ground station Scheduling Problem (GSP) involves reasonable arrangements of satellites, time windows for completing tasks, e.g., telemetry tasks, and maximizing the ground

station usage. The GSP is a many-objective problem in its general formulation [1]. It has been solved among others by weighted-sum methods [2], multi-objective approaches such as with a memetic algorithm [3] or genetic algorithm [4], an ant colony optimization [5] and heuristic and local search methods [6]. These methods find either one solution of the Pareto front or several well-spread solutions of the Pareto front.

The optimization methods for multi/many-objective optimization problems are divided into 3 categories from a decision-making point of view. When no prior information about the preferred solutions exist, the whole Pareto front needs to be explored, and the decision-maker is choosing one or more preferred optima from the Pareto-optimal solution set at the end of the optimization (a posterior method), which is basically any traditional multi/many-objective evolutionary algorithms, such as NSGA-II [7], MOEA/D [8], etc. When some information regarding the preferred region of a Pareto Front is known in advance, the decision-makers are usually interested in exploring only that small part of the front and a priori optimization methods can be applied, such as [9]. Lately, interactive preference methods have been proposed, e.g. in [10], where the decision maker's preferences are taken into account dynamically during the optimization, saving computational cost.

The formulation of GSP we adopt in this paper is the one used in [2], considering the following objectives: 1) maximizing the events that fall inside the available Access Windows of the ground stations, 2) minimizing the clashes when more than one satellite is communicating with the same ground station, 3) maximizing the time that is required to finalize specific telemetry tasks, such as the download of images, and finally, 4) taking advantage as much as possible from the ground station network by minimizing their idle time. As we have already noted in [11] and [12], the objectives of access windows and clashes are in fact constraints. By simultaneously solving this many-objective problem, it may lead to solutions of the Pareto front, that result in infeasible final schedules. Therefore one is interested in a particular part

This work was supported by the Slovenian Research Agency (research core funding No. P2-0098), by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 692286 (SYNERGY), and under the Marie Skłodowska-Curie grant agreement No. 722734 (UTOPIAE).

978-1-7281-8393-0/21/\$31.00 ©2021 IEEE

of the Pareto-optimal front, that gives priority to these two objectives.

The NSGA-III [13] algorithm is similar to the NSGA-II [14] in its behavior but is further adapted to better solve many-objective problems, where the main difference is the changed selection operator, that uses a number of reference points. It should be noted that the reference points in the NSGA-III play a similar role with that of searching directions of subproblems in the MOEA/D. More specifically, the use of multiple reference points in the NSGA-III guides the population to converge towards the Pareto Front along with different search directions. The NSGA-III can therefore be regarded as a combination of Pareto-based algorithm and decomposition-based evolutionary algorithm [15]. The NSGA-III gives users the option to specify their own reference points and therefore can be used as apriori preference-based optimization algorithm. The idea is to find Pareto optimal points that are closer to these supplied reference points, focusing on the specific region of interest.

In this paper, we use this option by modifying the uniformly distributed reference points sampled from Das and Dennis's method, shifting them depending on the specific weights of the objectives for preferred solutions. Then, NSGA-III is applied to the real-world problem of the GSP problem. The rest of the paper is organized as follows. Firstly, the NSGA-III and the proposed reference point generation are described and explained in Section II. In Section III the notation and the mathematical formulation of the Ground station Scheduling Problem are presented, along with the specific implementation with the NSGA-III. In section IV the experimental setup is reported along with the experiments and the relevant results. Finally, in Section V we conclude the paper and give some future steps of the research.

## II. NSGA-III AND PREFERRED SOLUTIONS

### A. NSGA-III

The NSGA-III consists of a framework similar to the NSGA-II, where crossover and mutation operators are applied to generate offspring population and employ a fast non-dominated sorting approach to determine the non-dominated rank of individuals. An elite preservation strategy is utilized to select a new generation from the parent and offspring population. The main difference between these two algorithms is the selection mechanism, which is modified to work better with many-objective problems. The crowding distance of the NSGA-II is replaced in the NSGA-III with a number of reference points that improve the diversity and the spreadiness of the Pareto optimal fronts.

At first, the algorithm generates an initial population of size  $N$  and a series of well-spread reference points on a hyperplane. The points are placed on the normalized hyperplane –  $(M-1)$ -dimensional unit simplex – by using Das and Dennis's systematic approach [16]. Let  $p$  be the division and is specified by the user. Then the total number of reference points ( $H$ ) in an  $M$ -objective problem is:

$$H = \binom{M+p-1}{p} \quad (1)$$

A solution of the population during the optimization process is then associated with a reference point according to the shortest perpendicular distance between its solution and the reference line, starting from the origin and passing through the corresponding reference point. In the case of these widely distributed points, the obtained solutions have more chances to be diverse on the Pareto-optimal front. More about the next steps of NSGA-III can be found in papers [13], [17].

### B. Generating reference points for preferred objectives

The NSGA-III gives the option to the users to mark their own preferred reference points, to obtain near Pareto-optimal solutions close to these reference points. In this section, we explain the reference point sampling method used in this paper. Das and Dennis's method is adopted and modified in the last step, giving preference to a specific area of the Pareto front that favors specific objectives, depending on some weights that are decided in advance.

Each objective has importance translated to a weight  $w_i$  that is a positive real value. These values are used to transform the reference points space toward objectives with higher importance (greater weights). Let  $r = (r_1, r_2, \dots, r_M)$  be a single reference point out of  $H$  generated reference points in an  $M$ -objective problem by the Das and Dennis's method and let  $w = (w_1, w_2, \dots, w_M)$  be the real-valued weighting vector that signals the importance of individual objectives. Each point  $r$  from a reference point set is moved to a new location in the following way:

$$r_{new} = \left( \frac{r_1/w_1}{\sum_{q=1}^M r_q/w_q}, \frac{r_2/w_2}{\sum_{q=1}^M r_q/w_q}, \dots, \frac{r_M/w_M}{\sum_{q=1}^M r_q/w_q} \right) \quad (2)$$

Figure 1 illustrates an example of reference points distribution sampled on a two objective unit simplex before (black points) and after (green points) using weights, using divisions  $p = 6$  and weights =  $(0.2, 0.8)$ . The shifting of the original points to the new positions, that are favoring the objective with the higher weights can be easily seen. For example, when we have the reference point  $r = (0.5, 0.5)$ , then the new position of the reference point is:

$$r_{new} = \left( \frac{0.5/0.2}{3.125}, \frac{0.5/0.8}{3.125} \right) = (0.8, 0.2) \quad (3)$$

Please note that the NSGA-III by default is used for minimization problems. Therefore, we seek points that are minimizing the preferred objective function.

In Figure 2 the initial reference points are shown with orange colour, while with blue colour the final Pareto front solutions are shown, for the baseline case (Fig.2a) and the case with weights  $(0.6, 0.2, 0.3)$  (Fig.2b) for the DTLZ2 with three objectives. The reference points dictate the final solutions on the Pareto front. The well spread Pareto front of the baseline

case is clearly shifted to favor the objective with higher weight, seen as denser points on the right side of the Pareto front in (Fig.2b).

### III. GROUND STATION SCHEDULING PROBLEM

In this section, the GSP is described shortly. The mathematical formulation and notation of the problem are presented, along with the parameters, variables, and objective functions used for the optimization problem.

#### A. Problem Description

The GSP has a goal to optimize the schedule of the communication between satellites (SCs) and the ground stations (GSs). The formulation of the objectives corresponds to the input and output parameters of the benchmark instance from Khafa et. al [1], [2]. The mathematical notation is the same as presented in papers [11] and [12]. We assume that ground stations can communicate at most with one satellite at a time.

#### B. Mathematical Formulation

The notation and the mathematical problem is presented as follows:

##### Parameters

- $s \in 1, \dots, S$  satellite set, index  $s$ ;
- $g \in 1, \dots, G$  ground station set, index  $g$ ;
- $h \in 1, \dots, H$  set of available Access Windows for a specific  $g$  and a specific  $s$  for all days of the schedule, index  $h$ ;
- $d \in 1, \dots, D$  & set of days, index  $d$ ;
- $tw_{sg}^h$ :  $h^{th}$  time window between a specific  $g$  and a specific  $s$ ;
- $T_{AOS}(tw_{sg}^h), T_{LOS}(tw_{sg}^h)$  are the visibility and losing signal times of a  $g$  from a  $s$ ;

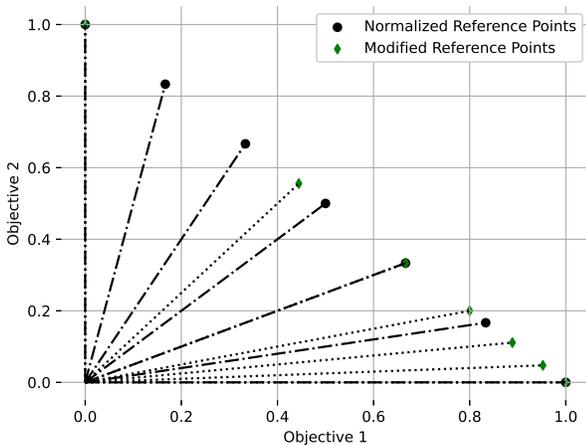
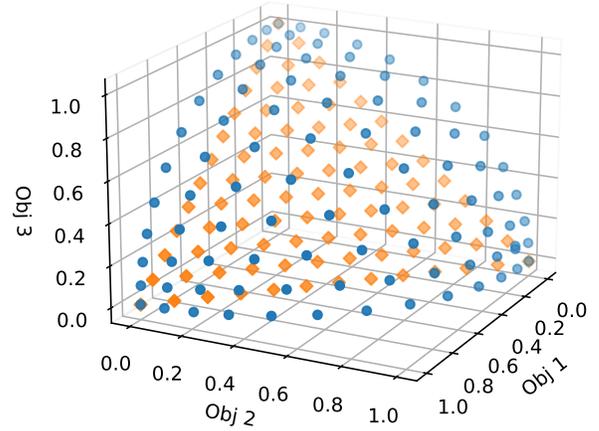
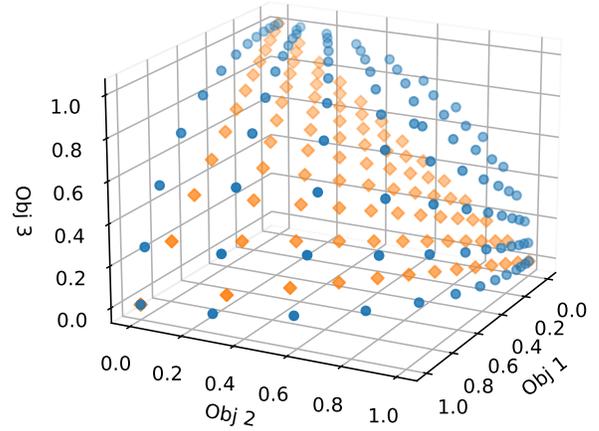


Fig. 1: Illustration of reference points sampled by Das and Dennis's method before and after using weights on a two objective unit simplex. The example is for divisions  $p = 6$  and weights = (0.2, 0.8).



(a) Standard NSGA-III



(b) NSGA-III with weights ( $w_1 = 0.6, w_2 = 0.2, w_3 = 0.2$ )

Fig. 2: Final solutions (blue) and reference points (orange) obtained by (a) standard NSGA-III and (b) NSGA-III with weights ( $w_1 = 0.6, w_2 = 0.2, w_3 = 0.2$ ) for the DTLZ2 problem with 3 objectives,  $p = 12$ .

- $\forall g \in G, s \in S \text{ AW}_{s,g} = \bigcup_{h=1}^H [T_{AOS}(tw_{sg}^h), T_{LOS}(tw_{sg}^h)]$  where AW defines all the time periods  $s$  and  $g$  can communicate;
- $k_s^d \in 1, \dots, K$  are requirements for each  $s$  each day  $d$ ;
- $T_{beg}(k_s^d), T_{end}(k_s^d)$  are the beginning and ending time of a requirement where connection has to be established for at least  $T_{req}(k_s^d)$  during a specified period  $d$ .

#### Decision Variables

- $n_{sg}^m \in N$  an event of the schedule, where  $m \in M$  is the consecutive number of event when a specific  $g$  communicates with a specific  $s$ ,  $N$  is the total number of events of the schedule;
- $T_{start}(n_{sg}^m), T_{dur}(n_{sg}^m)$  Starting and Duration time between  $s$  and  $g$ .

## Objective Functions

**Access windows fitness function:** Maximizing the visibility windows or Access Windows of SCs and GSs, meaning maximizing the time windows during which a  $g$  can establish communication with an  $s$ . For a specific schedule, this means that we aim to maximize the number of events that fall into these time windows. Events outside these time windows lead to infeasible schedules.  $\forall g \in G, s \in S, m \in M$

$$f_{AW}(n_{sg}^m) = \begin{cases} 1 & \text{if } [T_{start}(n_{sg}^m), T_{start}(n_{sg}^m) + T_{dur}(n_{sg}^m)] \subseteq AW_{s,g} \\ 0 & \text{else} \end{cases} \quad (4)$$

$$Fit_{AW} = \frac{\sum_{m=1}^M \sum_{g=1}^G \sum_{s=1}^S f_{AW}(n_{sg}^m) * 100}{N} \quad (5)$$

**Communication clash fitness function:** A communication clash occurs when different satellites are communicating with the same Ground Station at the same time, leading to infeasible schedules. To minimize infeasible events, the clashes that are produced between several SCs to one GS are minimized. From  $n_{gs}^m$  we create the sets  $\forall s : n_g^l \in N$  where  $l \in L \subset N$  is the index of the  $m^{th}$  event of a specific  $g$  to all the  $s$ , after its events are sorted in ascending order for a fixed  $g$  and  $\forall s$  according to their  $T_{start}(n_s^m)$  Then:

$$f_{sc}(n_g^l) = \begin{cases} -1 & \text{if } T_{start}(n_s^{l+1}) < T_{start}(n_s^l) + T_{dur}(n_s^l) \\ 0 & \text{else} \end{cases} \quad (6)$$

$$Fit_{CS} = \frac{N + \sum_{g=1}^G \sum_{l=1}^L f(n_g^l) * 100}{N} \quad (7)$$

**Communication time requirements fitness function:** Some tasks, e.g., data download tasks, need some minimum time requirements in order to be completed. The starting and ending times of these tasks for each day and satellite are known in advance. The final schedule should satisfy as much as possible these requirements. The fitness function is the following:

$$\forall n \in N \text{ and } \forall k \in K$$

$$f(k_s^d, n_{sg}^m) = \|[T_{start}(n_{sg}^m), T_{start}(n_{sg}^m) + T_{dur}(n_{sg}^m)] \cap [T_{beg}(k_s^d), T_{end}(k_s^d)]\| \quad (8)$$

$$f_{TR}(k_s^d) = \begin{cases} 1 & \text{if } (\sum_{g=1}^G \sum_{m=1}^M f(k_s^d, n_{sg}^m)) \geq T_{Req}(k_s^d) \\ 0 & \text{else} \end{cases} \quad (9)$$

**Ground Station usage fitness function:** With this fitness function, we are maximizing the percentage of the GSs busy time to the total available communication time of the GS. The idea behind this is to minimize the GSs idle time.

$$Fit_{GU} = \frac{\|\cup_{m=1}^M \cup_{g=1}^G \cup_{s=1}^S [T_{start}(n_{gs}^m), T_{start}(n_{gs}^m) + T_{dur}(n_{gs}^m)]\|}{\sum_{g=1}^G T_{total}(g)} * 100 \quad (10)$$

where  $T_{total}(g)$  is the total available time of the GS, in this case the number of days of the schedule.

## C. Implementation of NSGA-III for solving GSP

For solving the GSP with the NSGA-III, the following representation of the chromosome was adopted. Each chromosome encodes a schedule as a list of communication events, where each event is represented by five binary variables.  $I$  is a binary variable that indicates whether the specific event is taken into account or not.  $SCs$  represents the satellites with their corresponding  $GSs$  and their starting  $t_{Start}$  and duration times  $t_{Dur}$ . One chromosome consists of a number of these tuples/events to create a whole schedule. The structure of the tuple can be expressed by a matrix of decision variables as:

$$X = (I, SC, GS, t_{Start}, t_{Dur}) \quad (11)$$

As a crossover operator the HUX was selected, while the Bit-Flip mutation was used as a mutation strategy for the specific implementation.

As mentioned also in our earlier paper [11], the objectives  $Fit_{AW}$  and  $Fit_{CS}$  are a measure of the feasibility of the schedule. By solving the problem simultaneously and obtaining well-spread Pareto front, many of the solutions of the optimization will result in a different final schedule. The events that are giving 0 values to these objectives, such as the ones that are outside the AW, will be omitted. The lesser of these events there are, the more representative to the final feasible schedule the Pareto front is. Taking this into account, we are employing weights to reference point distribution on the NSGA-III, that favor these objectives. The final goal is to obtain a Pareto front around the feasible region of the final schedules. Therefore, the combination of weights used in this implementation are the following:

- GSP Baseline Instance
- GSP Instance 1 with the weights (1.5, 0.1, 1.0, 0.01). With these weights, we test the Pareto Front solutions favoring the  $Fit_{AW}$  and the  $Fit_{TR}$  objectives. These are the same weights that were applied in the objectives in the reference paper [1] for solving the problem as a single objective problem with the weighted-sum method.
- GSP Instance 2 with the weights (1.5, 1.0, 0.1, 0.01). Here the weights are chosen such as to give higher importance to the  $Fit_{AW}$  and the  $Fit_{CS}$ , which are the objectives that are actually constraints.
- Instance 3 with weights (0.85, 0.05, 0.05, 0.05). Here we give the priority to the  $Fit_{AW}$  objective since all the events should firstly fall inside the AW.

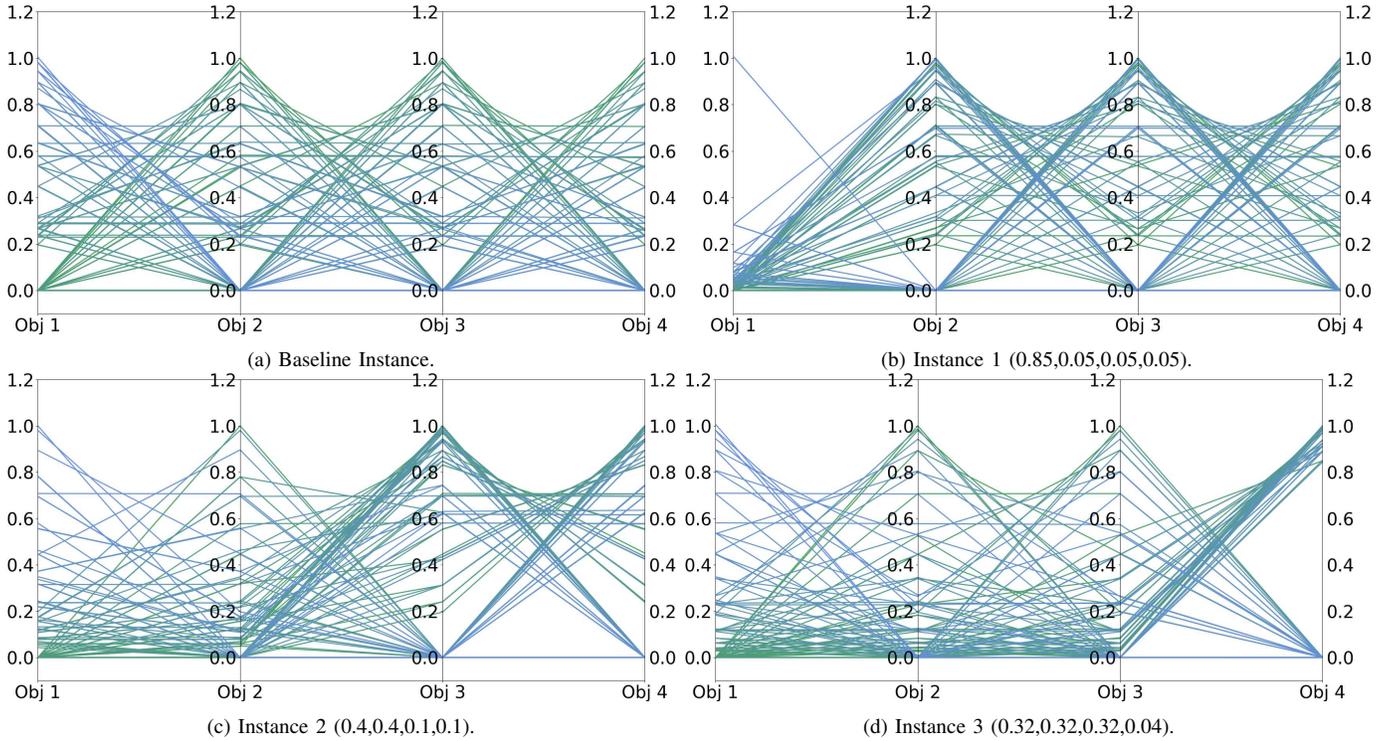


Fig. 3: Parallel coordinate plots of non-dominated solutions of one-run obtained by different instances of the NSGA-III on 4-objective DTLZ2 test problem. (a) no weights, (b) higher weight for objective 1, (c) higher weights for objectives 1 and 2, and (d) higher weights for objectives 1, 2, and 3.

## IV. EXPERIMENTS AND RESULTS

### A. Experimental setup

For the implementation of the NSGA-III, the platypus<sup>1</sup> framework in python is used. The first series of experiments refers to the DTLZ2 test function with 4 objectives, found at the platypus framework as well. The second series of experiments are those of the GSP problem instances. The first, small size, of the benchmarks generated by Xhafa et al. from STK toolkit<sup>2</sup>, where there are 5 Ground Stations, 10 Spacecraft and the number of days is 10 is used in our experiments.

The instances are independently run 30 times on an Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz, 1 TB of RAM, and the Fedora operating system.

In Table I, the control parameter values used in our experiments are reported. The relevant code can be found at the gitlab repository<sup>3</sup>.

TABLE I: CONTROL PARAMETERS USED IN THE REPORTED RESULTS.

	DTLZ2	GSP
Crossover rate	SPX(0.3)	HUX(0.3)
Mutation rate	Uniform(0.001)	BitFlip(0.001)
Population size	84	84
Function Evaluations	30000	120000
Division ( $p$ )	6	6

<sup>1</sup><https://platypus.readthedocs.io/en/latest/getting-started.html>

<sup>2</sup>[https://www.researchgate.net/publication/260086344\\_GS\\_Scheduling\\_Inputs](https://www.researchgate.net/publication/260086344_GS_Scheduling_Inputs)

<sup>3</sup><https://repo.ijs.si/e7/scheduling/cec-2021-satellites-scheduling-nsgaiii>

### B. Results for DTLZ2 with 4 objectives

To confirm and test the method explained in Section II, we solve the DTLZ2 [18] test problem with 4 objectives 30 times with the standard version of the NSGA-III and the weight reference point version of the NSGA-III. To understand better the influence of the weights, we use different combinations of weights and report the results. In Table II we report the mean, median and the standard deviation of each objective after 30 runs, along with the minimum and maximum values obtained. To demonstrate the influence of the weights on the values of the objectives, we report the relative mean improvement of each objective. The Relative Mean Improvement (RMI) is computed as follows:

$$RMI = 100 * \frac{(mean_w - mean_b)}{mean_b} \quad (12)$$

where  $mean_w$  is the mean value of the final objective values obtained by the weighted NSGA-III instance and  $mean_b$  is the mean value of the final objective values obtained by the standard NSGA-III in the baseline instance. Note that "negative improvement" means deterioration for the specific objective values.

The combination of weights used in this experiment are the following:

- Baseline Instance: No weights (or equal weights), corresponding to the standard NSGA-III
- Instance 1: weights (0.850, 0.050, 0.050, 0.050). In this instance, we test the influence of the weights when one

TABLE II: STATISTICAL RESULTS OF THE OBJECTIVES FOR THE DIFFERENT REFERENCE POINTS INSTANCES OF THE NSGA-III AFTER 30 RUNS FOR THE DTLZ2 WITH 4 OBJECTIVES.

Instances		Mean	Median	Min	Max	std	% RMI
Baseline Instance	Obj 1	0.368	0.270	0.000	1.017	0.340	
	Obj 2	0.368	0.270	0.000	1.015	0.340	
	Obj 3	0.367	0.269	0.000	1.013	0.340	
	Obj 4	0.367	0.268	0.000	1.015	0.339	
Instance 1 w[0.850,0.050,0.050,0.050]	Obj 1	0.052	0.016	0.000	1.044	0.122	+85.931
	Obj 2	0.432	0.408	0.000	1.049	0.377	-17.519
	Obj 3	0.431	0.408	0.000	1.012	0.377	-17.261
	Obj 4	0.431	0.407	0.000	1.051	0.377	-17.450
Instance 2 w[0.400,0.400,0.100,0.100]	Obj 1	0.202	0.108	0.000	1.036	0.253	+45.078
	Obj 2	0.202	0.109	0.000	1.035	0.253	+45.108
	Obj 3	0.490	0.578	0.000	1.069	0.397	-33.286
	Obj 4	0.488	0.578	0.000	1.022	0.396	-32.920
Instance 3 w[0.320,0.320,0.320,0.040]	Obj 1	0.220	0.111	0.000	1.062	0.286	+40.310
	Obj 2	0.219	0.111	0.000	1.017	0.286	+40.410
	Obj 3	0.219	0.111	0.000	1.009	0.286	+40.367
	Obj 4	0.638	0.913	0.000	1.008	0.453	-73.698

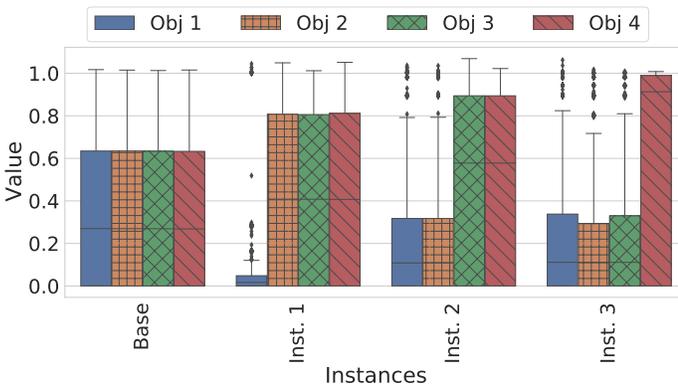


Fig. 4: Boxplot of objective values obtained by the different instances of the NSGA-III for the 4-objective DTLZ2 problem.

objective is set up to be clearly more important than the rest 3 objectives.

- Instance 2, weights (0.400, 0.400, 0.100, 0.100), where 2 objectives are equally more important than the other two.
- Instance 3: weights (0.320, 0.320, 0.320, 0.040), where 3 objectives are equally more important than the last one.

The shape of the Pareto front is affected by the weights used, which is illustrated in Figure 3. This Figure depicts the parallel coordinate plot of non-dominated solutions of one-run obtained by different reference points instances of the NSGA-III on the test problem is presented. The values of Table II are illustrated in Fig.4 as a boxplot. It is worth noting, that for Instance 1, where objective 1 has a weight of 0.85, there is an 84.931% improvement in the mean value. For the Instance 2, the first 2 objectives with weights 0.4 and 0.4 share an almost 45% improvement of their mean objective values, while for Instance 3, the first 3 cases share a 40% of improvement of their mean objectives values. Moreover, the objectives with lower weights result in the deterioration of their mean objectives values. This is expected for the specific problem, as its Pareto Front is concave and the objectives can be regarded as separable [19].

### C. Results for Ground Station Scheduling Problem

The combinations of weights mentioned in Section III are used for these experiments and their statistical results are reported in Table III. A boxplot of these results is presented in Figure 6. A large number of outliers in the values of each objective in all cases is justified since we take into account the solutions on the whole Pareto Front. In Figure 5 parallel coordinate plots of non-dominated solutions of one of the 30 runs are plotted for the different instances. As we can see from the GSP Baseline Instance results, FitAW and FitCS seem to be non-conflicting objectives. This leads to results such as in the GSP Instance 3, wherein a mean improvement of 21.011% of the FitAW objective is obtained and a 6.552% improvement of the FitCS objective, despite the higher weights being assigned only to FitAW. It is obvious that the weights have a different influence on the objectives, compared to the DTLZ2 problem. For example, for Instance 1, where objective 1 (FitAW) and objective 3 (FitTR) have higher weights, their mean improvement is 9.537% and 0.166% respectively, while for objective 2 that has a lower weight, the mean improvement reaches 2.036%. This further implies the collaborative nature of objective 1 and objective 2. Therefore, giving higher importance to FitAW, leads to solutions that are solving to (near)-optimality FitCS as well, reaching the target of finding solutions of this region of the Pareto Front. For the GSP Instance 2, where priority is given to FitAW and FitCS, they show a small improvement of 0.8 and 1.6% in their mean values. This difference of the influence of weights on the final results is related to the complexity of the problem, being large scale, and the shape of the Pareto Front (which is not known in advance for this problem).

## V. CONCLUSIONS AND FUTURE WORK

The paper presents the application of the NSGA-III on the GSP, by using the option of selecting the reference points for preferred solutions. The reference point generation is based on the importance of the objectives, specified by the user as weights at the beginning of the optimization. This method

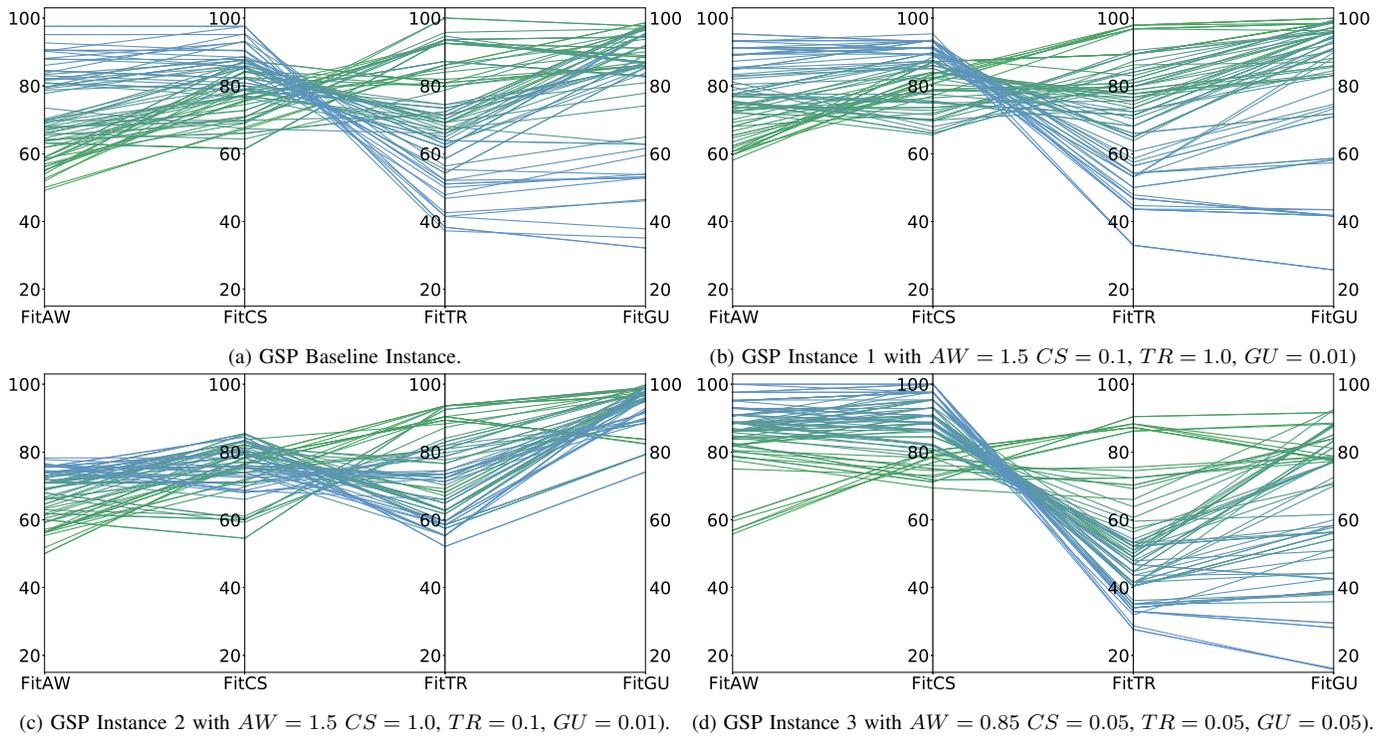


Fig. 5: Parallel coordinate plots of non-dominated solutions of one run obtained by different instances of the NSGA-III on 4-objective Ground Station Scheduling Problem.

TABLE III: STATISTICAL RESULTS OF THE OBJECTIVES FOR THE DIFFERENT REFERENCE POINTS INSTANCES OF THE NSGA-III AFTER 30 RUNS FOR THE GSP.

Instances		Mean	Median	Min	Max	Std	% RMI
GSP Baseline Instance	Obj 1	68.937	69.643	27.778	97.561	11.479	
	Obj 2	78.925	80.000	45.283	97.561	9.020	
	Obj 3	76.242	76.596	29.787	100.000	15.031	
	Obj 4	86.898	91.973	18.317	99.840	13.542	
GSP Instance 1 w[1.5,0.1,1.0,0.01]	Obj 1	75.512	77.273	35.938	97.500	10.747	+9.537
	Obj 2	80.532	81.356	55.319	100.000	7.843	+2.036
	Obj 3	76.368	78.723	31.915	100.000	14.441	+0.166
	Obj 4	86.031	94.674	21.068	99.949	16.654	-0.997
GSP Instance 2 w[1.5,1.0,0.1,0.01]	Obj 1	69.537	71.698	34.545	89.796	11.087	+0.870
	Obj 2	80.214	81.818	48.333	97.872	8.441	+1.633
	Obj 3	75.581	75.532	37.234	100.000	11.928	-0.866
	Obj 4	93.496	97.597	39.536	99.963	9.148	+7.593
GSP Instance 3 w[0.850,0.050,0.050,0.050]	Obj 1	83.421	85.455	38.776	100.000	9.616	+21.011
	Obj 2	84.096	85.417	56.140	100.000	9.427	+6.552
	Obj 3	62.793	61.702	26.596	97.872	16.274	-17.639
	Obj 4	73.013	78.015	14.399	99.122	19.599	-15.979

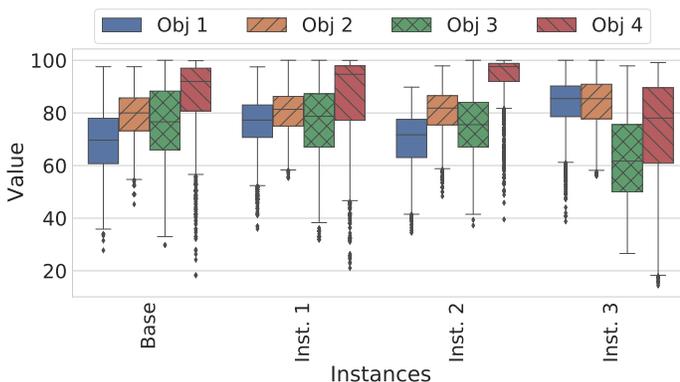


Fig. 6: Boxplot of objective values obtained by the different instances of the NSGA-III for the GSP problem.

is first tested for a combination of weights on the DTLZ2 test problem, to evaluate their influence on the final solutions. Then it is applied to one instance of the GSP problem, with a combination of weights motivated by the importance of the objectives interpreted as constraints. The applied algorithm indeed shifts the Pareto Front to the preferred solutions, when high weights are assigned.

Further testing on the method to different test problems could be interesting, as the influence of weights changes

depending on the nature of the Pareto fronts. Moreover, the algorithm could be tested on different real-world problems, where the users know in advance a part of the region of the Pareto front they are interested in.

#### REFERENCES

- [1] F. Xhafa, X. Herrero, A. Barolli, and M. Takizawa, "Using STK toolkit for evaluating a ga base algorithm for ground station scheduling," in *2013 Seventh International Conference on Complex, Intelligent, and Software Intensive Systems*. IEEE, 2013, pp. 265–273.
- [2] F. Xhafa and A. W. Ip, "Optimisation problems and resolution methods in satellite scheduling and space-craft operation: a survey," *Enterprise Information Systems*, pp. 1–24, 2019.
- [3] Y. Du, L. Xing, J. Zhang, Y. Chen, and Y. He, "Moea based memetic algorithms for multi-objective satellite range scheduling problem," *Swarm and Evolutionary Computation*, vol. 50, p. 100576, 2019.
- [4] Y.-J. Song, X. Ma, X.-J. Li, L.-N. Xing, and P. Wang, "Learning-guided nondominated sorting genetic algorithm ii for multi-objective satellite range scheduling problem," *Swarm and Evolutionary Computation*, vol. 49, pp. 194–205, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2210650219301701>
- [5] Z. Zhang, F. Hu, and N. Zhang, "Ant colony algorithm for satellite control resource scheduling problem," *Applied Intelligence*, vol. 48, no. 10, pp. 3295–3305, 2018.
- [6] L. Barbulescu, A. E. Howe, J.-P. Watson, and L. D. Whitley, "Satellite range scheduling: A comparison of genetic, heuristic and local search," in *International Conference on Parallel Problem Solving From Nature PPSN VII*. Springer, 2002, pp. 611–620.
- [7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [8] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [9] F. Wang, Y. Li, H. Zhang, T. Hu, and X.-L. Shen, "An adaptive weight vector guided evolutionary algorithm for preference-based multi-objective optimization," *Swarm and Evolutionary Computation*, vol. 49, pp. 220–233, 2019.
- [10] Y.-N. Guo, X. Zhang, D.-W. Gong, Z. Zhang, and J.-J. Yang, "Novel interactive preference-based multiobjective evolutionary optimization for bolt supporting networks," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 4, pp. 750–764, 2019.
- [11] M. Antoniou, G. Petelin, and G. Papa, "On formulating the ground scheduling problem as a multi-objective bilevel problem," in *International Conference on Bioinspired Methods and Their Applications*. Springer, 2020, pp. 177–188.
- [12] G. Petelin, M. Antoniou, and G. Papa, "Multi-objective approaches to ground station scheduling for optimization of communication with satellites," *Optimization and Engineering*, pp. 1–38, 2021.
- [13] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints," *IEEE Transactions on evolutionary computation*, vol. 18, no. 4, pp. 577–601, 2013.
- [14] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Parallel Problem Solving from Nature PPSN VI*, ser. Lecture Notes in Computer Science, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, and H.-P. Schwefel, Eds. Springer Berlin / Heidelberg, 2000, vol. 1917, pp. 849–858.
- [15] H. Li, K. Deb, Q. Zhang, P. N. Suganthan, and L. Chen, "Comparison between moea/d and nsga-iii on a set of novel many and multi-objective benchmark problems with challenging difficulties," *Swarm and Evolutionary Computation*, vol. 46, pp. 104–117, 2019.
- [16] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM journal on optimization*, vol. 8, no. 3, pp. 631–657, 1998.
- [17] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach," *IEEE Transactions on evolutionary computation*, vol. 18, no. 4, pp. 602–622, 2013.
- [18] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary multiobjective optimization*. Springer, 2005, pp. 105–145.
- [19] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.