Stochastic satellite tracking with constrained budget via structured-chromosome genetic algorithms

Lorenzo Gentile^{1,2} · Cristian Greco² · Edmondo Minisci² · Thomas Bartz-Beielstein¹ · Massimiliano Vasile²

Received: 24 August 2020 / Revised: 7 September 2021 / Accepted: 28 September 2021 /
Published online: 7 November 2021
© The Author(s) 2021

Abstract

This paper focuses on the scheduling under uncertainty of satellite tracking from a heterogeneous network of ground stations taking into account allocated resources. An optimisation-based approach is employed to efficiently select the optimal tracking schedule that minimises the final estimation uncertainty. Specifically, the scheduling is formulated as a variable-size problem, and a Structured-Chromosome Genetic Algorithm is developed to tackle the mixed-discrete global optimisation. The search algorithm employs genetic operators specifically revised to handle hierarchical search spaces. An orbit determination routine is run within each call to the fitness function to quantify the estimation uncertainty resulting from each candidate tracking schedule. The developed scheduler is tested on the tracking optimisation of a satellite in low Earth orbit, a highly perturbed dynamical regime. The obtained results show that the variable-size variants of Genetic Algorithms always outperform the fixed-size counterparts employed for comparison. In particular, Structured-Chromosome Genetic Algorithm is shown to find significantly better schedules under severely limited budgets.

Keywords Optimisation · Genetic algorithms · Variable-size optimisation · Satellite tracking

✉ Lorenzo Gentile
lorenzgentile1892@gmail.com

Cristian Greco
c.greco@strath.ac.uk

¹ Technische Hochschule Köln, Steinmüllerallee 1, 51643 Gummersbach, Germany

² University of Strathclyde, 75 Montrose Street, G1 1XJ Glasgow, UK

1 Introduction

Tracking operational and non-operational satellites is a crucial task for modern space missions. To deliver increasingly precise services and solid scientific outcomes, operational satellites require multi-source observations to achieve accurate knowledge of their position and velocity. Tracking of non-operational satellites or debris is fundamental for considerate space traffic management and reliable re-entry forecasting.

In the near-Earth environment, the amount of objects has been swiftly growing due to the enlarged range of terrestrial services that space technology provides. In this context, next to traditional satellites, small satellites have gained significant momentum as they represent a cost-efficient solution for Earth observation, communications, and science (Helvajian and Janson 2009; Kopacz et al. 2020). Indeed, small organisations have exploited the enhanced maturity of miniaturised systems and platforms to launch numerous missions in low Earth orbits (Toorian et al. 2008).

Another recent trend is the use of mega-constellations of hundreds to thousands of satellites to achieve global coverage by New Space actors, as SpaceX's Starlink (McDowell 2020) and Planet Labs' Dove (Safyan 2020), or the planned Amazon's Project Kuiper, Telesat LEO, and Boeing's (Rossi et al. 2017; Lewis et al. 2017). The consequence of this fast growth is that near-Earth space traffic is becoming more and more congested, causing a sharp increase in the number of conjunction events. Even in the unrealistic "no new launches" scenario, studies suggest that, if no mitigation action is enforced, the current Low Earth Orbit (LEO) environment has already reached an unstable condition (Liou and Johnson 2008) where collisions will increase noticeably and cause a domino effect (Kessler et al. 2010).

This planned and unplanned traffic surge will pose substantial pressure on current satellite operations to track satellites accurately, analyse more conjunctions, design optimal collision avoidance manoeuvres, and upload commands. Hence, modern solutions for efficient operations, tracking, and scheduling need to be designed to tackle this growing pressure on the ground segment (Muelhaupt et al. 2019). Indeed, while modern operational satellites flying below the medium Earth orbit can exploit global navigation satellite systems to estimate their state accurately, non-operational objects are challenging to track (Mehrholtz et al. 2002). The major issues are the debris' potential small sizes, data latency, data and association uncertainty, and low reflectivity (Xi et al. 2020). Besides, while usually large missions can rely on a dedicated but expensive ground network, e.g. the Deep Space Network (DSN) (Clement and Johnston 2005), smaller or non-collaborative ones often use amateur or external services, which, however, are often characterised by scheduling conflicts, queue times, and external constraints (Kleinschrodt et al. 2017; Kathleen et al. 2018; Ali et al. 2021).

Optimisation-based approaches for operations planning and scheduling of ground stations have been proposed, e.g. using Genetic Algorithms (GA) (Soma et al. 2004; Sun and Xhafa 2011), simulated annealing (Xhafa et al. 2013), mixed

integer programming (Spangelo et al. 2015), a multi-objective bi-level formulation (Antoniou et al. 2020; Petelin et al. 2021), and dynamic programming for ESA's EPS (Damiani et al. 2007). Specific approaches for low-cost stations have been investigated (Schmidt et al. 2008) and compared (Kleinschrodt et al. 2016). Almost all methods in literature neglect non-deterministic effects typical of real-world operational scenarios to avoid the computational burden of uncertainty quantification routines. However, communication with and tracking of space objects in highly nonlinear dynamical environments, e.g. LEO, are complex tasks subject to multiple stochastic sources. Indeed, the state knowledge and its error statistics are only estimated by indirect observations, which are noisy and often biased because of sensor and atmospheric noises. Besides, in particular for space debris, geometrical and dynamical parameters governing the object's motion are uncertain, compromising the ability to propagate uncertainties in time accurately. Hence, there is a timely need for methods accounting for these stochastic sources to enhance the reliability of operations, tracking and communication scheduling.

Hence, this paper proposes a scheduling method for observation from heterogeneous ground stations that accounts for the stochastic sources involved in the tracking process. Therefore, the problem addressed requires the coupling of two major disciplines in aerospace engineering. The outer loop is an optimisation for selecting the optimal tracking schedule. The inner model involves an orbit determination routine for evaluating the estimation accuracy resulting from a candidate schedule. The method's scientific objective is to efficiently select the optimal schedule that reduces the final state estimation uncertainty while respecting allocated resources. From a wider perspective, the proposed scheduling aims at contributing toward an augmented autonomy, efficiency, and safety in the space sector.

Different schedules can be characterised by the use of different ground stations, different orbital passages to track, and different time allocations. Consequently, the tracking scheduling is a variable-size mixed-discrete global optimisation where different candidates generally have a different number of free variables. Dynamically varying search spaces entail very specific challenges which, often, cannot be tackled by simply adapting fixed-size solvers. The most promising class of algorithms to tackle these problems is GA (Pelamatti et al. 2017), which can exploit an appropriated encoding to partially overcome some of the variable-size challenges.

The literature already comprises a optimisation algorithms to cope with variable-size global optimisation mostly applied to space trajectory design (Nyew et al. 2012, 2015; Darani and Abdelkhalik 2017). One relevant example is the *hidden-genes* GA variant introduced in Darani and Abdelkhalik (2017). This method encodes the problem formulation with two associated set of genes, namely the *design genes* and *activation gene*. The firsts represents all the possible design variables that can characterise a chromosome, the second indicates which genes have to be considered when computing the objective and constraint functions. The main drawbacks are that the complete set of genes that can describe a feasible solution has to be set a-priori and that the operation made over the deactivated genes has no effect other than wasting computational resources. Another optimisation algorithms worthwhile to be mentioned is the GA adaptation proposed in Nyew et al. (2012), Nyew et al. (2015). The backbone of this algorithm is the problem formulation by means of hierarchical

multi-level chromosomes structure. In this case the genes are not all flattened at the same level (*string representation*) as in standard GA formulation, but they represent the leaf of a tree hierarchical structure. In respect to the hidden-genes strategy, this approach does not waste any computational resources operating on deactivated genes. Furthermore, such a hierarchical formulation can enhance the exchange of information between candidate solutions. Following in the footsteps of Nyew et al. (2012, 2015), a new interpretation of GA named Structured-Chromosome Genetic Algorithm (SCGA) based on structured-chromosomes has been proposed in Gentile et al. (2019a, b), Greco et al. (2019), Gentile et al. (2020, a). The algorithm was applied to a variety of fields of aerospace engineering.

SCGA was already used to handle tracking optimisation problems. In Gentile et al. (2019a) the application of SCGA for determining the observation schedule of a deep space mission is presented. Notably, this first work indicated how the algorithm's features could constitute a remarkable instrument for small organisations willing to embrace deep-space missions. Moreover, in Gentile et al. (2019b), Greco et al. (2019) SCGA has been used for the tracking of a low Earth satellite. In this paper, rigorous testing of the SCGA when applied to the scheduling of observation under reduced budget constraint is presented and discussed.

The paper has the following structure. The developed optimisation-based scheduling method is presented in Sect. 2. The scheduling formulation and estimation approach are discussed in Sect. 3. The test case description and the analysis of the results are detailed in Sects. 4 and 5, respectively. The final remarks are discussed in Sect. 6.

2 SCGA

SCGA bases its strategies on a revised version of standard GA operators (Mitchell 1998). These, established in purely continuous as well as in mixed-discrete optimisation, have been modified to handle chromosomes presenting diverse length and structure. The proposed SCGA, to make the search for the optimum more reliable, also uses other strategies, such as the population restart mechanism and the *partial* local optimisation. The main steps of the algorithm are shown in Algorithm 1, and a throughout explanation of the most relevant steps and operators are given in what follows (this implementation of SCGA is available as open-source SCGA R-package at Gentile 2020).

Algorithm 1 SCGA

```

1: Initialise the first population  $P_t$  with  $t = 0$ .
2: while not Termination Criterion do
3:   Evaluate  $P_t$  on  $f$  to get observations  $y_t$ .
4:   Assign fitness given  $y_t$ .
5:   if Population restart Criterion then
6:     Reinitialise the population  $P_{t+1}$ .
7:   else if Local optimisation Criterion then
8:     Run partial local optimisation starting from current best solution.
9:     Reinitialise the population  $P_{t+1}$ .
10:  else
11:    Apply operators (elitism, crossover, mutation) to create the new population  $P_{t+1}$ .
12:    Evaluate criteria
13:     $t = t + 1$ .
14:  end if
15: end while

```

2.1 Initial population

The first step in Algorithm 1 (Line 1) consists of generating the first set of candidates (population). Usually, for fixed-size optimisation, the initialisation of the population relies on techniques that aim to maximise the information-gain distributing samples in the search space following some design of experiment. Nevertheless, none can be blindly used in optimisation problems of this kind. (Pelamatti et al. 2017).

In the proposed SCGA, an iterative algorithm that stems from the strategy has been adopted. The approach, detailed in Algorithm 2, aims at maintaining the sparsity deriving from the stratified sampling techniques in the generation of a variable-size population. It starts creating an initial design with Latin Hypercube Sampling (LHS) considering only the genes in the highest level of hierarchy (Line 1). Then, all the remaining substructures are created using an iterative procedure that recursively defines the value of the genes and the ones of the directly dependent genes (Line 3). In this case, the values are assigned using a random uniform sampling. Then, the number of the different genes belonging to each gene class is counted (Line 7). Next, the gene classes are clustered in regard to the presence of their genes in the population (Line 8). In this way, a series of Design of Experiment (DOE) are created considering only the gene classes having the same number of genes in the population (Line 10). Finally, the so generated values are randomly reassigned to the genes created as initial design. If the chromosomes created are unfeasible, the dependent genes are reinitialised by random uniform sampling (the same as in Line 3).

Algorithm 2 Initial design

```

1: Create an initial design of all the independent genes,  $\mathcal{I}$ , using a LHS approach.
2: for  $p \leq$  population size do
3:   for  $i \in \mathcal{I}$  do
4:     Starting from the independent gene  $i$ , create all the dependent genes structure
       assigning random values within their bounds.
5:   end for
6: end for
7: Count the number of genes of each gene classes is present in the whole population
   CountGenes
8: create a sets of genes classes  $\mathcal{C}_{CountGenes}$  with common presence in the population.
9: for  $GC \in \text{dim}$  do
10:   Create an initial design of dimension coherent with CountGenes of all the genes
       classes in  $\mathcal{C}_{CountGenes}^{GC}$  using a LHS approach.
11: end for
12: for  $p \leq$  population size do
13:   for  $i \in \mathcal{I}$  do
14:     Starting from the independent gene  $i$ , recreate all the dependent genes structure,
       if feasible, assigning the values stored in the designs just created. Else, random
       values are assigned.
15:   end for
16: end for
   return design of experiment

```

2.2 Encoding

The problem formulation can integrate problem-specific information encoding them by means of hierarchical dependencies to enhance the effectiveness of the genetic operators (Greco et al. 2019).

Ceiling the number of genes is not needed when employing SCGA. So, the number of the employed ground stations and the number of passes in which these are activated is free to vary. The search space has been formulated hierarchically, imposing dependencies between genes. Consequently, the target of the operators are not single genes but rather gene structures. This structure is actively helping the operators to perform more effective modifications over the candidates. In fact, the aim is to preserve the overall information limiting the extrapolation of one gene from its context. Due to the elevated interactions, blindly exchanging genes might lead to a high information destruction rate.

Differently from classical genetic algorithms, in SCGA a chromosome (solution) is represented by a matrix either indicating the value of the gene or its position in the hierarchy of the chromosomes. The genes are grouped into *gene classes*. Each gene class is defined by data type, dependent genes, and bounds (expressed as functions).

2.3 Crossover

The crossover operator exchanges genes between two different chromosomes (*parents*) to produce two new candidates (*children*), to exploit the information contained in the encoding strings of the parents and transfer it to the children. This is one of

the operators mentioned in Line 11 in Algorithm 1 employed to evolve the population of chromosomes from one generation to another.

In classical fixed-size algorithms, all the genes lie on the same level and have a well-defined position and meaning. Genes in the same positions in the strings of two different chromosomes represent the same variable. This is not the case for structured chromosome (Greco et al. 2019), and swapping genes among parent chromosomes based on their position may result in selecting genes representing different variables and creating unfeasible and meaningless solutions. For this reason, a different strategy for selecting genes to swap based on their class has been proposed by Nyew et al. (2012). In this case, the crossover operation is permitted only on genes belonging to the same class. This approach guarantees a semantically correct crossover where the picked gene, as well as the dependent genes, are swapped. However, in the preliminary stage of this research, the strategy proposed in Nyew et al. (2012) results too destructive, making the information contained in the parents disappear over the generations. For this reason, the developed operators inherit the functional principles from Nyew et al. (2012), but the implementation and behaviour differ significantly. The number of exchanging genes belonging to each class is computed in regards to the structure of the two parents chromosomes (Lines 2 and 3 in Algorithm 3). This helps to homogenise the crossover operation all over the hierarchy of the chromosome. Moreover, the already swapped genes are removed from the list of eligible genes for crossover to maximise the information exchange per crossover operation (Line 9). This helps prevent the repetition of the crossover operation on the same genes that might be penalised the effectiveness of the information exchange.

It is worth mentioning that the feasibility of the resulting chromosome is not guaranteed. For this reason after each transformation the candidates are checked (Line 10) and, if needed, repaired (Line 12). The procedure adopted is then able to create meaningful candidates that respect the hierarchical structure of the parents.

Algorithm 3 Crossover operator

```

1: Initialise the children as the copy of the parents  $C = P$ 
2: Compute the number of genes to swap,  $n_{target}$ , in light of number of the genes belonging
   to gene classes present in both the parents
3: Compute the probability of selecting the genes class considering their presence in the
   parents
4: Initialise the set of swapped genes as an empty set  $\mathcal{G}_s$ 
5: while the number of swapped genes  $n_s \leq n_{target}$  do
6:   Randomly select the gene class of the gene to swap
7:   Randomly select the genes to swap avoiding the genes  $\in \mathcal{G}_s$ 
8:   Update  $C$  swapping the selected genes and their hierarchical dependent genes
9:   Update  $n_s$ 
10:  Check children feasibility
11:  if Any child is unfeasible then
12:    Repair unfeasible child
13:  end if
14: end while
  
```

2.4 Failures handling

In many real-world applications, the computation of the objective function (in Algorithm 1 Line 3) is itself an arduous and complex task that requires the execution of numerous routines and the data transfer between different environments. An error encountered during these computations, e.g., convergence failures, connection issues, file creation or reading failures, might crash the whole optimisation chain. For this reason, SCGA integrates an error-handling routine that saves the integrity of the optimisation chain and handles the solutions for which it was not possible to evaluate the objective function. An artificial value proportional to the worst solution found in the current generation is assigned in these cases.

2.5 Elitism

The elitism operator guarantees that the fittest handful of individuals are being brought forward in the next generation without mutation. This ensures a reasonably good chromosomes in the next population (Line 11 in Algorithm 1). In fact, the changes realised by the mutation and crossover do not always have a positive impact.

2.6 Mutation

The mutation operator is characteristic of the most population based optimiser, also SCGA (Line 11 in Algorithm 1). Many different variants of this operator can be found in the literature for standard fixed dimension optimisation (Falco et al. 2002; Jong 1975). As in the crossover case, a unique mutation operator has been used to deal with structured chromosomes. Both the mutation and the crossover have been already presented in previous works, in Greco et al. (2019) and Gentile et al. (2019b) respectively. These have not been changed in this work. However, in this paper, they have been described in further details. In this case, two

main steps can be recognised in the mutation operation (Algorithm 4). Firstly, the mutation of the gene chosen to be modified and, in particular, its value. This task finds many similarity points with the standard mutation operators. In fact, its aim is essentially to perturb the current value of the gene to introduce new information in the chromosome. Since the algorithm manages variables of different types, the basics of the mutation operations described in Li et al. (2013), Schwefel (1993), Rudolph (1996), Schütz and Sprave (1996) have been used as gene value perturbation operators Greco et al. (2019). The mutation of the offspring relies on operations acting differently on real, integer, and discrete variables, all respecting the requirements for a mutation strategy in the search spaces: accessibility, feasibility, symmetry, similarity, scalability, and maximal entropy (Li et al. 2013). The operator achieves this by adding normally distributed noise to real-valued variables. For integer variables, the distribution is based on the difference between two geometrical distributions. Categorical variables are re-sampled (uniform randomly) with some probability. The probability and the magnitude of mutation are controlled by a set of hyperparameters, the *strategy parameters*, which evolve during the optimisation for self-adaptive parameter control. To each candidate, a set of hyperparameters is associated. This is initialised as described in Algorithm 5 and evolves undergoing the same operators of the associated structured chromosome. However, the crossover operation for the mutation hyperparameter consists of simply averaging the hyperparameter sets associated with the chromosome selected for the crossover operations. In the case of mutation, the mutation strength itself evolves accordingly as described in Algorithm 6. The philosophy behind self-adaptation is that the evolutionary process can solve two problems simultaneously: the determination of the best strategy variables, and the determination of the best design variables.

The second step is treating the substructure of the mutated gene (Line 23). This is a unique feature of SCGA. In particular, also the dependent genes have to be mutated, and their number has to be consistent with the new value assumed by the mutated gene.

Algorithm 4 Mutation operator

```

1: Compute the number of genes to mutate,  $n_{target}$ , in light of the number of the genes
   belonging to gene classes present in the original chromosome
2: Compute the probability of selecting the genes class considering their presence in the
   original chromosome
3: Initialise the set of mutated genes as an empty set  $\mathcal{G}_m$ 
4: while the number of swapped genes  $n_s \leq n_{target}$  do
5:   Randomly select the gene class,  $i$ , of the gene to mutate
6:   Randomly select the gene to swap avoiding the genes  $\in \mathcal{G}_m$ 
7:    $value$  is the value of the selected gene
8:   if Gene type of  $i$  is real-valued then
9:      $value = value + \sigma_i \cdot N(0, 1) \cdot (UB - LB)$ 
10:     $value = \max(\min(value, UB_i), LB_i)$ 
11:   else if Gene type of  $i$  is integer then
12:     Compute the number of genes of class  $i$  as  $NGene$ 
13:      $p = \frac{\sigma_i}{NGene}$ 
14:      $p = \max(\frac{1-p}{1+\sqrt{1+p^2}}, 1e-3)$ 
15:      $G1 = rgeom(NGene, prob = p)$ 
16:      $G2 = rgeom(NGene, prob = p)$ 
17:      $value = value + G1 - G2 \cdot (UB_i - LB_i)$ 
18:      $value = \max(\min(value, UB_i), LB_i)$ 
19:   else if gene type of  $i$  is categorical then
20:     if  $N(0, 1) > \sigma_i$  then
21:       mutate  $value$  picking randomly a feasible value
22:     end if
23:     if  $i$  has dependent genes then
24:       if number of dependent genes has to be changed then
25:         if genes have to be added then
26:           initialise new dependent genes
27:         else if genes have to be removed then
28:           randomly pick dependent genes to be removed
29:         end if
30:       end if
31:     end if
32:   end if
33:   Mutate all the dependent genes
34: end while

```

Algorithm 5 Mutation's hyperparameters initialisation

-
- 1: Compute the average number of genes of the different type
(continuous, integer, categorical) $NGenesPerType$
 - 2: global learning rates $\tau = \frac{-2}{\sqrt{2 \cdot NGenesPerType}}$
 - 3: local learning rate $\rho = \frac{-2}{\sqrt{2 \cdot \sqrt{NGenesPerType}}}$
 - 4: step size for all the continuous variables $\sigma_{continuous} = (UB - LB) \cdot 0.1$
 - 5: step size for all the integer variables $\sigma_{integer} = (UB - LB) \cdot 0.33$
 - 6: step size for all the categorical variables $\sigma_{categorical} = 0.1$
-

2.7 Chromosomes repairing

For the correct execution of the state estimation routine, the candidates have to be semantically correct. To ensure the chromosomes fulfil all the requirements, a repairing routine, shown in (Algorithm 7), is applied after the population initialisation and after each modification made by the operators. First, it is not allowed to have multiple gene structures pointing at the same information. Then, under the same *GS* all the genes of the *Orbit Index (OI)* gene class have to assume a different value (Line 2). Moreover, these values have to be compatible with the *GS* from which they are dependent.

Second, the number of genes of the gene class *OI* have to correspond to the one indicated by the *GS* gene value associated (Line 10). Finally, to simulate a realistic scenario, the maximum budget available for the tracking observation campaign, $Cost^*$, is imposed. Instead of imposing it as an external constraint, the repair routine checks whether the budget limitation is exceeded (Line 15). If this happens, all the *BfM* are scaled to respect the limitation (Line 17).

Algorithm 6 Step size parameter σ mutation

-
- 1: **if** gene type of i is real or integer **then**
 - 2: $\sigma_i = \max(1, \sigma_i \cdot e^{\tau_i * N(0,1) + \rho_i * N(0,1)})$
 - 3: **else if** gene type of i is categorical **then**
 - 4: $\sigma_i = (1 + (\frac{1 - \sigma_i}{\sigma_i}) \cdot e^{\tau_i * N(0,1) + \rho_i * N(0,1)})^{-1}$
 - 5: **end if**
-

2.8 Population restart

Because of the intrinsic heuristic nature of GA, it is not possible to guarantee the convergence to the global optimal solution. The search might be compromised by sub-optimal basins of attraction from which escaping results unlikely if not impossible. To make the algorithm more reliable and less sensitive to the initialisation, SCGA uses a population restarting mechanism, which is when the relative change in the best solution found over the last generations is less than a specified tolerance. If the criterion is met, the whole population and the correlated mutation hyperparameters are reinitialised with a global restart.

2.9 Local optimisation

When the population is considered *collapsed* in a basin of attraction from which it is unlikely to escape, just before restarting the population, it might be beneficial to start a local search that more efficiently converges to the local optimum. However, the hierarchy in the problem formulation and the presence of integer and categorical genes makes the employment of such a strategy not straightforward. Therefore, the local search implemented in SCGA starts from the current best solution keeping untouched the chromosome structure and varying only the genes which are defined in the continuous space.

3 Tracking model

This section shows the problem formulation and fitness metric for the scheduling of tracking campaigns from heterogeneous sensors. The same model has been used in Greco et al. (2019), Gentile et al. (2019b), and it is reported here, although more compactly, to improve the paper's readability.

3.1 Optimal tracking

Uncertainty over the initial conditions, dynamical parameters, and observations values is accounted for in the scheduling model. Thus, the tracking model is framed in a filtering probabilistic fashion (Sarkka 2013)

$$p(\mathbf{x}_0) \quad (3.1a)$$

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}) \quad (3.1b)$$

$$p(\mathbf{y}_k | \mathbf{x}_k). \quad (3.1c)$$

The first probability density function (pdf) (3.1a) describes uncertainty on the initial state. The second one (3.1b) is the transition probability modelling uncertainty on the dynamical evolution. This transition probability derives from uncertainty on the nonlinear differential equation governing the spacecraft state evolution

$$\dot{\mathbf{x}} = f(t, \mathbf{x}) + \boldsymbol{\omega}, \quad (3.2)$$

where f models the deterministic components and $\boldsymbol{\omega}$ is the process noise included to account for dynamical approximations and inaccuracies (Schutz et al. 2004). Thus, the transition probability describes how the state evolves probabilistically according to the differential Eq. (3.2). The third pdf (3.1c) is the observation likelihood of each of the measurements $\mathbf{y}_{1:M} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M]$ acquired at discrete instances of time $t_1 < t_2 < \dots < t_M$. Also in this case, this likelihood stems from uncertainty on the measurement model

$$\mathbf{y} = h(t, \mathbf{x}, \mathbf{u}, \epsilon), \quad (3.3)$$

where h is a nonlinear function of the state, sensor action, and sensor noise ϵ . The observation likelihood specifies how likely an observation is for a given state according to the model (3.3). Hence, the goal of orbit determination is to compute the posterior distribution $p(\mathbf{x}_f | \mathbf{y}_{1:M})$, or its relevant statistics, of the final state $\mathbf{x}_f$, with $t_f > t_M$, given the information encoded in the previously received measurements.

Given this probabilistic model specification, the optimal scheduling is formulated as a sensor control problem (Hero et al. 2007), that is, a decision-making process to find the best sensor actions $\mathbf{u}_k$, associated to the measurement $\mathbf{y}_k(\mathbf{u}_k)$, that optimise a fitness metric $\mathcal{J}$. Thus, the sensor action influences the observation directly and, as a consequence, the state knowledge. In this paper, offline scheduling is of interest, that is, the whole tracking schedule is optimised before the first satellite passage. Thus, the scheduling problem is formulated as

$$\begin{aligned} \mathbf{u}_{1:M}^* = \underset{\mathbf{u}_{1:M} \in \mathcal{U}_{1:M}}{\operatorname{argmin}} \quad & \mathcal{J}(\mathbf{u}_{1:M}, p(\mathbf{x}_f | \mathbf{y}_{1:M})) \\ \text{s.t.} \quad & \mathcal{G}_k(\mathbf{u}_k, p(\mathbf{x}_k | \mathbf{y}_{1:k})) \in \Phi_{\mathcal{G}_k} \text{ for } k = 1, \dots, M, \end{aligned} \quad (3.4)$$

where $\mathcal{U}_{1:M}$ is the set of admissible sensor controls, $\mathcal{J}$ and $\mathcal{G}_k$ are functionals that return deterministic statistics from the posterior density and the actions, respectively, for the objective and constraints, and $\Phi_{\mathcal{G}_k}$ is the set of admissible constraints.

Given the sensor control formulation, the specific problem is scheduling optimal tracking campaigns to reduce the satellite knowledge uncertainty using observations from a heterogeneous ground network.

3.2 Problem scenario

The specific observation model depends on the nature of the tracking station employed, its geographical location, and the sensors available. Thus, in this paper, each GS- j is specified by the following characteristics:

- geodetic coordinates $(lat, lon, alt)_j$, that is, latitude, longitude and altitude over the Earth reference ellipsoid;
- specific observation model h_j depending on the physical hardware available, e.g. range, range-rate, azimuth and elevation measurements;
- reference sensor accuracy at 90 deg elevation over the station's local horizon; at different elevation angles the covariance elements worsen as $\sim 1/\sin(El(\mathbf{x}))$ (Schutz et al. 2004);
- the operational cost of the ground station $Cost_j$.

The operational cost is used to impose budget constraints on the optimal schedules to avoid the trivial solution of using as many accurate measurements as possible.

Given the network specification, the satellite could fall within the station field-of-view (FOV) never, once, or multiple times, depending on the specific orbit and

absolute time interval considered. Hence, a first deterministic visibility analysis is performed by numerically propagating the satellite reference trajectory and checking the condition $El(\mathbf{x}) > 0$ deg. From this, the total number of passages $\bar{p}_j$ in the GS-j FOV, and the contact window times $[T_{in}, T_{out}]_{p_j}$ for each passage $p_j = 0, \dots, \bar{p}_j$ when the object is visible, are computed.

The actual measurement and its estimated accuracy depend on the time at which it is taken and the station-satellite relative geometry. The sensor actions can influence these factors and, therefore, could help to acquire measurements with a greater information content. In this research, the action should first decide whether a measurement is taken from GS-j or not, then should set the number of tracking arcs, the orbital passages, and the times within each passage. The control variables encoded in the sensor action are reported in Table 1 for the general scheduling problem.

A boolean variable sets if the ground station GS-j is used. If it is employed, an integer variable describes in how many orbital passages measurements are acquired (up to $\bar{p}_j$), and a corresponding number of discrete variables is used to pinpoint the specific passage. Each orbital passage can be used only once. Thus, each of these discrete variables should have a unique value to construct a feasible schedule. Finally, for each orbit passage, an integer variable describes the Number of Measurements (NoM_j) to acquire, and continuous variables set the specific measurement times. Then, the specialised formulation for the specific test case analysed in this paper will be reported in Sect. 4.

An optimal combination of ground stations, observation types, high-elevation passages, and close measurements helps to improve the estimation accuracy. A 2D simplified depiction of a possible solution candidate is reported in Fig. 1, where only three ground stations are employed using a different number of observations.

3.3 Uncertainty measure

The goal of tracking is to improve the state knowledge as much as possible by using measurements. Different metrics were examined to quantify the estimation accuracy given a candidate tracking campaign (Schutz et al. 2004). Among others, a common metric is the RMSE of the measurement residuals. However, small residuals do not necessarily imply an accurate state estimate as there may be unobservable state components. Another approach to quantify the estimation accuracy requires comparing estimates coming from different filtering schemes. However, this metric would

Table 1 Free variables and types encoded within sensor action $\mathbf{u}_k$

Description	Variable type	Values
Use ground station j	Discrete	ON/OFF
Number of passages to use	Integer	$[0, \bar{p}_j] \in \mathbb{N}$
Indexes of passages to use	Discrete	$[1, \bar{p}_j] \in \mathbb{N}$
Number of observations per passage	Integer	$[0, NoM_{max}] \in \mathbb{N}$
Times of observations per passage	Continuous	$t \in [T_{in}, T_{out}]_{p_j}^p$

Fig. 1 Sketch of space object passage over ground station network

be rather expensive to call numerous times within the scheduling search. Finally, the estimation covariance is a metric that describes how spread the posterior pdf is around the optimal estimate. The covariance is directly computed in Kalman-based filters or easily retrieved from samples in particle filters. As a downside, the covariance results typically optimistic and highly dependent on the selected (if any) process noise (Schutz et al. 2004). However, only a relative measure of the estimation accuracy between different candidate schedules is of interest in the scheduling optimisation, and the process noise can be kept constant in each fitness' evaluation. Therefore, the covariance was chosen as the accuracy metric of the estimation process for the scope of this analysis. As the optimisation works on a scalar quantity, the trace of the covariance is employed as metric to quantify the confidence in the state elements

$$\mathcal{J} = \text{Tr}\{\Sigma_{x_f|y_{1:M}}\}. \quad (3.5)$$

Since the actual observations are not available yet at the time of scheduling, a covariance analysis (Gelb 1974; Geller 2006; Stastny and Geller 2008) is employed to compute the final covariance using the expected measurements' accuracy. Covariance analysis is mostly used for the navigation analysis of space missions to design suitable tracking strategies (Stastny and Geller 2008; Ionasescu 2010; Ionasescu et al. 2014). In this approach, orbit determination is performed around the reference trajectory, and the measurements are employed to model the reduction of the second moment of the state distribution. Practically, this is achieved by simulating observations along the reference trajectory with zero noise.

A square root Unscented Kalman Filter (SR-UKF) (Merwe and Wan 2001) is employed as a filtering algorithm suitable for normal distributions and non-linear dynamical and observational models. This nonlinear filter employs dynamical and observation model's evaluations at collocation points to compute the mean and covariance prediction and update. The square root of the covariance matrix is employed in place of the covariance itself to improve the numerical stability of the method (Schutz et al. 2004). The covariance needed for Eq.(3.5) is then computed by multiplying the covariance square root by its transpose.

3.4 Budget constraint

In general, the more accurate measurements employed, the better the estimation accuracy would be. However, operating a tracking station has an associated monetary and human cost. Thus, each observation has an associated cost to quantify the budget corresponding to a specific schedule. In this model, the cost $Cost_j$ will depend on the sensors' accuracy, station architecture, ground operation costs, number of measurements, etc. The total cost $Cost$ is the sum of the individual costs, and the budget constraint is formulated as

$$Cost(\mathbf{u}_{1:M}) \leq \overline{Cost}, \quad (3.6)$$

where $\overline{Cost}$ is the maximum budget available.

3.5 Tracking encoding

Once the optimiser has set a collection of sensor controls, a hierarchical structure with three levels has been employed for the chromosome encoding, as depicted in Fig. 2.

The top class *Ground Station* indicates the number of different satellite passes in which a specific ground station will be employed, and therefore N_{GS} genes of this class will be present in all the solutions. However, because different GSs may have a different number of orbital passages, the upper bound $\bar{p}_j$ depends on the specific GS. The intermediate class *Orbit index (OI)* represents specifies the indexes of the orbital passages in which observations are acquired. The bottom gene class *Budget for Measurements (BfM)* indicates the percentage of the total available budget for each specific orbital passage of a given GS. The actual *Number of Measurements (NoM_j)* to be acquired is expressed by the equation

$$NoM_j = \left\lfloor \frac{BfM_j}{\overline{Cost_j}} \right\rfloor \quad (3.7)$$

where BfM_j is the allocated *BfM* to GS- j .

Fig. 2 Chromosomes-hierarchy in the SCGA's formulation

Fig. 3 Example of two candidate observation schedules

As an example, two candidate schedules are shown in Fig. 3 by depicting the multi-level structure for GS-1, GS-2 and GS-9 only. The solution in Fig. 3a allocates observations as: GS-1 on 2 different orbital passages $OI-1$ and $OI-2$ with BfM of 0.12 and 0.16 respectively; GS-2 is not employed; GS-9 on 3 different orbital passages $OI-1$, $OI-4$, and $OI-5$ with the reported BfM . The schedule in Fig. 3b has observations for: GS-1 on 1 orbital passage, $OI-1$; GS-2 on 1 orbital passage, $OI-1$; GS-9 on 3 orbital passage, $OI-1$, $OI-3$, and $OI-5$. The two candidates have differences on each structure level. Indeed, the second candidate uses GS-2 while the first does not, the second schedule requires observations on the $OI-3$ of GS-9 whereas the first employs the orbital passage $OI-4$, and the BfM are all different.

Furthermore, the accuracy of the estimation process strongly depends on the exact measurement times. The choice of measurement times within an orbital passage in the station FOV comes from the compromise of two conflicting factors: acquiring the most accurate measures by concentrating them around the maximum elevation point; acquiring the most independent observations possible spreading them in the whole observation window.

Hence, in this work, given the NoM_j , the measurement times are computed as the values of the normal cumulative distribution with mean = 0.5 and standard deviation = 0.16 at the points that evenly divide the interval $[0,1]$ NoM_j times. The times are then linearly scaled within the corresponding time interval $[T_{in}, T_{out}]_j^p$ (see Table 1). A graphical representation is given in Fig. 4. This method concentrates the measurements around the maximum elevation point (scaled time 0.5) while adding observations at the interval extrema as NoM increases.

4 Experiments

In this section, the details of the setup used in the experiments are presented. Specifically, it describes the dynamical model, the GS network, the algorithms used for comparison, and SCGA's settings.

Fig. 4 Cumulative distribution function employed to compute measurement times within the FoV

4.1 Test case

The dynamical system is parameterised in Cartesian coordinates, and several forces are included to model the motion in LEO faithfully (Montenbruck and Gill 2012): Earth's gravitational force using the EGM96 geopotential model up to degree and order 10; atmospheric drag with Jacchia-Gill model; third-body disturbances due to the Moon and Sun gravitational pull; Solar Radiation Pressure (SRP) with a conical shadow model. The satellite epoch date is 2018 October 29 12:00 UTC, with reference Keplerian elements

$$\begin{aligned}\boldsymbol{\mu}_{kep_0} &= (a[\text{km}], e[-], i[\text{rad}], \Omega[\text{rad}], \omega[\text{rad}], \theta[\text{rad}]) \\ &= (6608.17, 1.61e-3, 1.685, 5.662, 1.199, 1.589) .\end{aligned}$$

After conversion to Cartesian coordinates $\boldsymbol{\mu}_0$, this estimate is the mean of the Gaussian initial state distribution $p(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ with covariance set as

$$\begin{aligned}\boldsymbol{\Sigma}_0 &= \text{diag}(1.0e^{-2} \text{ km}, 1.0e^{-2} \text{ km}, 1.0e^{-2} \text{ km}, \\ &\quad 1.0e^{-4} \text{ km/s}, 1.0e^{-4} \text{ km/s}, 1.0e^{-4} \text{ km/s}) .\end{aligned}$$

The drag cross-section is set to 15.0 m^2 . The mean solar flux considered is 106.4 in solar flux units, with a mean SRP cross-section of 1.625 m^2 and a SRP coefficient of 1.3 . These system and orbital parameters are set to resemble a GOCE-like satellite. The tracking window lasts 8 hours, and it ends 2018 October 29 20:00 UTC.

4.1.1 Ground station network

The simulated and hypothetical GSs network considered for this work is composed by a set of $N_{GS} = 9$ modelled GSs. To investigate the effect of the GSs

Table 2 Ground station network configurations (Conf-1, Conf-2 and Conf-3) employed for the test case describing stations geodetic latitude [deg], longitude [deg], 1σ accuracy in measurements for Range (R) [km], Range-Rate (RR) [km/s] and Azimuth and Elevation (AzEl) [rad] and cost given by Equation (4.2) per measurement

GS	Latitude	Longitude	Feature	Conf-1	Conf-2	Conf-3
GS_1	35.43	243.11	R	NO	NO	5.00E-01
			RR	NO	7.76E-05	NO
			AzEl	3.64E-03	6.52E-03	NO
			Cost	1.04E-01	1.57E-01	1.25E-01
GS_2	− 35.4	148.98	R	NO	1.36E-01	5.00E-01
			RR	NO	NO	NO
			AzEl	2.56E-03	NO	NO
			Cost	4.36E-02	7.04E-02	1.25E-01
GS_3	40.43	355.75	R	NO	2.24E-01	5.00E-01
			RR	4.39E-05	NO	NO
			AzEl	2.90E-03	NO	NO
			Cost	1.04E-01	8.36E-02	1.25E-01
GS_4	40.53	17.43	R	NO	7.14E-01	5.00E-01
			RR	1.13E-05	1.00E-04	NO
			AzEl	NO	2.44E-03	NO
			Cost	6.56E-02	3.00E-01	1.25E-01
GS_5	41.89	12.48	R	NO	NO	5.00E-01
			RR	2.66E-05	8.14E-05	NO
			AzEl	9.04E-03	NO	NO
			Cost	1.37E-01	8.59E-02	1.25E-01
GS_6	40.65	16.7	R	NO	9.74E-01	5.00E-01
			RR	NO	6.97E-05	NO
			AzEl	2.00E-03	7.75E-03	NO
			Cost	3.94E-02	3.56E-01	1.25E-01
GS_7	67.86	20.96	R	3.72E-01	9.10E-01	5.00E-01
			RR	NO	1.92E-05	NO
			AzEl	3.56E-03	NO	NO
			Cost	2.08E-01	2.25E-01	1.25E-01
GS_8	− 72.01	2.53	R	4.78E-01	7.53E-01	5.00E-01
			RR	NO	2.53E-05	NO
			AzEl	2.94E-04	NO	NO
			Cost	8.73E-02	2.06E-01	1.25E-01
GS_9	78.23	15.4	R	1.09E-01	5.18E-01	5.00E-01
			RR	6.28E-05	1.26E-05	NO
			AzEl	8.47E-03	NO	NO
			Cost	1.93E-01	1.62E-01	1.25E-01

characteristics on the solutions found and the behaviour of the search strategies, three different scenarios have been considered and implemented, as reported in Table 2. These scenarios differ for the measurement types acquired by each GS, their accuracy and, consequently, their cost of utilisation. The station coordinates have been saved in data kernels for NASA's SPICE toolkit (Acton et al. 2018) to retrieve their inertial state at different epochs while accounting for Earth's rotation. As it can be seen, each GS can observe one or more of the following quantities: the station-satellite Range (R), Range-Rate (RR) and Azimuth and Elevation (AzEl). The first two configurations (Conf-1 and Conf-2) have been randomly generated to reflect two realistic scenarios characterised by a network of variegated GSs. The third (Conf-3) is a simplified scenario in which all the GSs share the same characteristics. The simplification of the scenario analysed should facilitate the interpretation of the results obtained.

As a result of the network-satellite geometry, the object falls in the GS' FOVs multiple times. The satellite passes over the GSs are sketched in Fig. 5, which displays the azimuth and elevation angles over the GS' local horizon. The total number of potential tracking passages over all the stations is equal to 21.

The maximum elevation angle ≈ 71 deg is achieved over the tracking station GS-9, which, in addition, sees the satellite in multiple passes. Good elevation angles ≥ 50 deg

Fig. 5 Sky plots of satellite passes over ground stations, with concentric circles indicating different elevation levels, while the angular quantity represents the azimuth measured eastwards from the local north. Different colours indicate different satellite passes over the same station

are also realised above stations GS-3, GS-4, GS-8, whereas the worst ones correspond to station GS-2 and GS-5.

It is expected that the optimal tracking schedule would be characterised by accurate measurements at high elevation angles to improve the satellite visibility. Besides, the final accuracy depends also greatly on the observation timing, that is, when the observations are acquired along the trajectory. All these features make the problem highly multi-factorial and complex to handle.

4.1.2 GS cost model

In this test case, the cost of operating each ground station has been assumed to change linearly with the measurement accuracy. For each measurement type, the minimum and maximum accuracy and associated cost are defined in Table 3 equivalently for each GS.

Hence, if the GS_j can take measurement $M = \{R, RR, AzEl\}$ with accuracy Acc_j^M , the cost of a single measurement of type M from that ground station is computed as

$$Cost_j^M = Cost_{min}^M + (Acc_j^M - Acc_{min}^M) \cdot \frac{(Cost_{max}^M - Cost_{min}^M)}{(Acc_{max}^M - Acc_{min}^M)}. \quad (4.1)$$

This unit cost is summed over the types of measurements available at that station and multiplied by the number of measurements NoM_j to acquire by station GS_j

$$Cost_j = NoM_j \sum_M Cost_j^M \quad (4.2)$$

to compute the cost of operating GS_j . The total cost of performing the tracking campaign dictated by the control sequence $\mathbf{u}_{1:M}$ is given by the sum of the costs of the individual stations as

$$Cost(\mathbf{u}_{1:M}) = \sum_j Cost_j. \quad (4.3)$$

Consequently, a maximum budget $Cost^*$ constraining the number of observations is imposed in the optimal scheduling. The threshold value $\overline{Cost}$ is varied for different runs of the optimisation to analyse the impact of limited versus large budgets in the estimation accuracy. The maximum budget is varied in the range from 1.5 to 9 with a step of 1.5.

Table 3 Minimum and maximum 1σ accuracy and cost for each measurement type $M = \{R, RR, AzEl\}$

Parameter			Symbol	R [km]	RR [km/s]	AzEl [rad]
Accuracy	Min		Acc_{min}^M	1e-04	1e-06	1e-04
	Max		Acc_{max}^M	1e-00	1e-04	1e-02
Cost	Min		$Cost_{min}^M$	5e-02	25e-02	25e-02
	Max		$Cost_{max}^M$	2e-01	1e-01	1e-01

4.2 Comparison algorithms

The performances of the proposed optimiser have been compared to the ones of a standard GA (Scrucca 2013) and the hidden-genes variant proposed in Darani and Abdelkhalik (2017).

As well known, the standard GA is not intended to handle variable-sized optimisation problems. Therefore, applying this strategy for facing the presented problem requires assuming that all the GSs are always used. With this assumption, the only thing the optimiser manages is the budget for measurements at each satellite pass over the GSs. On the one hand, the usage of a GS for a given passage is fully determined by one variable, on the other hand, the problem definition is straightforward. The decision variables are 21 real-valued variables corresponding to all the 21 observations windows as shown in Table 4.

The hidden-genes variant makes use of activation genes for activating or deactivating genes. As in the previous case, it is assumed that the GSs are used at every pass. Then, at every gene an *activation* one is associated. Its value indicates whether to consider or not the associated gene in the decision space. Utilizing this trick, it is possible to encode and manage solutions with different architectures using standard genetic operators.

In the considered mission, the maximum number of decision variables is 21, one for each time the satellites falls in the FOV of one of the GSs of the tracking network. In the problem formulation for the standard GA, these constitute the complete decision space. In the hidden-genes problem formulation, the 21 additional activation genes are introduced, for a total of 42 genes. The decision variables are 21 real-valued variables and 21 categorical variables as shown in Table 4. A particular feature of the implemented hidden-genes GA is the crossover operator. As proposed in Darani and Abdelkhalik (2017), this operation is conducted separately on the decision and activation genes.

Both standard GA and hidden-genes variants have been implemented using the GA R-package and using the default options but for the crossover operator in the case of the hidden-genes variant.

Table 4 Decision variable of the Observation Scheduling optimisation Problem

Algorithm	Gene	Type	Lower Bound	Upper Bound
SCGA	Ground Station	Integer	$0 \forall GS_{1:9}$	[2, 1, 1, 1, 2, 1, 5, 3, 5]
	Orbit index	Discrete	$1 \forall GS_{1:9}$	[2, 1, 1, 1, 2, 1, 5, 3, 5]
	Budget	Real	$0 \forall GS_{1:9}$	$Cost^* \forall GS_{1:9}$
Hidden-genes GA	Budget	Real	$0 \forall OW_{1:21}$	$Cost^* \forall OW_{1:21}$
	Activation	Discrete	$0 \forall OW_{1:21}$	$1 \forall OW_{1:21}$
GA	Budget	Real	$0 \forall OW_{1:21}$	$Cost^* \forall OW_{1:21}$

4.3 SCGA settings

The SCGA setting is summarised in Table 4 too. In this case, the bounds vary depending on the related GS. In fact, the number of times the object falls in the FOVs of the different GSs is different (see Fig. 5).

SCGA has been tested using three different structures: SCGA with population restart mechanism, SCGA with local search and plain SCGA. Both the restart mechanism and local search strategies have been activated after 50 generations during which the improvement of the best found solution is lower than 1%. For all the structures, a population size equal to 30 has been used. All the optimisation runs have been limited to 13, 500 function evaluations (corresponding to 500 generations of the standard SCGA considering that 1% of the population is elitist). The setting for all the remaining parameters can be found at Gentile (2020).

To check the proper convergence of the algorithms, supplementary runs of the SCGA with population restart and hidden-genes GA limited to 27, 000 functions evaluations have been run for the instances with $Cost^*$ equal to 3.5 and 9 only.

All the optimisation strategies investigated, standard GA, the hidden-genes GA, plain SCGA, SCGA with population restart and SCGA with local search have been tested on the same instances of satellite tracking observation campaigns problem. For each instance, characterised by a GSs network configuration and a maximum available budget, 50 independent runs have been run to have statistical results.

The state estimation routine can return values that can differ by many order of magnitude or error flags due to model divergence. In this case, an artificial value equal to 110% of the worst solution found in the current generation is assigned.

5 Results

The results for all the problem configurations are summarised in Figs. 6 and 7, where the covariance trace of the final optimal solutions found in the different runs are depicted through box and violin plots using a linear and logarithmic y-axis, respectively. As expected, the solutions found for configurations Conf-1 and Conf-2 are far more accurate than the ones of Conf-3 for every budget allocation and algorithm. This is because Conf-3 uses only quite coarse range measurements, and therefore the state is not fully observable.

The effect of the budget is particularly evident in Fig. 7. It can be seen that all the tested algorithms can find better solutions as the budget increases, but, for smaller budgets, the results obtained by GA and GA-Hidden are several orders of magnitude worse than the ones obtained by SCGA.

Imposing severe budget constraints undermines GA and GA-Hidden's ability to find optimal solutions, whereas SCGA still manages to reduce the satellite uncertainty considerably. Increasing the available budget mitigates this effect considerably, and in the less constrained scenario with $Cost^* = 9$ and GS network Conf-3, the difference between the results found by GA-Hidden and plain SCGA are not statistically significant. An analysis about the statistical differences between the solutions

Fig. 6 Box and violin plots representing the best results found by the algorithms, in the 50 independent runs, for all the analysed problem configurations. A linear scale is used for the y-axis. The colour indicates the algorithm the plots refer to. The labels denote the instances specifying the configuration (C: Conf) and the $Cost^*$ (B: Budget)

Fig. 7 Box and violin plots representing the best results found by the algorithms, in the 50 independent runs, for all the analysed problem configurations. A fixed-logarithmic is used for the y-axis. The colour indicates the algorithm the plots refer to. The labels denote the instances specifying the configuration (C: Conf) and the $Cost^*$ (B: Budget)

Fig. 8 Results of the Wilcoxon rank sum test over the final optimal solutions. The colour indicates whether the difference in the results found by the algorithms for a determined instance is statistically significant. White spaces indicate that the algorithms with double function evaluations have not been tested for the correspondent instance

found by the different algorithms is reported in Fig. 8, where the Wilcoxon rank sum test has been applied on the solutions found with GS network Conf-3.

In all the investigated scenarios, but in a special way when the budget limitation significantly shrinks the feasible search space, SCGA demonstrates to be more reliable and better performing than GA-Hidden and standard GA. Moreover, the notable performances of SCGA are not limited in the quality of the final solutions found but also in the velocity of convergence, as it can be seen from Fig. 9. This is more emphasised in low-budget configurations like the one depicted in Fig. 9a. However, also in a less constrained instance as the one reported in Fig. 9b, the standard GA shows to be inadequate to cope with this kind of problems. This reinforces the hypothesis that traditional fixed-size optimisation strategies actually struggle to face variable-size problems even when reformulated.

The four best solutions for each algorithm in the two considered configurations and their objective value are visualised in Fig. 10, and their objective values (Obj) reported. For each ground station, there are as many rectangles as the number of passes above that stations employed, and the numbers within the rectangles indicate the corresponding *Orbit index*. If no box is reported, then that ground station is not employed at all. The rectangle colours indicate the number of measurements employed in each satellite pass: the more the colour is bright, the more

Fig. 9 Anytime performance plot of two problem instances

measurements are obtained within that passage, whereas the more it is faded, the fewer observations are acquired.

Looking at Fig. 10a, the tracking schedules found by the SCGA variants are very similar, whereas there are some important differences with the ones optimised by fixed-size GAs. In the solutions found by SCGA, one of the convenient stations for the early part of the tracking window, GS_1 , is always employed as well as the highest elevation ones GS_8 and GS_9 . GS_5 is employed in several solutions as well due to its accurate RR measurement capability, which is highly informative, at a relatively low cost (see Table 2).

Fig. 10 Visualisation of the four best solutions found by all the algorithms on two different instances

Fig. 11 Distribution of the observations over the tracking window in all the final solutions found

Similar considerations can be made by analysing the solutions shown in Fig. 10b. In this case, the difference in the solutions found by SCGA variants and GA-Hidden decreases while the solutions found by GA still significantly differ. In particular, it appears to be due to the difficulty in screening the more convenient GSs. A further motivation on the GA performance can be found looking at the budget allocation efficiency metric reported in Fig. 10. Given the definition of NoM_j in Eq. (3.7), there is a waste of resources when the budget allocated to a GS is not exactly a multiple of its cost of employment. The efficiency metric is then defined as

$$\text{Eff} = \frac{\sum_j \text{Cost}_j}{\text{Cost}^*} \cdot 100. \quad (5.1)$$

Therefore, looking at the efficiencies of the solutions found by standard GA, the reason for its poor performances could also be ascribed to its incapacity to allocate the budget efficiently despite the simple problem formulation. On the other hand, all the best solutions for Conf-3 employ GS_8 and GS_9 primarily because of their high elevation passages. Multiple passages of GS_9 are often employed to get observation during different phases of the satellite trajectory, rather than only the second passage like in Conf-2.

Interesting insights can also be obtained by analysing the observations' distribution over the time of the optimal observation campaigns. A representation of the observation times of all the 50 final solutions is visualised in Fig. 11 for every

Fig. 12 Distribution of the observations over the tracking window in the best found solutions by all the optimisers

algorithm and configuration and budgets 1.5, 7.5 and 9. The observation instances are also colour coded for the ground station used to take the measurement.

A first consideration might be related to the evolution of the observation campaigns with the growth of the maximum budget allocated. In all the solutions found by SCGA, the same qualitative behaviour can be observed. A large share of the budget is usually well spread over all the convenient GSs, even in the cases with a severe budget limitation. Relaxing this constraint results in a more generous budget allocation to one GS, which evidently is the most informative with respect to the cost of employment and offering the highest elevation angles (GS_4 in the case of Conf-1 and GS_9 in the case of Conf-2 and Conf-3). This also impacts the timing of the observations. In the first configuration, most observations are concentrated in the middle of the tracking window. Contrary, in the other configurations, the concentration is located toward the end. Referring at Table 2, it can be seen that there is a significant difference in the cost of employment of GS_4 in the different GSs configurations. Similarly, all the observation times for the other algorithms remain qualitatively similar for a fixed configuration but different budgets, differing only in the number of measurements acquired. This means that often a suitable schedule is also found at lower budgets.

Figure 12 shows the observation times of only the best-found solution by each algorithm for the same configurations and budget as before. The y-axis now reports the solution fitness value while the colours indicate the specific algorithm now.

Most of the observations are aggregated toward the end of the tracking campaign. This is because the reduced uncertainty on the state has less time to grow due to dynamical propagation. Increasing the budget leads to the rise of earlier observations as well, such that the state uncertainty is further reduced at later times.

6 Conclusions

This work introduced an optimisation-based scheduling method for generating optimal tracking campaigns of space objects under uncertainty and limited budget allocations.

All the ground pipeline consisting of stations' modelling, observation simulation, uncertainty propagation, and orbit determination has been designed. A Structured-Chromosome Genetic Algorithm has been developed to tackle the variable-size optimisation problem efficiently. The goal of the method is to minimise the uncertainty on the state knowledge of Earth-orbiting objects through efficient tracking from heterogeneous ground stations. This research aims at contributing towards enhanced methodological efficiency and autonomy in the ground segment of space missions to face their always-increasing operational demand.

The proposed scheduler has been used to identify the optimal observation campaigns for a satellite in LEO. The developed methodology has been tested on different configuration scenarios. In particular, the optimisations have been repeated using three GS network configurations and using six levels of severity for the maximum available budget limitation. The characteristics of the GS network available have been varied changing the measurement accuracy of each GS and its architecture together with its cost of utilisation. Three settings of SCGA have been implemented and tested: SCGA with population restart mechanism, SCGA with local search, and plain SCGA. For comparison, we tested on the same problem an implementation of the standard GA and the variable-size GA variant proposed in Darani and Abdelkhalik (2017) named hidden-genes GA. The results indicate that the proposed approach can enhance tracking and resources allocation, especially under severe budget limitations. Indeed, the analyses show the clear benefits granted by employing a variable-size algorithm regardless of the problem instance. On the other hand, the advantages of using SCGA rather than the hidden-genes GA tend to decrease with larger maximum budgets. Also, considering the implementation effort for a proper problem formulation required by SCGA, all the more reason the employment of this algorithm in place of the hidden-genes GA should be carefully considered in cases where the budget limitation does not play a crucial role.

Future developments could focus on the generalisation of tracking scheduling as a multi-objective problem. Indeed, an efficient implementation of multiple budget levels to generate Pareto fronts for a tracking window, comparing different levels of accuracy and allocated budget, could help the operator decide the most suitable schedule to select. Besides, a dynamic approach could be developed where the tracking optimisation decides a subset of observations ahead in time, updates the state estimate with the actual observations received, and re-optimises

future tracking schedules. This would improve the tracking optimality in online applications and partially remove the need for covariance analysis.

Acknowledgements This work was funded by the European Commission's H2020 programme, through the H2020-MSCA-ITN-2016 UTOPIAE Marie Curie Innovative Training Network, Grant agreement No.: 722734.

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/>.

References

- Acton C, Bachman N, Semenov B, Wright E (2018) A look towards the future in the handling of space science mission geometry. *Planet Space Sci* 150:9–12
- Alan de Jong K (1975) Analysis of the behavior of a class of genetic adaptive systems. Technical report
- Ali FZ, Rahim SNM, Jusoh MH (2021) Amateur satellite ground station: troubleshooting and lesson learned. *J Phys Conf Ser* 1768(1):012013
- Antoniou M, Petelin G, Papa G (2020) On formulating the ground scheduling problem as a multi-objective bilevel problem. In: international conference on bioinspired methods and their applications (BIOMA), Springer, pp 177–188
- Clement B, Johnston M (2005) The deep space network scheduling problem. In: seventeenth annual conference on innovative applications of artificial intelligence, pp 1514–1520, 01
- Damiani S, Dreihahn H, Noll J, Nizette M, Calzolari GP (2007) A planning and scheduling system to allocate esa ground station network services. In: the int'l conference on automated planning and scheduling. Citeseer
- Darani SA, Abdelkhalik O (2017) Space trajectory optimization using hidden genes genetic algorithms. *J Spacecr Rocket* 55(3):764–774
- De Falco I, Cioppa Della A, Tarantino E (2002) Mutation-based genetic algorithm: performance evaluation. *Appl Soft Comput* 1(4):285–299
- Gelb A (1974) Applied optimal estimation. MIT press Chap. 7
- Geller DK (2006) Linear covariance techniques for orbital rendezvous analysis and autonomous onboard mission planning. *J Guid Control Dyn* 29(6):1404–1414
- Gentile L (2020) LorenzoGentile/SCGA: SCGA for satellite tracking. <https://doi.org/10.5281/zenodo.3968260>
- Gentile L, Filippi G, Minisci E, Bartz-Beielstein T, Vasile M (2020) Preliminary spacecraft design by means of structured-chromosome genetic algorithms. In: 2020 IEEE congress on evolutionary computation (CEC), pp 497–505. IEEE
- 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 70th international astronomical congress
- Gentile L, Greco C, Minisci E, Bartz-Beielstein T, Vasile M (2019) Structured-chromosome GA optimisation for satellite tracking. In: proceedings of the genetic and evolutionary computation conference companion, pp. 1955–1963
- Gentile L, Morales E, Quagliarella D, Minisci E, Bartz-Beielstein T, Tognaccini R (2020) High-lift devices topology optimisation using structured-chromosome genetic algorithm. In: 2020 IEEE congress on evolutionary computation (CEC), pp 497–505, IEEE

- Greco C, Gentile L, Filippi G, Minisci E, Vasile M, Bartz-Beielstein T (2019) Autonomous generation of observation schedules for tracking satellites with structured-chromosome ga optimisation. In: 2019 IEEE congress on evolutionary computation (CEC) pp 497–505. IEEE
- Helvajian H, Janson S (2009) Small satellites: past, present, and future. Aerospace Press El Segundo, CA
- Hero AO, Castañón D, Cochran D, Kastella K (2007) Foundations and applications of sensor management. Springer Science and Business Media
- Ionasescu R (2010) Orbit determination covariance analysis for the Cassini solstice mission. In AIAA/AAS astrodynamics specialist conference
- Ionasescu R, Martin-Mury T, Valerino P, Criddle K, Buffington B, McElrath T (2014) Orbit determination covariance analysis for the Europa Clipper mission. In AIAA Space 2014
- Kessler Donald J, Johnson Nicholas L, Liou JC, Matney M (2010) The kessler syndrome: implications to future space operations. *Adv Astronaut Sci* 137(8):2010
- Kleinschrodt A, Freimann A, Christall S, Lankl M, Schilling K (2017) Advances in modulation and communication protocols for small satellite ground stations. In: proceedings of the 68th international astronautical congress, pp 2-s2
- Kleinschrodt A, Reed N, Schilling K (2016) A comparison of scheduling algorithms for low cost ground station networks. In: 67st international astronautical congress. Guadalajara, Mexico, pp 1–15
- Kopacz JR, Herschitz R, Roney J (2020) Small satellites an overview and assessment. *Acta Astronautica* 170:93–105
- Lewis H, Radtke J, Rossi A, Beck J, Oswald M, Anderson P, Bastida Virgili B, Krag H (2017) Sensitivity of the space debris environment to large constellations and small satellites. *J Br Interplanet Soc* 70(2–4):105–117
- Li R, Emmerich MTM, Eggermont J, Bäck T, Schütz M, Dijkstra J, Reiber JHC (2013) Mixed integer evolution strategies for parameter optimization. *Evolut Comput* 21(1):29–64
- Liou J-C, Johnson NL (2008) Instability of the present leo satellite populations. *Adv Space Res* 41(7):1046–1053
- McDowell JC (2020) The low earth orbit satellite population and impacts of the SpaceX starlink constellation. *Astrophys J* 892(2)
- Mehrholz D, Leushacke L, Flury W, Jehn R, Klinkrad H, Landgraf M (2002) Detecting, tracking and imaging space debris. *ESA Bull* (0376-4265), (109): 128–134
- Mitchell M (1998) An introduction to genetic algorithms. MIT press, Cambridge
- Montenbruck O, Gill E (2012) Satellite orbits: models, methods and applications. Springer Science and Business Media
- Muelhaupt TJ, Sorge ME, Morin J, Wilson RS (2019) Space traffic management in the new space era. *J Space Saf Eng* 6(2):80–87
- Nyew HM, Abdelkhalik O, Onder N (2012) Autonomous interplanetary trajectory planning using structured-chromosome evolutionary algorithms. In AIAA/AAS astrodynamics specialist conference, pp 4522
- Nyew HM, Abdelkhalik O, Onder N (2015) Structured-chromosome evolutionary algorithms for variable-size autonomous interplanetary trajectory planning optimization. *J Aerosp Inform Syst* 12(3):314–328
- Pelamatti J, Brevault L, Balesdent M, Talbi EG, Guerin Y (2017) How to deal with mixed-variable optimization problems: an overview of algorithms and formulations. In world congress of structural and multidisciplinary optimisation, Springer, pp 64–82
- Petelin G, Antoniou M, Papa G (2021) Multi-objective approaches to ground station scheduling for optimization of communication with satellites. *Optimization and Engineering* 1–38
- Riesing KM (2018) Portable optical ground stations for satellite communication. Massachusetts Institute of Technology PhD thesis
- Rossi A, Alessi EM, Valsecchi GB, Lewis H, Radtke J, Bombardelli C, Bastida Virgili B (2017) A quantitative evaluation of the environmental impact of the mega constellations. In: Flohrer T, Schmitz F (eds), Proceedings of the 7th European conference on space debris, volume 7. The European Space Agency (ESA)
- Rudolph PKC (1996) High-lift systems on commercial subsonic airliners
- Safyan M (2020) Planet's dove satellite constellation. Springer International Publishing, Cham, pp 1–17
- Sarkka S (2013) Bayesian Filtering and Smoothing, 1st edn. Cambridge University Press, New York

- Schmidt M, Rybysc M, Schilling K (2008) A scheduling system for small ground station networks. In SpaceOps 2008 conference
- Schutz B, Tapley B, Born GH (2004) Statistical orbit determination. Elsevier, Amsterdam
- Schütz M, Sprave J (1996) Application of parallel mixed-integer evolution strategies with mutation rate pooling. In: proceedings of the fifth annual conference on evolutionary programming. Citeseer
- Schwefel HPP (1993) Evolution and optimum seeking: the sixth generation. Wiley, Hoboken
- Scrucca L et al (2013) Ga: a package for genetic algorithms in r. *J Stat Softw* 53(4):1–37
- Soma P, Venkateswarlu S, Santhalakshmi S, Bagchi T, Kumar S (2004) Multi-satellite scheduling using genetic algorithms. In: Space OPS 2004 Conference, page 515
- Spangelo S, Cutler J, Gilson K, Cohn A (2015) Optimization-based scheduling for the single-satellite, multi-ground station communication problem. *Comput Operat Res* 57:1–16
- Stastny NB, Geller DK (2008) Autonomous optical navigation at jupiter: a linear covariance analysis. *J Spacecr Rocket* 45(2):290–298
- Sun J, Xhafa F (2011) A genetic algorithm for ground station scheduling. In: 2011 international conference on complex, intelligent, and software intensive systems, pp 138–145, IEEE
- Toorian A, Diaz K, Lee S (2008) The cubesat approach to space access. In: 2008 IEEE aerospace conference, pp 1–14
- Van Der Merwe R, Wan EA (2001) The square-root unscented kalman filter for state and parameter-estimation. In: acoustics, speech, and signal processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE international conference on, volume 6, pp 3461–3464. IEEE
- Xhafa F, Herrero X, Barolli A, Takizawa M (2013) A simulated annealing algorithm for ground station scheduling problem. In: 2013 16th international conference on network-based information systems, IEEE, pp 24–30
- Xi J, Xiang Y, Ersoy OK, Cong M, Wei X, Junkai G (2020) Space debris detection using feature learning of candidate regions in optical image sequences. *IEEE Access* 8:150864–150877

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