Multi-objective approaches to ground station scheduling for optimization of communication with satellites

Gašper Petelin^{1,2} · Margarita Antoniou^{1,2} · Gregor Papa^{1,2}

Received: 31 July 2020 / Revised: 8 January 2021 / Accepted: 9 March 2021 / Published online: 28 March 2021
© The Author(s) 2021

Abstract

The ground station scheduling problem is a complex scheduling problem involving multiple objectives. Evolutionary techniques for multi-objective optimization are becoming popular among different fields, due to their effectiveness in obtaining a set of trade-off solutions. In contrast to some conventional methods, that aggregate the objectives into one weighted-sum objective function, multi-objective evolutionary algorithms manage to find a set of solutions in the Pareto-optimal front. Selecting one algorithm, however, for a specific problem adds additional challenge. In this paper the ground station scheduling problem was solved through six different evolutionary multi-objective algorithms, the NSGA-II, NSGA-III, SPEA2, GDE3, IBEA, and MOEA/D. The goal is to test their efficacy and performance to a number of benchmark static instances of the ground scheduling problem. Benchmark instances are of different sizes, allowing further testing of the behavior of the algorithms to different dimensionality of the problem. The solutions are compared to the recent solutions of a weighted-sum approach solved by the GA. The results show that all multi-objective algorithms manage to find as good solution as the weighted-sum, while giving more additional alternatives. The decomposition-based MOEA/D outperforms the rest of the algorithms for the specific problem in almost all aspects.

Keywords Satellite scheduling · Evolutionary optimization · Multi-objective approach

All authors contributed equally to the paper.

✉ Gregor Papa
gregor.papa@ijs.si

Gašper Petelin
gasper.petelin@ijs.si

Margarita Antoniou
margarita.antoniou@ijs.si

¹ Computer Systems Department, Jožef Stefan Institute, Ljubljana, Slovenia

² Jožef Stefan International Postgraduate School, Ljubljana, Slovenia

1 Introduction

There are thousands of artificial (i.e., man-made) satellites in the Earth's orbit. They are used either for taking pictures of the planet (e.g., to help meteorologists in weather prediction) or for taking pictures of other planets and faraway galaxies (e.g., to help scientists better understand the universe) (NASA 2017). In general, these satellites can be classified as Earth observation satellites, communications satellites, navigation satellites, weather satellites, and space telescopes.

Earth observation satellites (EOSs) are intended for environmental monitoring, meteorology, map making, and others. EOSs acquire images of the Earth's surface according to given observation requests, like earth resources exploration, natural disaster surveillance, or military reconnaissance.

The satellite scheduling problem consists of aligning tasks (like communications and control maneuvers) with resources (ground stations, space stations, and satellites). In the satellite scheduling problem, some resources are stationary, while others are orbiting the earth. To optimally align the tasks and resources several constraints have to be met, which classifies the problem as constraint-optimization one (Pemberton and Galiber 2000).

More specifically, the satellite scheduling problem consists of scheduling communication windows between ground stations and satellites. These include day-to-day activities of an operational satellite, like configuration of different sensors on the satellite to collect data, maintenance of the satellite status, and transmission of data/commands between satellites and ground stations. During the allocated communication window the task to be performed should be completed (Xhafa and Ip 2019). The goal of scheduled data downloads from satellites to ground stations is to maximize the total amount of downloaded data while satisfying all visibility, energy, and data requirements; this is also known as Multiple-Satellite, Multiple Ground Station Scheduling Problem (MMSP) (Castaing 2014).

1.1 Related work

The ground station scheduling (GSS) is a variation of satellite scheduling optimization problem. As is the case with most of the scheduling problems, this is a very complex problem and is proven to be NP-hard (Barbulescu et al. 2004). As mentioned above, satellite scheduling problems require taking into consideration multiple objectives, making the GSS problem multi-objective in its general formulation (Xhafa et al. 2013). In simultaneous optimization, multi-objective problems are solved by considering solutions at the edge of the approximate Pareto front.

The ground station scheduling problem or other variants of the satellite scheduling problem has been approached both with by single and multi-objective methods. Several single objective metaheuristic algorithms were compared in Globus et al. (2003), where the objectives were combined into a weighted-sum. A comparison among a simple heuristic, local search, and a genetic algorithm was done, showing the ability of the GA to perform better on the larger scaled problems (Barbulescu

et al. 2002). In a similar research, (Lala et al. 2015) the effectiveness of the exploitation (through different local search methods) and exploration (through modified Genetic Algorithms) for the problem of Ground Station Scheduling was studied, showing that it highly depends on the fitness function and a balance between the two would be beneficial.

In Tangpattanakul et al. (2012) the authors presented a biased random-key genetic algorithm for solving the bi-objective optimization problem of agile earth-observing satellites and later (Tangpattanakul et al. 2015) solved the same problem with an indicator based multi-objective local search. A many-objective approach was implemented in Zhang et al. (2016), where the satellite-ground time synchronization problem was tackled with a decomposition-and-integration based evolutionary algorithm. For the same problem, (Wang et al. 2019) proposed a similar DI-MOEA method, by transforming the many-objective to a multi-objective problem in order to use a multi-objective algorithm. In Gentile et al. (2019), the autonomous scheduling of orbit determination campaigns for tracking spacecraft in deep-space was approached, by developing a dedicated Structured-Chromosome Genetic Algorithm, with an outcome of several optimal solutions for the decision-maker to choose.

Ground station scheduling can be regarded as a scheduling problem in general. In Zhao et al. (2017) an improved MOEA/D was proposed to solve multiple objectives of job shop scheduling problems, giving better results by Pareto-dominance EAs. In Zhai et al. (2015), the NSGA-II and rule-based heuristic algorithms are used to simultaneously maximize the profit and robustness of the schedule.

The aim of this paper is to evaluate and present the performance of different multi-objective evolutionary algorithms on a set of benchmark instances of different dimensionality of the GSS problem. This work extends the initial work (Papa and Petelin 2019) where the satellite scheduling problem was only presented as a small exemplary case within the presentation of the remote optimization tool. The rest of the paper is organized as follows. Firstly we describe the problem and its formulation in Sect. 2. In Sect. 3 the selected multi-objective evolutionary algorithms and their implementation are described. Moreover, in Sect. 4 the control parameter setup is explained. The results and their discussion of this study are presented in Sect. 5. Finally, in Sect. 6 we conclude our work and put forward some future research steps.

2 Problem description

The formulation of the problem and its objectives used in this paper are taken from Xhafa et al. (2013), Xhafa and Ip (2019), along with the benchmark problems. The benchmark problems are of 3 different sizes. The small instance size (I_S) consists of 5 ground stations and 10 spacecraft, the medium instance size (I_M) consists of 10 ground stations and 15 spacecraft, while the large instance size (I_L) has 15 ground stations and 20 spacecraft. All of the instance sizes include a schedule of 10 days. In this section, we give some basic definitions and notation of the problem along with the mathematical formulation of the objective functions taken into account in the problem.

Table 1 Parameters

Symbol	Description
$s \in 1, \dots, S$	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
$T_{AOS}(tw_{sg}^h), T_{LOS}(tw_{sg}^h)$	Starting and ending time when communication between s and g is physically possible
$k_s^d \in 1, \dots, K$	Requirements for each s on each day d
$T_{beg}(k_s^d), T_{end}(k_s^d); T_{req}(k_s^d)$	The beginning and ending time of a specific requirement. During this time, Communication has to be established for at least the required amount of time

Table 2 Decision variables

Symbol	Description
$n_{sg}^m \in N$	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 .

2.1 Basic definitions

The GSS focuses on scheduling station resources for communications between spacecraft (satellites) and ground stations that are feasible and follow different constraints. Since problems are usually large and with a lot of constraints, computing the exact solution is very computationally demanding.

The parameters and decision variables of the model are described in Tables 1 and 2.

2.2 Optimization objectives

2.2.1 Access window objective function

Due to the nature of the optimization problem, communication between spacecraft and ground stations is not always possible. Spacecrafts can communicate with ground stations only during specified time periods called visibility or access window. The goal of the access window objective function is to maximize the duration of communication between spacecraft and visible ground stations (see Fig. 1a). Let $AW_{s,g}$ be the union of

Fig. 1 Examples of **a** Fit_{AW} , **b** Fit_{CS} , **c** Fit_{TR} and **d** Fit_{GU} computation

all the periods where spacecraft s can establish communication with ground station g . This equation is thus defined for $\forall g \in G, s \in S$.

$$AW_{s,g} = \bigcup_{h=1}^H [T_{AOS}(tw_{sg}^h), T_{LOS}(tw_{sg}^h)] \quad (1)$$

New indicator function can be defined that will output value 1 if desired communication window n_{sg}^m from a generated schedule falls inside a range of visibility windows and 0 otherwise. This function is defined for $\forall g \in G, s \in S, m \in M$.

$$f_{AW}(n_{sg}^m) = \begin{cases} 1 & \text{if } [T_{start}(n_{sg}^m), T_{start}(n_{sg}^m) + T_{dur}(n_{sg}^m)] \subseteq AW_{s,g} \\ 0 & \text{else} \end{cases} \quad (2)$$

Access window objective function is thus defined as the normalized number of valid communication windows between pairs of ground stations and spacecraft if communication is physically possible.

$$Fit_{AW} = \frac{\sum_{m=1}^M \sum_{g=1}^G \sum_{s=1}^S f_{AW}(n_{sg}^m) * 100}{N} \quad (3)$$

2.2.2 Communication clash objective function

Schedule constructed to solve a concrete problem might contain periods where multiple spacecraft are simultaneously trying to communicate with the same ground station making this schedule unfeasible. Certain schedules might, therefore, produce clashes that need to be considered by the communication clash objective function. To detect clashes between events, they are first divided into multiple groups corresponding to the ground station they are trying to communicate with and then sorted by the starting times. A check is then made that determines if the current event clashes with the event that starts just before it.

Let from n_{sg}^m 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 $T_{start}(n_{sg}^m)$ (see Fig. 1b for details).

$$f_{CS}(n_g^l) = \begin{cases} -1 & \text{if } T_{start}(n_s^{l+1}) < T_{start}(n_s^l) + T_{dur}(n_s^l) \\ 0 & \text{else} \end{cases} \quad (4)$$

The total communication clash objective is normalized values of windows where no clash was detected.

$$Fit_{CS} = \frac{N + \sum_{g=1}^G \sum_{l=1}^L f_{CS}(n_g^l) * 100}{N} \quad (5)$$

2.2.3 Communication time requirement objective function

When communication between spacecraft and ground station is established, it must be open for a certain amount of time so all the data can be reliably transferred between them. The amount of time necessary for successful communication is given by $T_{req}(k_s^d)$. This requirement is satisfied only if it takes place during $T_{beg}(k_s^d)$ and $T_{end}(k_s^d)$ as defined in the problem. The communication time requirement objective is thus equivalent to a percentage of requirements that are satisfied with a given schedule (details are presented in Fig. 1c). The first equation defines the intersection between a specific requirement k_s^d and event n_{sg}^m and is valid for $\forall k \in K$ and $\forall n \in N$:

$$f(k_s^d, n_{sg}^m) = \|[T_{start}(n_{sg}^m), T_{start}(n_{sg}^m) + T_{dur}(n_{sg}^m)] \cap [T_{beg}(k_s^d), T_{end}(k_s^d)]\| \quad (6)$$

For a single requirement to be satisfied the following indicator function is used that returns 0 or 1:

$$f_{TR}(k_s^d) = \begin{cases} 1 & \text{if } (\sum_{g=1}^G \sum_{m=1}^M f(k_s^d, n_{sg}^m)) \geq T_{req}(k_s^d) \\ 0 & \text{else} \end{cases} \quad (7)$$

The final objective function is computed by counting the number of satisfying requirements and is defined as:

$$Fit_{TR} = \frac{\sum_{s=1}^S \sum_{d=1}^D f_{TR}(k_s^d)}{K} * 100 \quad (8)$$

2.2.4 Ground station usage objective function

The number of the ground stations is usually smaller than the number of spacecraft that are trying to communicate with them. The goal of this objective is to maximize the usage of ground stations. The fitness value is thus equal to the utilization percentage of each ground station (presented in Fig. 1d). It is defined as:

$$Fit_{GU} = \frac{\| \bigcup_{m=1}^M \bigcup_{g=1}^G \bigcup_{s=1}^S [T_{start}(n_{sg}^m), T_{start}(n_{sg}^m) + T_{dur}(n_{sg}^m)] \|}{\sum_{g=1}^G T_{total}(g)} * 100 \quad (9)$$

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

2.3 Multi-objective vs single objective formulation

When dealing with multi-objective optimization problems, one seeks a set of alternative solutions that can be considered equally good for all the objectives and the objective vectors cannot be simultaneously improved. These solutions are known as the Pareto optimal solutions, that are a set of trade-off optimal solutions (Deb 2014). From a practical point of view, once these alternatives are found, the final solution is then chosen by the decision-maker according to his/her preferences. When these preferences are known in advance, the different objectives can be artificially aggregated into one single objective, most commonly by the use of some weighted-sum function. In this case, a single objective optimization problem is formulated, allowing the use of common optimizers to find a solution and one final solution is found. This approach has the advantages of being usually computationally less expensive and faster, while also simplifying the communication with the decision-maker. On the other hand, if the solution found is not accepted, due to numerous reasons such as the use of inefficient weights, the wrong formulation of the problem due to lack of information prior to the optimization, the optimizer should be run again from the beginning (Fonseca and Fleming 1995). Moreover, weighted-sum methods might work well in convex Pareto-fronts, but when the front is concave there may exist solutions that cannot be reached with this method (Fonseca and Fleming 1995).

In our specific problem, the form of the Pareto-front is not known, therefore weighted-sum method might not always work. In addition, giving more alternative schedules to the operators is a great advantage, as often the communication requirements might change. Regarding the specific problem with the four objectives described in this section, the multi-objective formulation is as follows

$$\text{maximize } F(x) = [Fit_{AW}(x), Fit_{CS}(x), Fit_{TR}(x), Fit_{GU}(x)]^T \quad (10)$$

where x is a vector of the decision variables, as defined above. The 4 objectives are optimized simultaneously, providing a solution of a Pareto-front.

The weighted-sum function is formulated as follows:

$$Fit_{Total}(x) = w_{AW} * Fit_{AW}(x) + w_{TR} * Fit_{TR}(x) + w_{CS} * Fit_{CS}(x) + w_{GU} * Fit_{GU}(x) \quad (11)$$

The 4 objectives are aggregated in one, by the use of specific weights, providing one single solution after the optimization. In the reference paper (Xhafa et al. 2013) the following weights were used:

$$Fit_{Total}(x) = 1.5 * Fit_{AW}(x) + Fit_{TR}(x) + 0.1 * Fit_{CS}(x) + 0.01 * Fit_{GU}(x) \quad (12)$$

The same weights are used in the implementation of the weighted-sum approach of this paper.

3 Selected multi-objective evolutionary algorithms and their implementation

Six different multi-objective EAs are selected and tested, where three of them are Pareto-based: the NSGA-II, SPEA2, and GDE3, one indicator-based: the IBEA, one decomposition-based: the MOEA/D and finally the NSGA-III, which is a modified version of the NSGA-II to perform better to many-objective problems. The last algorithm is Pareto and reference-based. In this section, we briefly describe the above mentioned MOEAs and their implementation.

3.1 Description of selected algorithms

The Non-dominated sorting genetic algorithm-II (NSGA-II) (Deb et al. 2000) is an algorithm for solving non-convex and non-smooth single and multi-objective optimization problems. Its main features include a non-dominated sorting procedure, to sort all individuals according to the level of their non-domination; an elitism, to store all non-dominated solutions, and to enhance convergence properties; a crowding distance mechanism, to guarantee diversity and spread of solutions; consideration of constraints through dominance without the use of penalty functions. During the execution, the population is ordered into a hierarchy of non-dominated Pareto fronts. Also, the crowding distance between solutions of each front is calculated. The crossover and mutation functions are implemented as classical operators of the

genetic algorithm (GA). The solutions of the population are distinguished according to the rank of the front and distance within the front.

The NSGA-III follows the NSGA-II framework (Deb and Jain 2013) with the main difference in the selection mechanism, so as to improve the ability of the algorithm to solve many-objective problems. Crowding distance of the NSGA-II is converted to a selection based on reference points well-distributed in the search space of the objectives. In this way, the diversity of the population is kept and helps the algorithm to perform well in different scaled objectives.

The Strength Pareto evolutionary algorithm 2 (SPEA2) (Zitzler et al. 2001) is a multi-objective algorithm that uses the elitism approach during the evolution process. Each individual is assigned a raw fitness, calculated upon the strength value of solutions who dominate it. To differentiate the individuals having identical raw fitness values, additional density information is calculated. The SPEA2 calculates raw fitness as the sum of the strength values of the solutions that dominate a given candidate. Here, the strength is the number of solutions that a given solution dominates. The density of the Pareto front area is estimated upon the Euclidean distance of the objective values between a given solution and the nearest neighbors of the solution. It fills the archive population with the candidate solutions in order of their fitness, and the most similar solutions are removed from the archive population. For the selection of parents some classical method, such as binary tournament selection or random selection, is used. The crossover and mutation operators are performed as in the standard GA.

The GDE3 is the third version of the generalized Differential Evolution algorithm proposed in Kukkonen and Lampinen (2005). The algorithm is based on Differential Evolution, extended in such a way that can handle multiple objectives and constraints. The selection is based on crowdeness and non-dominance of the solutions (Kukkonen and Deb 2006), similar to that of the NSGA-II.

The Indicator-based evolutionary algorithm (IBEA) (Zitzler and Künzli 2004) is a multi-objective version of the GA, where the selection process is based on quality indicators. Here, the indicator function assigns each Pareto-set approximation a real value that reflects its quality. Then, the optimization goal is the identification of a Pareto-set that minimizes the indicator function. The indicator concept does not require any additional diversity-preservation mechanisms. It was shown by the authors (Zitzler and Künzli 2004) that an indicator-based search can yield results that are superior to some of the other similar algorithms, such as the SPEA2 and NSGA-II. In a basic version of the IBEA, it performs binary tournaments for the selection of individuals to undergo recombination. Next, it iteratively removes the worst individual from the population and updates the fitness values of the remaining individuals. There are a few different variants of the IBEA that are commonly used in practice. Two of the most commonly used variants are the $IBEA_{HD}$ and $IBEA_{\epsilon+}$ (Wagner et al. 2007). $IBEA_{HD}$ uses hypervolume difference indicator (I_{HD}) while $IBEA_{\epsilon+}$ compares individuals using $\epsilon - indicator$ ($I_{\epsilon+}$). In practice, the $IBEA_{\epsilon+}$ often achieves better performance as shown in Tanabe and Oyama (2017). In this study, the $IBEA_{\epsilon+}$ was used.

The MOEA/D refers to the multi-objective evolutionary algorithm based on decomposition (Zhang and Li 2007). This algorithm decomposes a multi-objective

problem to a number of smaller subproblems and optimizes them at the same time. The MOEA/D has two main features: local mating and local replacement. Local mating means that the selected parents are from some neighboring weight vectors, while local replacement means that the generated offspring is compared with solutions among some neighboring weight vectors.

3.2 Implementation of algorithm functions

Algorithms described in Sect. 3 rely on the appropriate problem encoding and operators that introduce small modifications to the chromosomes with the goal of producing better offspring. These operators have to be selected before the start of the optimization and are crucial for obtaining accurate results and fast convergence. This section describes the process of generating the initial population, the encoding of the generated population, and methods for generating new offspring.

3.2.1 Chromosome encoding

Schedules that are to be optimized can be encoded as a chromosome in multiple ways. Paper (Xhafa and Ip 2019) splits the schedule encoding into two separate chromosomes. The first chromosome encodes only the windows of communication for a certain satellite without defining a concrete ground station. The second chromosome represents a list of concrete ground stations with which communication should be established. Both chromosomes together form a valid schedule. Our approach only used one chromosome and encodes a schedule as a list of communication events. Each event consists of five binary variables that can be changed during optimization. The first bit marked as *I* indicates if this event is taken into account when the schedule is evaluated. If this bit is set to 0, the event will be omitted. This approach allows for the algorithms to tune the length of the schedule and to ignore events that are not important. The next four values are integers encoded as binary values that represent spacecraft number, ground station number, the start of the communication window, and its duration; see Fig. 2 for details. A maximal possible schedule length must be set in advance.

3.2.2 Population initialization strategy

The construction of the initial population has a large impact on the performance of certain algorithms (Benecke 2020; Diaz-Gomez et al. 2007). In our case, the initial population is generated by randomly sampling values from specific ranges. Value *I* indicates if the event is used for schedule construction and is assigned random Boolean value. Spacecrafts and ground stations are integers, randomly sampled from a uniform distribution in the range from 0 to the number of spacecraft/ground stations. The last two values of each event from a chromosome are the start and duration time of that event. These values are sampled from the range from 0 up to the maximal time in which the schedule has to be executed. The only additional

Fig. 2 A chromosome-encoded schedule as a list of communication events (left) and actual schedule extracted from a given chromosome (right). Grayed rows denote the omitted event (i.e., $I=0$)

constraint that is put on the last two values is that their sum has to be lower than the number of days for the schedule to be executed.

3.2.3 Crossover operator

To pass information from parents to their offspring, HUX crossover (Bilbao and Alba 2010) was selected to create new individuals for the scheduling problem. HUX crossover (see Fig. 3) creates a new offspring from two randomly selected parents in the current population. Communication event i in offspring's chromosome is randomly selected event at position i in one of the parents.

3.2.4 Mutation operator

In our implementation of GA, BitFlip mutation (Chicano et al. 2015) was chosen. This mutation randomly mutates chromosome bits with a certain probability (see Fig. 4). All of the chromosome's values can be mutated, including the first bit, that indicates if the event is valid. Each bit is mutated with a certain predetermined probability.

4 Control parameter setup

The performance of multi-objective evolutionary optimization algorithms is highly dependant on several control parameters that are usually manually selected before the start of the optimization and affect the structure of chromosomes and their evolution over time. In this section, we report how the different control parameter values were selected. More specifically, the values of the maximal chromosome encoding size, the population size, the crossover and mutation probability, and the maximum number of function evaluations (this is used as our termination criteria) are tuned based on some pre-experiments. The final values that were used in all of the reported experiments of the selected control parameters are summarized in Table 3.

Fig. 3 Two parent chromosomes (top) combined using HUX crossover to form a new chromosome (bottom) that inherits some properties from both of them

Fig. 4 Original chromosome (left) and chromosome after application of BitFlip mutation that changed 4 bits (right)

Fig. 5 The maximal size of the chromosome is determined by the schedule lengths that are produced at the end of pre-experiments. Figure shows the lengths for the exemplary tests for GA after 100,000 function evaluations and three different instance sizes

4.1 Maximal chromosome encoding size

The maximal size of the chromosome and therefore maximal schedule size is one of the most important parameters. If the maximal chromosome size is too small, it might be impossible to encode the optimal schedule, while too large chromosomes might slow down the computation and increase the complexity of the search space. During optimization, events, where indicator variable I is set to 0, are ignored and do not affect the value of objectives. Figures 5 and 6 show the distribution of schedule lengths for problems of different size after optimization is completed and all events where bit I is set to 0 are omitted. For single objective results, the majority of the optimal schedules contain only between 50 and 100 events. Results are also similar for the multi-objective optimization where the majority of the final schedule lengths are between 30 and 130 with slight differences between algorithms. The maximal schedule length was therefore limited to 150 events since the chromosome of that size has enough variables to construct good schedules anywhere on the Pareto front without a large number of unused variables.

4.2 Population size

Population size is one of the key components of optimization algorithms (Mora-Melià et al. 2017). For a single objective population size of 30 was used as in the reference paper (Xhafa et al. 2013). For multi-objective cases, a larger population is required. A population of size 100 was selected since this produces a diverse enough pool of individuals to form a good Pareto front without a lot of overhead computation (see Fig. 7 for details). To make a fair comparison, we used the same population size for all the algorithms. However, for the NSGA-III population size is slightly different (104 instead of 100). The effective population of the NSGA-III cannot be arbitrarily selected in advance as for the other algorithms. The NSGA-III uses a predefined set of reference points distributed

Fig. 6 The maximal size of the chromosome is determined by the schedule lengths that are produced at the end of pre-experiments. Figure shows the lengths for the exemplary tests for multi-objective algorithms after 250,000 function evaluations and three different instance sizes

Table 3 Selected control parameters that are used in all of the reported results

	Single objective	Multi-objective
Maximal schedule length	150	150
HUX crossover rate	0.3	0.3
BitFlip mutation rate	0.001	0.001
Population size	30	100*
Number of evaluations	100,000	250,000

For the maximal schedule length, crossover probability and mutation probability values are the same for single and multi-objective approaches, while for population size and the number of evaluations the values are different for single and multi-objective approach. (*For the NSGA-III the population size was 104)

Fig. 7 The influence of the population size on the optimization convergence. The population size of 100 gives a reasonable trade-off between convergence speed and the number of fitness evaluations. Each line represents a mean of 10 algorithm runs

(a) Effect of HUX crossover probability on the speed of convergence. **(b)** Effect of the BitFlip mutation probability on the speed of convergence.

Fig. 8 The influence of **a** crossover and **b** mutation probability on the optimization convergence. Each line represents a mean of 10 algorithm runs

in a certain structure in the search space to ensure diversity in all the objective axes. For more information the reader can refer to Deb and Jain (2013).

4.3 Crossover and mutation probability

A HUX crossover probability of 0.3 and a mutation probability of 0.001 was selected both for the single and all of the multi-objective algorithms. The parameter values for the MOEAs were based on the single objective GA tuning. In Fig. 8, the influence of different crossover and mutation parameter values on the convergence of the GA is shown.

Fig. 9 Median convergence of weighted-sum objective function-number of function evaluations for three cases of different size (small/medium/large). The red vertical line corresponds to the number of function evaluations used as a termination criteria for the GA

Fig. 10 Median convergence of HV-function evaluations for all the algorithms for three cases of different size (small, medium, large). The red vertical line corresponds to the number of function evaluations used as termination criteria

4.4 Termination criteria

In optimization with evolutionary algorithms, one has to select how and when the algorithm stops (i.e., what are the termination criteria), e.g., the number of evaluations, the number of population-generations, the relative improvement, etc. This is especially important in problems where the optimal solution is not known in advance or in problems, like the one in this study, that are computationally demanding. Here, the maximal number of function evaluations was used as a stopping criterion. To determine the maximal number of function evaluations, pre-experiments were conducted. For each size, the first instance case was run 30 times, where single objective optimization was terminated after 200,000 function evaluations while multi-objective optimization ran

Table 4 The total mean, standard deviation and median of the weighted objective for the GA after 30 runs for each case

	Mean Fit_{Total}	Std Fit_{Total}	Median Fit_{Total}	Median Fit_{AW}	Median Fit_{CS}	Median Fit_{TR}	Median Fit_{GU}
I_S_01	239.92	1.83	240.03	87.12	85.05	99.75	99.22
I_S_02	239.17	2.08	239.54	86.56	84.81	99.85	99.20
I_S_03	238.54	2.26	239.24	86.16	84.52	99.85	99.39
I_S_04	238.98	1.84	239.11	86.44	84.68	99.85	99.20
I_S_05	238.91	1.94	239.21	86.33	84.93	99.92	99.42
I_M_01	231.15	2.08	231.32	81.46	83.95	99.60	96.61
I_M_02	229.73	2.64	229.33	80.44	83.17	99.77	97.95
I_M_03	231.15	2.34	230.93	81.42	83.96	99.65	97.21
I_M_04	231.10	2.53	231.77	81.42	84.13	99.58	97.02
I_M_05	230.18	2.44	230.24	80.86	83.40	99.57	98.07
I_L_01	222.76	2.94	223.33	75.99	83.07	99.52	94.78
I_L_02	222.76	2.94	223.33	75.99	83.07	99.52	94.78
I_L_03	222.09	2.44	222.51	75.50	82.21	99.67	95.39
I_L_04	222.51	2.23	222.76	75.90	82.17	99.49	95.33
I_L_05	222.49	2.84	222.90	75.86	82.44	99.50	95.01

Also, the medians of each objective are reported

for 400,000 function evaluations. We observed, as shown in Figs. 9, 10, that on average single objective algorithm converges after 100,000 function evaluations, while most of the multi-objective algorithms converge after 250,000 function evaluations. Based on these observations, a maximal number of 100,000 (see Fig. 9) and 250,000 (see Fig. 10) function evaluations were used for single and multi-objective algorithms respectively. This termination condition is also marked as a red line in all of the figures that indicates the relative convergence.

5 Results and discussion

There are 16 benchmark instances for each size that can be found in Xhafa and Ip (2019). Due to the similarities of these instances within the same dimensionality of the scheduling problem, and the computational expense, we decided to test only five instances of each size. All algorithms have been independently run for 30 times for each test instance on an Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz, 1 TB of

Fig. 11 Boxplots of hypervolume for each size and instance of the 30 runs of the 6 algorithms

RAM, and the Fedora operating system. The algorithms are implemented in Python 3.7 (Petelin 2019) without any parallelization. Each algorithm run on average for 5-6 hours to converge. Our implementation of the objective functions, algorithm functions and the related code can be found at Petelin (2019).

In Table 4 the mean, standard deviation and median weighted objective together with the medians for individual parts of objective function for our implementation of the GA after 30 runs for each instance case are reported. These are also the solutions used for the comparison to the multi-objective results.

Table 5 The mean, standard deviation, and median of hypervolume achieved by different multi-objective algorithms after 30 runs for each case

	NSGA-II			NSGA-III			SPEA2		
	Mean	Std	Median	Mean	Std	Median	Mean	Std	Median
I_S_01	0.649	0.027	0.643	0.696	0.040	0.689	0.793	0.023	0.795
I_S_02	0.645	0.021	0.646	0.720	0.053	0.731	0.793	0.023	0.793
I_S_03	0.652	0.028	0.646	0.700	0.046	0.698	0.788	0.024	0.791
I_S_04	0.662	0.027	0.662	0.711	0.044	0.716	0.795	0.024	0.797
I_S_05	0.654	0.026	0.655	0.705	0.042	0.697	0.780	0.026	0.781
I_M_01	0.476	0.036	0.476	0.486	0.065	0.486	0.578	0.048	0.575
I_M_02	0.477	0.030	0.476	0.478	0.051	0.476	0.573	0.047	0.581
I_M_03	0.484	0.022	0.485	0.489	0.059	0.472	0.575	0.040	0.579
I_M_04	0.493	0.029	0.490	0.470	0.053	0.467	0.570	0.060	0.564
I_M_05	0.490	0.026	0.491	0.476	0.044	0.467	0.572	0.042	0.574
I_L_01	0.387	0.025	0.392	0.337	0.038	0.332	0.418	0.038	0.421
I_L_02	0.380	0.026	0.381	0.322	0.035	0.337	0.414	0.040	0.409
I_L_03	0.380	0.025	0.378	0.324	0.042	0.323	0.432	0.039	0.429
I_L_04	0.385	0.024	0.388	0.332	0.039	0.325	0.431	0.038	0.431
I_L_05	0.385	0.021	0.388	0.329	0.044	0.334	0.419	0.034	0.410
	GDE3			IBEA			MOEA/D		
	Mean	Std	Median	Mean	Std	Median	Mean	Std	Median
I_S_01	0.661	0.073	0.676	0.754	0.030	0.759	0.841	0.016	0.842
I_S_02	0.647	0.074	0.652	0.743	0.024	0.743	0.841	0.017	0.840
I_S_03	0.654	0.069	0.668	0.740	0.026	0.742	0.834	0.014	0.833
I_S_04	0.657	0.059	0.677	0.743	0.033	0.741	0.837	0.015	0.836
I_S_05	0.661	0.079	0.683	0.754	0.030	0.756	0.833	0.019	0.829
I_M_01	0.384	0.047	0.385	0.591	0.036	0.591	0.731	0.018	0.731
I_M_02	0.370	0.039	0.364	0.574	0.042	0.572	0.727	0.021	0.723
I_M_03	0.386	0.033	0.378	0.601	0.032	0.600	0.735	0.021	0.741
I_M_04	0.375	0.043	0.369	0.581	0.041	0.584	0.725	0.021	0.725
I_M_05	0.387	0.061	0.378	0.575	0.035	0.565	0.718	0.022	0.721
I_L_01	0.289	0.028	0.290	0.461	0.038	0.464	0.607	0.026	0.606
I_L_02	0.283	0.026	0.280	0.449	0.035	0.447	0.615	0.025	0.619
I_L_03	0.278	0.032	0.281	0.458	0.038	0.464	0.623	0.028	0.621
I_L_04	0.285	0.023	0.288	0.471	0.029	0.468	0.609	0.030	0.615
I_L_05	0.277	0.029	0.277	0.457	0.044	0.454	0.608	0.033	0.607

The best results in terms of the hypervolume value are shown in bold

Table 6 Three different ratios

	NSGA-II			NSGA-III			SPEA2		
	Ratio1	Ratio2	Ratio3	Ratio1	Ratio2	Ratio3	Ratio1	Ratio2	Ratio3
I_S_01	1.000	0.448	0.448	1.000	0.326	0.326	1.000	0.765	0.765
I_S_02	1.000	0.462	0.462	1.000	0.402	0.402	1.000	0.740	0.740
I_S_03	1.000	0.465	0.465	1.000	0.359	0.359	1.000	0.739	0.739
I_S_04	1.000	0.490	0.490	1.000	0.378	0.378	1.000	0.769	0.769
I_S_05	1.000	0.468	0.468	1.000	0.376	0.376	1.000	0.716	0.716
I_M_01	1.000	0.449	0.449	1.000	0.207	0.207	1.000	0.541	0.541
I_M_02	1.000	0.439	0.439	1.000	0.164	0.164	1.000	0.473	0.473
I_M_03	1.000	0.449	0.449	1.000	0.184	0.184	1.000	0.496	0.496
I_M_04	1.000	0.446	0.446	1.000	0.153	0.153	1.000	0.483	0.483
I_M_05	1.000	0.429	0.429	1.000	0.178	0.178	1.000	0.472	0.472
I_L_01	1.000	0.463	0.463	1.000	0.162	0.163	1.000	0.328	0.328
I_L_02	1.000	0.447	0.447	1.000	0.220	0.220	1.000	0.370	0.370
I_L_03	1.000	0.460	0.460	1.000	0.210	0.210	1.000	0.344	0.344
I_L_04	1.000	0.497	0.497	1.000	0.230	0.230	1.000	0.444	0.444
I_L_05	1.000	0.466	0.466	1.000	0.235	0.235	1.000	0.377	0.377
	GDE3			IBEA			MOEA/D		
	Ratio1	Ratio2	Ratio3	Ratio1	Ratio2	Ratio3	Ratio1	Ratio2	Ratio3
I_S_01	1.000	0.452	0.452	0.601	0.554	0.919	0.871	0.649	0.747
I_S_02	1.000	0.419	0.419	0.636	0.557	0.878	0.868	0.654	0.754
I_S_03	1.000	0.426	0.426	0.612	0.548	0.902	0.861	0.642	0.746
I_S_04	1.000	0.434	0.434	0.576	0.519	0.905	0.875	0.666	0.761
I_S_05	1.000	0.456	0.456	0.632	0.592	0.933	0.883	0.658	0.745
I_M_01	1.000	0.298	0.298	0.600	0.518	0.864	0.908	0.577	0.636
I_M_02	1.000	0.220	0.220	0.596	0.429	0.735	0.902	0.611	0.677
I_M_03	1.000	0.231	0.231	0.581	0.382	0.653	0.898	0.590	0.658
I_M_04	1.000	0.243	0.243	0.593	0.434	0.731	0.902	0.540	0.599
I_M_05	1.000	0.225	0.225	0.608	0.394	0.643	0.901	0.556	0.617
I_L_01	1.000	0.216	0.216	0.598	0.418	0.694	0.911	0.538	0.591
I_L_02	1.000	0.234	0.234	0.579	0.403	0.687	0.911	0.521	0.572
I_L_03	1.000	0.243	0.243	0.556	0.399	0.715	0.908	0.551	0.606
I_L_04	1.000	0.277	0.277	0.558	0.457	0.819	0.910	0.534	0.586
I_L_05	1.000	0.254	0.254	0.578	0.428	0.742	0.903	0.510	0.564

Each value is computed as the mean of 30 runs. *Ratio1* indicates the ratio of the final population of the MOEAs that is on the approximate Pareto front ($\frac{\text{Non-dom}}{\text{Population}}$). *Ratio2* is the ratio of the final population of the MOEAs that is not dominated by the weighted-sum solution ($\frac{\text{Non-dom by single}}{\text{Population}}$). *Ratio3* is the ratio of the final non-dominated solutions of the population of the MOEAs that are also non-dominated by the weighted-sum solution ($\frac{\text{Non-dom by single}}{\text{Non-dom}}$). The best results in terms of ratio value are bolded

Table 7 p-value of Wilcoxon signed rank test of comparison between MOEA/D and other algorithms, in terms of hypervolume indicator over the 30 runs

	NSGA-II	NSGA-III	SPEA2	GDE3	IBEA
I_S_01	1.734e-06	1.734e-06	2.353e-06	1.734e-06	1.734e-06
I_S_02	1.734e-06	1.734e-06	2.878e-06	1.734e-06	1.734e-06
I_S_03	1.734e-06	1.734e-06	2.353e-06	1.734e-06	1.734e-06
I_S_04	1.734e-06	1.734e-06	4.729e-06	1.734e-06	1.734e-06
I_S_05	1.734e-06	1.734e-06	6.983e-06	1.734e-06	1.734e-06
I_M_01	1.734e-06	1.734e-06	1.734e-06	1.734e-06	1.734e-06
I_M_02	1.734e-06	1.734e-06	1.734e-06	1.734e-06	1.734e-06
I_M_03	1.734e-06	1.734e-06	1.734e-06	1.734e-06	1.734e-06
I_M_04	1.734e-06	1.734e-06	1.734e-06	1.734e-06	1.734e-06
I_M_05	1.734e-06	1.734e-06	1.734e-06	1.734e-06	1.734e-06
I_L_01	1.734e-06	1.734e-06	1.734e-06	1.734e-06	1.734e-06
I_L_02	1.734e-06	1.734e-06	1.734e-06	1.734e-06	1.734e-06
I_L_03	1.734e-06	1.734e-06	1.734e-06	1.734e-06	1.734e-06
I_L_04	1.734e-06	1.734e-06	1.734e-06	1.734e-06	1.734e-06
I_L_05	1.734e-06	1.734e-06	1.734e-06	1.734e-06	1.734e-06

For the multi-objective optimization, we report the hypervolume (Zitzler and Thiele 1999) as a performance metric. Hypervolume is one of the most popular performance metrics as it can simultaneously measure the convergence and the diversity of the multi-objective solutions.

In Table 5 the mean, standard deviation, and median of the hypervolume of 30 runs for the five instances of each size for all the algorithms are reported. These values are illustrated in Fig. 11 as a boxplot. It is easily seen that the MOEA/D not only achieves the highest hypervolume values in all cases but has the most robust performance as well. It is worth mentioning, that the NSGA-II, while not achieving among the highest hypervolume values, performs robustly in all the cases. It is interesting to note that the NSGA-III performs worse than the NSGA-II in many cases in terms of hypervolume values, despite being an algorithm specially designed for many-objectives such as in our problem. This is in line with the paper (Ishibuchi et al. 2016), where the performance of the two algorithms was assessed in a number of problems and the NSGA-III performed better than the NSGA-II in numerical many-objective test problems, but much worse on many-objective knapsack problems. GDE3 performs the worst in terms of hypervolume in all cases. The GDE3 is using DE as a

Table 8 Ranking based on Deep statistical comparison using two-sample Anderson-Darling test with significance level 0.05

	NSGA-II	NSGA-III	SPEA2	GDE3	IBEA	MOEA/D
I_S_01	6.0	4.5	2.0	4.5	3.0	1.0
I_S_02	5.5	3.5	2.0	5.5	3.5	1.0
I_S_03	6.0	4.0	2.0	5.0	3.0	1.0
I_S_04	5.5	3.5	2.0	5.5	3.5	1.0
I_S_05	6.0	4.0	2.0	5.0	3.0	1.0
I_M_01	4.5	4.5	2.5	6.0	2.5	1.0
I_M_02	4.5	4.5	2.5	6.0	2.5	1.0
I_M_03	4.5	4.5	2.5	6.0	2.5	1.0
I_M_04	4.5	4.5	2.5	6.0	2.5	1.0
I_M_05	4.5	4.5	2.5	6.0	2.5	1.0
I_L_01	4.0	5.0	3.0	6.0	2.0	1.0
I_L_02	4.0	5.0	3.0	6.0	2.0	1.0
I_L_03	4.0	5.0	2.5	6.0	2.5	1.0
I_L_04	4.0	5.0	3.0	6.0	2.0	1.0
I_L_05	4.0	5.0	3.0	6.0	2.0	1.0
	7.022e-8	5.822e-7	1.590e-2	1.495e-11	1.590e-2	

The last line presents the results of the Hochberg post-hoc test showing the p-values below 0.05 and confirming that the MOEA/D is significantly better than the other algorithms

method of selection, mutation and recombination, making it more appropriate for real-valued problems (Quresh et al. 2019) and less for scheduling and discrete problems as the studied one. Each algorithm seems to have similar performance for different cases of each size, except for the IBEA and case I_M_03, where as seen from the boxplot, it appears to have a less robust performance, expressed in wider distribution of hypervolume values (Table 6).

The Wilcoxon signed-rank test (Derrac et al. 2011) was carried out. In Table 7 the p-values of the Wilcoxon signed-rank test are reported, where MOEA/D is used as the control algorithm against the other algorithms. The p-values were computed using python's scipy library¹. It is shown that the MOEA/D outperforms the other algorithms in terms of hypervolume value with a statistical difference at a 0.05 significance level.

Additional statistical evaluation was performed by the DSCTool (Eftimov et al. 2020) that ranks optimization algorithms by comparing distributions of obtained solutions for a problem. Since we are dealing with approximation sets, we used hypervolume as an approximation set quality indicator that was used for statistical analysis. The applied approach for statistical comparison of meta-heuristic multi-objective optimization algorithms is further described in Eftimov and Korošec (2021). Table 8 shows the rankings obtained by the DSC approach (Eftimov et al.

¹ <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html>

2017) using two-sample Anderson-Darling test with significance level 0.05, where the MOEA/D is ranked first in all instances of all sizes. Further omnibus test, using Friedman test with significance level of 0.05, shows that there are significant differences between the compared algorithms by returning p-value $5.59\text{e}-13$. To check where these differences reside, we applied post-hoc test with the MOEA/D as the control algorithm. The last line of the Table 8 consists of the Hochberg test results, where all p-values are below 0.05, which confirms that MOEA/D significantly outperforms other compared algorithms.

From the rankings in Table 8, for example, we observe that, according to Anderson-Darling test with significance level of 0.05, there is significant difference among the algorithms NSGA-II and NSGA-III for all the small and large instances, considering the distribution of their solutions. In the small size, the NSGA-III outperforms the NSGA-II, while the opposite applies to the large scale. For the medium size, according to Anderson-Darling test with significance level of 0.05, there is no significant difference among the two algorithms, considering the distribution of their solutions.

In Fig. 10 the median convergence of the algorithms in terms of hypervolume and function evaluations is shown for the first instance of each size. The MOEA/D converges significantly faster than the rest of the algorithms. In these particular cases, for example, it reaches very high hypervolume value in less than 100,000 function evaluations, making its performance comparable to the weighted-sum single objective approach in terms of computational expense. That means that with the same number of function evaluations, the algorithm gives multiple equally good solutions to the decision-maker instead of a single one. We note that as appeared in Fig. 10b the hypervolume values do not always increase in a monotonic fashion over iterations. Hypervolume values may decrease when solutions that cover sparse areas are favored even if they are away from the reference Pareto front over solutions in the same Pareto front rank, due to the diversity factors of the algorithms (Ben Abdesslem et al. 2016). In the case of the NSGA-II this happens due to the crowding distance factor, calling the problem “hypervolume degeneration problem” (Peng and Tang 2011). We suspect the same applies for the NSGA-III which uses a set of uniformly distributed direction vectors for diversity maintenance.

The superiority of MOEA/D can be better seen in Table 6. In order to fairly compare the weighted-sum solution with the approximate Pareto fronts reached by the different MOEAs, three ratios were computed. *Ratio1* refers to the ratio of the final population that is on the approximate Pareto front for each algorithm. *Ratio2* is the ratio of the final population that is non-dominated of the weighted-sum solution. Since the number of final points of the Pareto front is different for each algorithm, another ratio was also computed for a more fair comparison. *Ratio3* is the percentage of the final non-dominated solutions (solutions that are part of the approximated Pareto front) of each algorithm that are

also non-dominated by the weighted-sum solution. The ratios are computed as the mean of 30 runs. In terms of *Ratio3*, meaning how many of the final non-dominated solutions found by the algorithm are equally good as the weighted-sum solution, the IBEA seems to perform the best in all cases except I_M_03, where the MOEA/D has better values. In terms of *Ratio2*, for the small cases, the SPEA2 has the best performance, while for medium and large the MOEA/D reaches the best values. An example of what we measure can be seen in Figs. 12, 13, 14, where an example of the scatter plot of approximate Pareto-front pairs in the objective space is shown (this refers to the first instances of each size and to one run of them). The blue color points are non-dominated by the weighted-sum solution (red triangle), while orange points are the rest found by each algorithm. It can be seen, for example for the small instance, that almost half of the points are non-dominated in the NSGA-II and almost all the points are non-dominated by the red triangle in the IBEA, corresponding to the *Ratio3* values of Table 6 0.448 and 0.919, respectively.

In Figs. 15, 16 and 17 for the same instances, the combined pair plots of all the approximate Pareto fronts found by the algorithms are plotted, to demonstrate the differences of the fronts each algorithm manages to reach. Here, each color corresponds to another algorithm. In addition, for the same instances and runs, the parallel coordinate plots of the final solutions are shown in Figs. 18, 19, 20. In the same plots, the median of the weighted-sum solution of each objective is plotted with red line. The IBEA seems to find a range of solutions close to the weighted-sum one in all the cases, while the MOEA/D finds more diverse solutions, but also solutions close to the weighted-sum, contrary to the rest of the algorithms. Indicator based algorithms might bias the search to a specific region and “prefer” convergence to diversity, as it is noted in Li et al. (2016). The existence of decomposition of operations of MOEA/D, optimizing in parallel the multiple objectives gives a great advantage to obtain a dispersion of the final solution and forms the most adequate algorithm for the studied problem, especially for the larger scale instances.

It should be stressed, that in general, all the algorithms manage to find solutions that can be regarded as equally good from a multi-objective point of view

Fig. 12 Scatter plot of the final approximate Pareto front objective pairs for I_S_01: **a** NSGA-II, **b** NSGA-III, **c** SPEA2, **d** GDE3, **e** IBEA and **f** MOEA/D. The red triangle indicates the median solution of the weighted-sum single objective approach obtained by the GA. Orange and blue dots are the final Pareto front obtained by each algorithm, where the blue dots are the non-dominated solutions by the weighted-sum solution

(non-dominated) as the single solution found by the weighted-sum. Meaning, even for the worst-performing algorithm in terms of hypervolume, 15–40 percent of its final solutions are competitive to just the single one on one run. From the decision-maker point of view for this problem, it is interesting to see that two of the objectives when violated create infeasible schedules (Fit_{AW} and Fit_{CS}). Some of the algorithms managed to find solutions that solve these two objectives to optimality or near optimality (i.e., 100) simultaneously, giving to the decision-maker the opportunity to find solutions like this, which would be impossible with an incorrect combination of weights. From this aspect, the MOEA/D again performs the best, finding solutions near 100 for these two objectives in all three first instances of each size (I_S_01 , I_M_01 , I_L_01), while for the small case also the NSGA-II finds a similar solution as can be seen in Figs. 18a, f, 19 and 20f.

6 Conclusion and future work

In this paper, we have evaluated and analyzed the performance of six different multi-objective algorithms on a set of benchmark instances of the Ground Station Scheduling problem. Moreover, the scalability of these algorithms was tested, as the benchmarks were of different problem dimensionality. The GSS problem is of great importance, as its solution allocates optimally the communication of many satellites to a limited number of ground stations. The problem has, in its general form, multiple objectives, while in this study four of them were taken into account. The specific formulation and benchmark instances were previously approached in the literature with the weighted-sum method, providing one solution at the end of the optimization. In this paper, a set of Pareto optimal solutions are evaluated for each algorithm. We have performed an experimental evaluation using a benchmark of three different sizes and five instances of each size. The hypervolume indicator has been used to evaluate the quality of the approximated fronts. Moreover, we compared the performance of the algorithms with the state-of-the-art weighted-sum solution. This comparison was done by identifying the solutions of all multi-objective algorithms that are not dominated by the weighted-sum solution, and used the relevant ratios as a performance metric. The results have shown interesting conclusions. In general, the decomposition-based multi-objective algorithm outperforms the rest of the algorithms in all the cases for the reported hypervolume values. The MOEA/D also converges faster and shows good performance when compared to the weighted-sum solution. Moreover, the IBEA shows good performance when compared to the number of final non-dominated solutions found, which are also non-dominated by the weighted-sum solution. In general, all the algorithms manage to find solutions that can be regarded as equally good with the weighted-sum solution in one single run, giving more alternatives to the decision-makers. We have also gone one step further by spotting solutions that solve to optimality two of the objectives of the

Fig. 13 Scatter plot of the final approximate Pareto front objective pairs for I_M_01: **a** NSGA-II, **b** NSGA-III, **c** SPEA2, **d** GDE3, **e** IBEA and **f** MOEA/D. The red triangle indicates the median solution of the weighted-sum single objective approach obtained by the GA. Orange and blue dots are the final Pareto front obtained by each algorithm, where the blue dots are the non-dominated solutions by the weighted-sum solution

problem, that give infeasible solutions when not optimal, meaning that they provide a schedule that can be regarded as final.

A future line of research would be to take into consideration the hierarchy in the objective functions by giving more importance to the objectives that correspond to constraints (i.e., Fit_{AW} and Fit_{CS}) and formulate the problem as a multi-objective bilevel optimization problem. Another interesting step would be to test the NSGA-III with different reference point distribution, such as to have more points in the most important or difficult objective functions and alleviate the Hypervolume Degeneration problem. Last but not least, using adaptive control parameters (Antonioni et al. 2021) instead of static ones for the implementation of the MOEAs and their effect to the convergence (Papa 2013) in a real-world application (Papa and Doerr 2020) such as the GSS is a very promising research area.

Acknowledgements The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0098). This work has also received funding from 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).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Fig. 14 Scatter plot of the final approximate Pareto front objective pairs for ILL_01: **a** NSGA-II, **b** NSGA-III, **c** SPEA2, **d** GDE3, **e** IBEA and **f** MOEA/D. The red triangle indicates the median solution of the weighted-sum single objective approach obtained by the GA. Orange and blue dots are the final Pareto front obtained by each algorithm, where the blue dots are the non-dominated solutions by the weighted-sum solution

Fig. 15 Summarized scatter plot of all algorithms of the final approximate Pareto front objective pairs for I_S_01. The red triangle indicates the median solution of the weighted-sum single objective approach obtained by the GA. Other dots are the final Pareto fronts obtained by different algorithms (refer to the legend for color-algorithm pairs)

Fig. 16 Summarized scatter plot of all algorithms of the final approximate Pareto front objective pairs for I_M_01. The red triangle indicates the median solution of the weighted-sum single objective approach obtained by the GA. Other dots are the final Pareto fronts obtained by different algorithms (refer to the legend for color-algorithm pairs)

Fig. 17 Summarized scatter plot of all algorithms of the final approximate Pareto front objective pairs for I_L_01. The red triangle indicates the median solution of the weighted-sum single objective approach obtained by the GA. Other dots are the final Pareto fronts obtained by different algorithms (refer to the legend for color-algorithm pairs)

Fig. 18 Parallel coordinate plot of the approximate Pareto front of I_S_01 cases for each algorithm: **a** NSGA-II, **b** NSGA-III, **c** SPEA2, **d** GDE3, **e** IBEA and **f** MOEA/D. The red line denotes the averaged single objective runs. The color transition from blue to green for lines denoting multi-objective solutions is used just for easier visualization and has no other meaning

Fig. 19 Parallel coordinate plot of the approximate Pareto front of I_M_01 cases for each algorithm: **a** NSGA-II, **b** NSGA-III, **c** SPEA2, **d** GDE3, **e** IBEA and **f** MOEA/D. The red line denotes the averaged single objective runs. The color transition from blue to green for lines denoting multi-objective solutions is used just for easier visualization and has no other meaning

Fig. 20 Parallel coordinate plot of the approximate Pareto front of I_L_01 cases for each algorithm: **a** NSGA-II, **b** NSGA-III, **c** SPEA2, **d** GDE3, **e** IBEA and **f** MOEA/D. The red line denotes the averaged single objective runs. The color transition from blue to green for lines denoting multi-objective solutions is used just for easier visualization and has no other meaning

References

- Antoniou M, Hribar R, Papa G (2021) Parameter control in evolutionary optimisation. In: Vasile M (ed) Optimization under uncertainty with applications to aerospace engineering. Springer, pp 357–385. https://doi.org/10.1007/978-3-030-60166-9_11
- Barbulescu L, Howe AE, Watson JP, Whitley LD (2002) Satellite range scheduling: a comparison of genetic, heuristic and local search. In: International conference on parallel problem solving from nature PPSN VII, pp 611–620. Springer
- Barbulescu L, Watson JP, Whitley LD, Howe AE (2004) Scheduling space-ground communications for the air force satellite control network. *J Sched* 7(1):7–34. <https://doi.org/10.1023/B:JOSH.0000013053.32600.3c>
- Ben Abdesslem R, Nejati S, Briand LC, Stifter T (2016) Testing advanced driver assistance systems using multi-objective search and neural networks. In: Proceedings of the 31st IEEE/ACM international conference on automated software engineering, pp 63–74
- Benecke T (2020) Tracing the impact of the initial population in evolutionary algorithms. Ph.D. thesis, Otto-von-Guericke Universität, Magdeburg, Germany
- Bilbao M, Alba E (2010) CHC and SA applied to wind energy optimization using real data. In: IEEE congress on evolutionary computation, pp 1–8. IEEE. <https://doi.org/10.1109/CEC.2010.5586395>
- Castaing J (2014) Scheduling downloads for multi-satellite, multi-ground station missions. In: proceedings of the 28th annual AIAA/USU conference on small satellites, pp 1–12

- Chicano F, Sutton AM, Whitley LD, Alba E (2015) Fitness probability distribution of bit-flip mutation. *Evol Comput* 23(2):217–248. https://doi.org/10.1162/evco_a_00130
- Deb K (2014) Multi-objective optimization. In: Search methodologies, pp 403–449. Springer. https://doi.org/10.1007/978-1-4614-6940-7_15
- Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Schoenauer M, Deb K, Rudolph G, Yao X, Lutton E, Merelo J, Schwefel HP (eds) Parallel problem solving from nature PPSN VI, lecture notes in computer science, vol 1917, pp 849–858. Springer, Berlin. https://doi.org/10.1007/3-540-45356-3_83
- Deb K, Jain H (2013) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Trans Evol Comput* 18(4):577–601. <https://doi.org/10.1109/TEVC.2013.2281535>
- Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evolut Comput* 1(1):3–18. <https://doi.org/10.1016/j.swevo.2011.02.002>
- Diaz-Gomez PA, Hougen DF (2007) Initial population for genetic algorithms: a metric approach. In: Arabnia HR, Yang JY, Yang MQ (eds) Proceedings of the 2007 international conference on genetic and evolutionary methods, GEM 2007, pp 43–49. CSREA Press
- Eftimov T, Korošec P (2021) Deep statistical comparison for multi-objective stochastic optimization algorithms. *Swarm Evolut Comput* 61:100837. <https://doi.org/10.1016/j.swevo.2020.100837>
- Eftimov T, Korošec P, Koroušić Seljak B (2017) A novel approach to statistical comparison of meta-heuristic stochastic optimization algorithms using deep statistics. *Inf Sci* 417:186–215. <https://doi.org/10.1016/j.ins.2017.07.015>
- Eftimov T, Petelin G, Korošec P (2020) Dsctool: a web-service-based framework for statistical comparison of stochastic optimization algorithms. *Appl Soft Comput* 87:105977. <https://doi.org/10.1016/j.asoc.2019.105977>
- Fonseca CM, Fleming PJ (1995) An overview of evolutionary algorithms in multiobjective optimization. *Evol Comput* 3(1):1–16. <https://doi.org/10.1162/evco.1995.3.1.1>
- Gentile L, Greco C, Minisci E, Bartz-Beielstein T, Vasile M (2019) An optimization approach for designing optimal tracking campaigns for low-resources deep-space missions. In: Proceedings of the 70th international astronomical congress, pp 1–11. IAA & IISL & SGAC, Washington, DC, USA
- Globus A, Crawford J, Lohn J, Pryor A (2003) Scheduling earth observing satellites with evolutionary algorithms. In: International conference on space mission challenges for information technology (SMC-IT), pp 1–11
- Ishibuchi H, Imada R, Setoguchi Y, Nojima Y (2016) Performance comparison of NSGA-II and NSGA-III on various many-objective test problems. In: Proceedings of the 2016 IEEE congress on evolutionary computation (CEC), pp 3045–3052. IEEE
- Kukkonen S, Deb K (2006) A fast and effective method for pruning of non-dominated solutions in many-objective problems. In: Parallel problem solving from nature PPSN IX, pp 553–562. Springer
- Kukkonen S, Lampinen J (2005) GDE3: the third evolution step of generalized differential evolution. In: Proceedings of the 2005 IEEE congress on evolutionary computation, vol 1, pp 443–450. IEEE. <https://doi.org/10.1109/CEC.2005.1554717>
- Lala A, Kolici V, Xhafa F, Herrero X, Barolli A (2015) On local vs. population-based heuristics for ground station scheduling. In: Proceedings of the 2015 ninth international conference on complex, intelligent, and software intensive systems, pp 267–275. IEEE
- Li B, Tang K, Li J, Yao X (2016) Stochastic ranking algorithm for many-objective optimization based on multiple indicators. *IEEE Trans Evol Comput* 20(6):924–938
- Mora-Melià D, Martínez-Solano FJ, Iglesias-Rey PL, Gutiérrez-Bahamondes JH (2017) Population size influence on the efficiency of evolutionary algorithms to design water networks. *Proced Eng* 186:341–348. <https://doi.org/10.1016/j.proeng.2017.03.209>
- NASA: What is a satellite? <https://www.nasa.gov/audience/forstudents/k-4/stories/nasa-knows/what-is-a-satellite-k4.html> (2017). Accessed 13 May 2020
- Papa G (2013) Parameter-less algorithm for evolutionary-based optimization. *Comput Optim Appl* 56(1):209–229. <https://doi.org/10.1007/s10589-013-9565-4>
- Papa G, Doerr C (2020) Dynamic control parameter choices in evolutionary computation: GECCO 2020 tutorial. In: Proceedings of the 2020 genetic and evolutionary computation conference companion, pp 927–956. GECCO '20, Association for Computing Machinery, New York. <https://doi.org/10.1145/3377929.3389876>

- Papa G, Petelin G (2019) Optimisation platform for remote collaboration of different teams. In: Proceedings of the 5th optimisation in space engineering workshop (OSE5), p 1. ESA
- Pemberton JC, Galiber F (2000) A constraint-based approach to satellite scheduling. In: DIMACS workshop on constraint programming and large scale discrete optimization, pp 101–114. American Mathematical Society, USA
- Peng F, Tang K (2011) Alleviate the hypervolume degeneration problem of NSGA-II. In: International conference on neural information processing, pp 425–434. Springer
- Petelin G (2019) Satellite scheduling repository. <https://repo.ijs.si/e7/scheduling/multi-objective-satellites-scheduling>. Accessed 17 July 2020
- Platypus: Platypus library. <https://github.com/Project-Platypus/Platypus> (2015). Accessed 17 July 2020
- Quresh K, Rahnamayan S, He Y, Liscano R (2019) Enhancing lqr controller using optimized real-time system by gde3 and NSGA-II algorithms and comparing with conventional method. In: Proceedings of the 2019 IEEE congress on evolutionary computation (CEC), pp 2074–2081. IEEE
- Tanabe R, Oyama A (2017) Benchmarking MOEAs for multi-and many-objective optimization using an unbounded external archive. In: Proceedings of the genetic and evolutionary computation conference GECCO '17, pp 633–640. Association for Computing Machinery. <https://doi.org/10.1145/3071178.3079192>
- Tangpattanakul P, Jozefowicz N, Lopez P (2012) Multi-objective optimization for selecting and scheduling observations by agile earth observing satellites. In: International conference on parallel problem solving from nature PPSN XII, pp 112–121. Springer
- Tangpattanakul P, Jozefowicz N, Lopez P (2015) A multi-objective local search heuristic for scheduling earth observations taken by an agile satellite. *Eur J Oper Res* 245(2):542–554. <https://doi.org/10.1016/j.ejor.2015.03.011>
- Wagner T, Beume N, Naujoks B (2007) Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: Obayashi S, Deb K, Poloni C, Hiroyasu T, Murata T (eds) Evolutionary multi-criterion optimization. Springer, Berlin, Heidelberg, pp 742–756
- Wang Z, Zhang Z, Chen Y (2019) Multi-objective optimization of satellite-ground time synchronization scheduling problem. In: Proceedings of the 2019 IEEE congress on evolutionary computation (CEC), pp 530–537. IEEE
- Xhafa F, Herrero X, Barolli A, Takizawa M (2013) Using STK toolkit for evaluating a ga base algorithm for ground station scheduling. In: Proceedings of the 2013 seventh international conference on complex, intelligent, and software intensive systems, pp 265–273. IEEE
- Xhafa F, Ip AW (2019) Optimisation problems and resolution methods in satellite scheduling and space-craft operation: a survey. *Enterprise information systems*, pp 1–24. <https://doi.org/10.1080/17517575.2019.1593508>
- Zhai X, Niu X, Tang H, Wu L, Shen Y (2015) Robust satellite scheduling approach for dynamic emergency tasks. *Math Prob Eng* 2015:1–20. <https://doi.org/10.1155/2015/482923>
- Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731. <https://doi.org/10.1109/TEVC.2007.892759>
- Zhang Z, Xing L, Chen Y, Wang P (2016) Evolutionary algorithms for many-objective ground station scheduling problem. In: International conference on bio-inspired computing: theories and applications, pp 265–270. Springer
- Zhao F, Chen Z, Wang J, Zhang C (2017) An improved MOEA/D for multi-objective job shop scheduling problem. *Int J Comput Integr Manuf* 30(6):616–640. <https://doi.org/10.1080/0951192X.2016.1187301>
- Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou KC, Tsahalis DT, Périaux J, Papailiou KD, Fogarty T (eds) Evolutionary methods for design optimization and control with applications to industrial problems. International Center for Numerical Methods in Engineering, Athens, Greece, pp 95–100
- Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search. In: Proceedings of the 8th International Conference on Parallel Problem Solving from Nature PPSN VIII. pp. 832–842. Springer
- Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans Evol Comput* 3(4):257–271. <https://doi.org/10.1109/CEC.2019.8790101>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.