Preliminary spacecraft design by means of Structured-chromosome Genetic Algorithms

Lorenzo Gentile[1,2], Gianluca Filippi[2], Edmondo Minisci[2], Thomas Bartz-Beielstein[1], Massimiliano Vasile[2]
[1]*Institute for Data Science, Engineering, and Analytics*, *TH Köln*, Köln, Germany
{lorenzo.gentile, thomas.bartz-beielstein}@th-koeln.de
[2]*Aerospace Centre of Excellence*, *University of Strathclyde*, Glasgow, UK
{g.filippi, edmondo.minisci, massimiliano.vasile}@strath.ac.uk

***Abstract*—This paper presents a new methodology for complex system design by means of optimisation techniques. Within the Model-based Engineering approach, optimisation algorithms are used to explore optimal solutions of highly coupled and nonlinear systems. In such scenario, the optimal technology has to be identified and its settings have to be optimised. Relying on optimisation strategies for both the challenges brings to complex mixed-variable problem formulations involving continuous, integer and categorical parameters. Furthermore, part of the parameters are required only if certain technologies are adopted, bringing to variable-size formulations that standard optimisers cannot manage. Therefore, the proposed methodology relies on the use of variable-size mixed-variable global optimiser Structured-Chromosome Genetic Algorithm (SCGA). The advantages of this new method are shown by applying it for solving a space system preliminary design. In particular, two variants have been implemented distinguished by two different levels of complexity. To better appreciate the proposed approach, the same problems have been reformulated to be treated by a well known and appreciated optimiser in the field of spacecraft design, Multi–Population Adaptive Inflationary Differential Evolution Algorithm (MP–AIDEA). The final results of the two approaches are compared and commented.**

***Index Terms*—Space systems, Optimisation, Mixed-variable, Variable-size**

I. Introduction

The complexity of the man-made world is increasing and technology is the environment to which men is subjected [15]. Dynamical systems can be chaotic, having a deterministic behaviour but giving a stochastic output. Similarly, the man-made complex systems we are creating are defined by rationality but can produce completely irrational consequences if we are not able to deal with them and understand their complexity. For this reason, in the recent years, the concept of complexity and its application to engineering systems attracted the interest of researchers and engineers. The 21st century, indeed, has been defined the "Systems Century" [4]. Most of the engineering systems are systems-of-systems and/or complex systems with intrinsic interactions and nonlinear dynamics between components [23]. To mitigate these difficulties, modeling and simulation-based systems engineering and model-based systems engineering (MBSE) have started to be used. These methods adopt models of the engineering system that, even if not perfect physical representations and intrinsically "wrong", can be "useful" for virtual prototyping, exploring, and communicating system aspects. Models also allow engineers to quickly and incrementally learn about the system under development before the cost of change gets too high.

This paper presents a new methodology for complex systems design, within MBSE, based on optimisation algorithms in order to meet the growing needs required by space applications. In particular, the paper focuses on the application of the proposed approach for the preliminary design of a spacecraft. Nowadays, the success of a spacecraft design heavily relies on both the choice of the correct technology to be employed and its components settings definition. As for the latter common optimisation algorithms represent a valid assistant that helps the designer to get the maximum of performance given a predefined set of parameters, they are still of too little practical use to help with the former.

Here, we suggest to capture the complexity of a space system (and in general of any engineering system), by means of a network representation as already proposed in [1], [2], [8]–[11], [13], [14], [25]. In this way, the problem formulation is characterised by a hierarchical structure where the possible values and even the existence of variables are conditional on others. Such problem formulation cannot be handled by standard optimisers. Therefore, the proposed methodology relies on Structured-Chromosome Genetic Algorithm optimiser [17]–[19]. This is a variable-size mixed-variable global optimiser. Among others variable-size optimisation strategies, it allows a more flexible formulation of interdependence between variables. The *hidden gene* adaptation of Genetic Algorithm (GA) for the optimisation introduced in [6] activates variables with the introduction of an additional set of variables called *activation genes*. Thus, the selection of the activate set of variables is part of the decision space and then out of the control of the user. In [22] an algorithm to solve metameric variable-size optimisation problems is presented. Though, the formulation only allows having, beside of variables present in all the candidate solutions, a variable number of repetitions of a set of variables gathered in a *template*. Then, the set of variables cannot change between solutions, so it is unappropriated to cope with this kind of problems. An adaptation of GA able to handle hierarchical formulation is used in [21].

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 722734.

However, it cannot handle problem's formulations where the value of a given variable influences which are the set of its dependent variables. Furthermore, it implements operators that are not specifically designed to cope with continuous (numerical continuous), integers (numerical discrete) and categorical (nominal) variables.

In the spacecraft design problem, many of the choices of the different technologies are encoded as categorical variables. To be effective, the operators for mixed-variable problems have to be specifically designed as in the case of SCGA [16].

Two models for the space system, with different order of complexity, have been implemented. To validate the proposed methodology, the performance of SCGA has been compared with the ones obtained with an established optimiser in the field, namely Multi–Population Adaptive Inflationary Differential Evolution Algorithm [5], [7].

The paper is structured as follows. In Section II the models adopted to represent the spacecraft will be described. Then, in Sections III and IV details of the algorithms employed will be given. Sections V and VI will focus on describing the experiments and commenting the results. Finally, a summary of the paper will be given in Section VII.

II. Space System

The two optimisers, SCGA and MP–AIDEA, are applied to the design of system and operations of a Cube-Sat in Low Earth Orbit (LEO). For this problem, two test cases have been developed, f_1 and f_2 adopting two levels of complexity. Specifically f_2 is the higher-complexity model and f_1 is the lower-complexity. Consequently, the formulation of f_2 is more detailed and rich of design parameters.

Both the models have been constructed following the network approach proposed in Refs [1], [2], [8]–[11], [13], [14], [25]. The network representation, indeed, has been shown to be advantageous in order to model complex systems and the interactions between their subsystems and components.

The spacecraft's model is build up by the coupling of the following 6 nodes: Orbit, Telemetry, Tracking and Command Subsystem (TTC), Attitude and Orbit Control Subsystem (AOCS), On-board Data Handling Subsystem (OBDH), Power Subsystem (P) and Payload Subsystem (PL). The objective function is the overall mass of the satellite $f_i(\mathbf{d})$ where $i \in \{1, 2\}$ and $\mathbf{d}$ is the design vector. It is given by the sum of the masses of the subsystems:

$$f_i(\mathbf{d}) = M_{i,ttc} + M_{i,obdh} + M_{i,aocs} + M_{i,pl} + M_{i,p} \quad (1)$$

The calculation of the subsystem masses $M_{i,ttc}$, $M_{i,obdh}$, $M_{i,aocs}$, $M_{i,pl}$ and $M_{i,p}$ will be described in more details in the following sections with reference to both f_1 and f_2.

The test case's models use both continuous (numerical) and categorical (nominal) variables. To note that categorical variables have been mapped in to sequences of integers going from 0 to the number of available choices. Moreover, the number of active parameters varies as function of the values assumed by a subset of the categorical parameters. The list of parameters grouped by node is described in Tables I to VI where it is possible to see: the identification number of each variable (ID), their lower (LB) and upper (UB) bounds, the type (CO for continuous variables, CA for categorical variables), the influence on the activation of dependent parameters, and in which test case they are used.

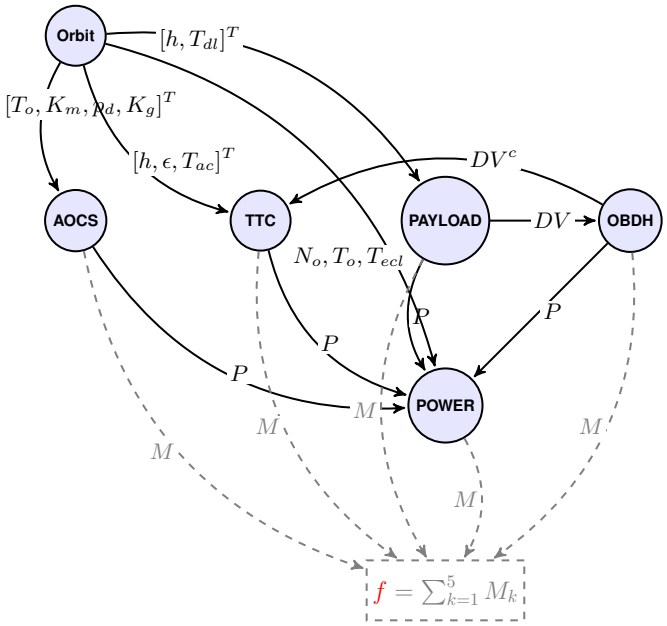


Fig. 1: Representation of the spacecraft as a complex system. The objective function is the overall mass f.

A. System Models

This section presents the mathematical models used to simulate each subsystem and calculate its contribution to the objective function f_i. Part of the models has been described in details in Ref. [12]. Thus, the following subsections will highlight only the differences and novelties of f_1 and f_2.

TABLE I: Design parameters of the orbit node.

Description	ID	LB	UB	Type	Dependencies	Function
Altitude	1	100	1400	CO	–	1 & 2
Elevation agle	2	10	20	CO	–	1 & 2
Inclination	3	0	10	CO	–	1 & 2

1) Orbit: this node is used to determine, based on the input parameters listed in Table I, some of the coupling values shared with other nodes as in Fig. 1. They are the altitude h, the orbit time T_o, the day-light time T_{dl} the gravitational field K_g, the magnetic field K_m, dynamic pressure p_d and the access time T_{ac}.

This node does not refer to any of the cube-sat physical subsystems and indeed it does not contribute to the overall mass of the satellite. In fact it is: $M_{1,orbit} = M_{2,orbit} = 0$, where $M_{1,orbit}$ and $M_{2,orbit}$ are the masses of the orbit subsystem when considered for f_1 and f_2, respectively. To note that the same orbit model is used within both f_1 and f_2.

2) Payload Subsystem: the task of this sub-system is to take images of the Earth during daylight-time T_{dl}, with a

TABLE II: Design parameters of the payload node.

Description	ID	LB	UB	Type	Dependencies	Function
Maximum incidence angle	4	70	75	CO	–	1 & 2
Max along-track ground sampling distance	5	60	80	CO	–	1 & 2
With for square detector	6	20	40	CO	–	1 & 2
Quality factor Q	7	0.5	1	CO	–	1 & 2
Operating wavelength	8	3	4	CO	–	1 & 2

camera, and send them to the OBDH for compression. The design parameters are the variables in Table II. The payload node is influenced by the orbit node through h and T_{dl} as in Fig. 1. Following the design process in [27], the mass ($M_{i,pl}$) and power ($P_{i,pl}$) of the payload node are evaluated through a procedure of scaling based on the *aperture ratio* $R = \frac{A}{A_0}$ where A is the optical aperture of the instrument under design and A_0 the optical aperture of the selected instrument from the catalogue.

Mass and Power are finally defined as:

$$M_{i,pl} = KR^3 M_{pl,0} C_{pl,i,m} + C^*_{pl,i,m} \tag{2}$$

and

$$P_{i,pl} = KR^3 P_{pl,0} C_{pl,i,p} \tag{3}$$

where $K = 2$ if $R < 0.5$ and $K = 1$ otherwise. The terms C and C^* assumes different values for the two test cases. For f_1, $C_{pl,1,m} = C_{pl,1,p} = 1$ and $C^*_{pl,1,m} = 0$, while for f_2, $C_{pl,2,m} = 10+10\sin(\pi+10R)$, $C_{pl,2,p} = 1+10\sin(\pi+10R)$ and $C^*_{pl,2,m} = |d_5 \sin(d_6) - d_7 cos(d_8)|$.

TABLE III: Design parameters of the OBDH node.

Description	ID	LB	UB	Type	Dependencies	Function
Type of OBDH	9	0	3	CA	–	1 & 2
Margin on mass	10	0	20	CO	–	1 & 2
Margin on power	11	0	20	CO	–	1 & 2
Compression rate	12	0.2	0.6	CO	–	1 & 2

3) On-board Data Handling Subsystem: the main purpose of the OBDH is assumed to be the compression and storage of the images coming from the payload. The design parameters are listed in Table III. In both f_1 and f_2 the coupling variable DV, that is an output of the Payload node, is directly proportional with mass and power of OBDH. The type of OBDH hardware, d_9, is here considered as a categorical variable given that it selects a particular component from a list of four unordered possibilities.

An additional term, C_{obdh}, has been multiplyed to mass ($M_{2,obdh}$) and power ($P_{2,obdh}$) of OBDH in f_2. In particular it is: $C_{obdh} = (1 + d_{12})^3$.

4) Attitude and Orbit Control Subsystem: the AOCS is in charge of controlling the orientation of the cube-sat with a three axis stabilisation system. The actuators are reaction wheels, magneto-torquers and thrusters. During the mission, the cube-sat is assumed to be affected by a number of disturbances and then it has to perform some manoeuvres

TABLE IV: Design parameters of the attitude and orbit node.

Description	ID	LB	UB	Type	Dependencies	Function
Reflectance factor	13	0,5	0,7	CA	–	1 & 2
Spacecraft residual dipole	14	0,0005	0,0015	CO	–	1 & 2
Drag coefficient	15	2	10	CO	–	1 & 2
Actuator type for dumping	16	0	1	CA	–	1 & 2
Slew angle (deg)	17	10	60	CO	–	1 & 2
Time for slew manoeuvre (s)	18	10	20	CO	–	1 & 2
Flag reaction wheel	19	0	1	CA	if 0 ⇒ 16 if 1 ⇒ –	1 & 2

TABLE V: Design parameters of the TTC node.

Description	ID	LB	UB	Type	Dependencies	Function
Frequency	20	7	10	CO	–	1 & 2
Modulation type	21	0	7	CA	–	1 & 2
Antenna efficiency	22	0.6	0.9	CO	–	1 & 2
Antenna gain	23	1	5		–	1 & 2
Mass of distribution network	24	0.1	0.5	CO	–	1 & 2
Type amplifier	25	0	1	CA	–	1 & 2
Type antenna	26	0	2	CA	if 0 ⇒ 27, 28 if 1 ⇒ 29 if 2 ⇒ 30, 31, 32	1 & 2
Density copper	27	8000	10000	CO	–	1 & 2
Density dielectric	28	1500	3000	CO	–	1 & 2
Density surface	29	10	20	CO	–	1 & 2
Parameter a	30	2.5	5	CO	–	1 & 2
Parameter b	31	6	12	CO	–	1 & 2
Parameter c	32	2.5	5	CO	–	1 & 2

to compensate the solar radiation pressure T_s, the magnetic torque T_m, the torque due to aerodynamic drag T_a and the gravity gradient torque T_g [2]. The parameter d_{16} decides if the thruster has to be used. If not, d_{19} decides between reaction wheels and magnetic-torques. The mass (M_{act}) and power (P_{act}) of the actuators are computed by interpolation from available real data.

Finally, the evaluated mass and power of the AOCS node are moltiplied by a factor:

$$M_{i,aocs} = M_{act} C_{aocs,i,m} \tag{4}$$

$$P_{i,aocs} = P_{act} C_{aocs,i,p} \tag{5}$$

where $C_{aocs,1,m} = C_{aocs,1,p} = 1$ in f_1 while $C_{aocs,2,m} = 10 + \sin(\pi + T_d)$ and $C_{aocs,2,p} = 10 + \sin(T_d)$ in f_2 and $T_d = T_s + T_m + T_a$ is the sum of all the disturbances.

5) Telemetry, Tracking and Command Subsystem: The TTC is composed of an antenna, an amplified transponder and a Radio Frequency Distribution Unit (RFDU). TTC connects the transmitter antenna mounted on the CubeSat with the receiving antenna on the ground station. The design parameters are in Table V. Also, TTC is coupled with the Orbit node through h, ϵ, T_{ac} and with OBDH through DV^c. The parameter d_{26} is a categorical parameter that selects the type of antenna between three possible options: patch, horn and parabolic antenna. As Table V shows, the value taken by that parameter has also the effect of activating only a subset of $\{d_{27}, d_{28}, d_{29}, d_{30}, d_{31}, d_{32}\}$. It, indeed, select a specific model of the antenna in order to evaluate its mass M_{ant}.

The RFDU mass M_{rfdu} is the variable d_{24}. The amplified transponder mass M_{amp} and the power requirement P_{amp} are derived from available data as a function of the transmitter power P_t (power in output from the antenna).

TABLE VI: Design parameters of the power node.

Description	ID	LB	UB	Type	Dependencies	Function
Type of solar cell	33	0	8	CA	–	1 & 2
Required bus voltage	34	3	5	CO	–	1 & 2
Eps configuration	35	0	1	CA	–	2
Cell packing efficiency	36	0,8	0,9	CO	–	1 & 2
Harness mass factor	37	0,01	0,1	CO	–	1 & 2
Allowable voltage drop	38	1	3	CO	–	1 & 2
Worst case angle of incidence	39	20	40	CO	–	1 & 2
Bus regulation	40	0	1	CA	–	2
Efficiency primary fuel cell	41	0,4	0,6	CO	–	1 & 2
Efficiency secondary fuel cell	42	0,34	0,54	CO	–	1 & 2
Tank figure of merit	43	1700	1900	CO	–	1 & 2
Fuel cell voltage discharge	44	0,75	1,68	CO	–	1 & 2
Fuel cell specific area	45	1500	1600	CO	–	1 & 2
Fuel cell temperature	46	180	220	CO	–	1 & 2
Max tank operating temperature	47	5	10	CO	–	2
Type of energy source	48	0	2	CA	0 ⇒ 33 1 ⇒ 41 43:46 50:59 2 ⇒ 42:47 60:61 3 ⇒ 49	1 & 2
Type primary battery	49	0	11	CA	–	2
FC1 oxidant air filter size	50	5	10	CO	–	2
FC1 oxidant air filter efficiency	51	0,7	0,9	CO	–	2
FC1 oxidant air pump pressure	52	5	20	CO	–	2
FC1 particulate filter density	53	10	50	CO	–	2
FC1 humidification module size	54	5	10	CO	–	2
FC1 check valve pressure	55	5	10	CO	–	2
FC1 converter efficiency	56	0,8	0,9	CO	–	2
FC1 purge particulate filter density	57	12	21	CO	–	2
FC1 pressure transducer efficiency	58	0,7	0,9	CO	–	2
FC1 purge valve efficiency	59	0,69	0,99	CO	–	2
FC2 oxidant air filter size	60	5	10	CO	–	2
FC2 oxidant air filter efficiency	61	0,7	0,9	CO	–	2
Rastrigin parameter 1	62	5	20	CO	–	2
Rastrigin parameter 2	63	10	50	CO	–	2
Rastrigin parameter 3	64	5	10	CO	–	2
Rastrigin parameter 4	65	5	10	CO	–	2
Rastrigin parameter 5	66	0,8	0,9	CO	–	2
Rastrigin parameter 6	67	12	21	CO	–	2
Rastrigin parameter 7	68	0,7	0,9	CO	–	2
Rastrigin parameter 8	69	0,69	0,99	CO	–	2

The mass of the whole TTC system is the sum of its components and the same models are considered for both f_1 and f_2:

$$M_{ttc,1} = M_{ttc,2} = M_{ant} + M_{amp} + M_{rfdu}. \quad (6)$$

6) Power System: The Electrical Power System (EPS) node is composed of a Power Generator (Electric Generator), an Energy Storage and a Power Control and Distribution Unit (PCDU) subsystems. The design parameters are listed in Table VI. As Fig. 1 shows, the EPS is coupled with all the other nodes in the network. The parameter d_{48} (type of power generator) is a categorical parameter selecting a particular type of power generator. For f_1 the power source can be: solar array, primary fuel cell, and secondary fuel cell. For f_2, instead, besides the former possibilities, also primary battery is considered. Each choice brings to a specific model for the mass estimation of the power source and to the activation of different variables as in Table VI. Furthermore, d_{48} has also influence on the energy storage system and in the PCDU. The solar array generates energy only during light-time and requires an energy storage for eclipse periods with a corresponding mass M_{es}. Fuel Cells and Primary Battery, instead, allows for a continuous generation of energy and do not require an energy storage system. The evaluation of the mass of the solar array M_{sa} follows what presented in Ref. [12], however, an additional part $C_{sa,i,m}$ has been added:

$$M_{sa} = A_{sa}\rho_{sa} + C_{sa,i,m}. \quad (7)$$

In order to increase the complexity of f_2, $C_{sa,i,m}$ is considered 0 when $i = 1$ and is a Rastrigin function over the exchange parameters T_{ecl} and T_o and over the design parameters h, bus voltage V_{bus} and temperature margin ΔT when $i = 2$.

The design of the primary and secondary fuel cell proceeds following the suggestions of [3] and uses the parameters from d_{50} to d_{61}.

The design of the mass M_{b1} of the primary battery follows the same procedure used for the secondary battery, however a different catalogue has been considered.

The PCDU is a modular unit. The number of its modules, and thus the mass of the unit, depends on the categorical parameters d_{48} and d_{35} that are respectively the type of power generator and the type of electrical configuration.

Finally, the mass of the EPS is the sum of the individual masses of power generator, energy storage and power conditioning and distribution unit:

$$M_p = M_{pg} + M_{es} + M_{pcdu} \quad (8)$$

where $M_{pg} \in \{M_{sa}, M_{fc1}, M_{fc2}, M_{b1}\}$ and $M_{es} \in \{\emptyset, M_{b2}\}$. To further increase the complexity of f_2, a translated and rotated version of the Rastrigin function has been applied to the categorical parameters of the problem d_{19}, d_{21} and d_{26}. For each possible value assumed by these parameters, different values $d \in \{d_{62}, ..., d_{69}\}$ are activated for the rotated version of the Rastrigin function.

III. SCGA

SCGA is a heuristic algorithm that aims at coping with mixed-variable variable-size optimisation problem by using revised genetic operators. Notably, it's main purpose is to handle problem definitions as flexible as possible. This feature is particularly useful in designing.

In this context, either tunable parameters either actual design choices appear in the problem formulation and have to be encoded as variables. For instance, the choice of the type of antenna to use for the TTC can be encoded as a design variable. As a consequence, depending on the value assumed, it may be necessary to specify or not additional characteristics. Standard approaches require encoding of all possible necessary variables and, using flags or other ad-hoc criteria, interpret which variables have to be neglected and which considered. This limits considerably the possible applications. Moreover, the most obvious drawback, in the case of complex systems, is the dimensionality explosion. Contrary, the flexibility of SCGA allows avoiding redundancies and permits to use straightforward formulations. This is possible thanks to the introduction of the concept of hierarchy in the problem formulation. Some variables can be set as *dependent* by others so their presence and bounds are influenced by their correlated variables. Furthermore, while continuous optimisers, as MP–AIDEA, require tricks as bounding or interpolation to treat integers or nominal categorical variables, SCGA allows

indicating whether a variable belongs to the continuous space, whether it can be assumed as sortable (integer) or not sortable (nominal categorical).

SCGA, as extensively described in [17]–[19], makes use of revised genetic operators that help to make meaningful and effective transformations taking into account the hierarchy in the problem formulation. SCGA adopts the *crossover* and *mutation* operators to evolve the population over the optimisation. The crossover is an operator that exchanges genes between two different chromosomes (parents) to produce two new candidates (children). This aims at combining and transferring the information contained in the parents to the children. In such a way, hopefully, the children will contain the relevant characteristics that originated the performance of their parents. One of the main features that distinguish the crossover implementation in SCGA is that if a variable that has dependent variables is selected to be swapped between two candidates, also all the dependent variables are swapped. This expedient helps at preserving the overall information contained in the selected variables. The second main operator in SCGA is the mutation. This aims at avoiding premature convergence and to mitigate the collapse to the current optimum found adding perturbing the candidates. Depending on the strength of the perturbation, can be introduced a different level of randomness in the population. As a rule of thumb, important perturbations are usually desirable at the beginning of the search whereas small ones are preferred towards the end. However, this is strongly problem dependent and difficult to foresee without a deep knowledge of the problem. In light of these considerations, the mutation employed in SCGA implements a self-adaptive step size mutation that aims an adequate strength of the perturbations autonomously. As in the case of crossover, the mutation is applied also to the variables dependent. The operator acts differently on continuous, integers and categorical variables.

IV. MP-AIDEA

MP–AIDEA [5], [7] is a population-based memetic algorithm for continuous optimisation, It was developed in order to improve the performance of existing Differential Evolution (DE) approaches [24]. The first proposed version, Inflationary Differential Evolution Algorithm (IDEA) [26], hybridised DE with the restarting procedure of the Monotonic Basin Hopping (MBH) algorithm. An adaptive version of IDEA, Adaptive Inflationary Differential Evolution Algorithm (AIDEA) [20], was then proposed to enhance its robustness. It was indeed pointed out that IDEA is higly influenced by the MBH parameters and, like any DE-based algorithm, by the values of the DE parameters. AIDEA is then able to automatically adapt crossover probability (CR) and differential weight (F). The final version of the algorithm, MP–AIDEA [7], evolves multiple populations and allows for different DE strategies. Besides CR and F it also automatically adapts the maximum number of local restart (ius) and the bubble's dimention for the local restart ($\rho_{\mathbf{local}}$). MP–AIDEA was extensively tested over 51 test problems from the single objective global optimisation competitions of the Congress on Evolutionary Computation (CEC) 2005, 2011 and 2014. For more details about the algorithm and its results, please refer to [7].

A. Approach for mixed-variable problems

To apply MP–AIDEA to mixed-variable problems, an approach that converts a mixed-varaible problem into a continuous problem by means of an interpolation procedure is here explored.

Consider, for example, the generic categorical or integer parameter $\tau \in \{\tau_1, \tau_2, ..., \tau_n\}$ with the elements of the set sorted as $\tau_1 < \tau_2 < ... < \tau_n$. The approach for MP–AIDEA replaces τ with the continuous parameter $\tau_c \in [\tau_1, \tau_n]$. For any possible value τ_c we can check in which subset $[\tau_{m-1}, \tau_m] \subset [\tau_1, \tau_n]$ it is included, where $\{\tau_{m-1}, \tau_m\} \in \{\tau_1, \tau_2, ..., \tau_n\}$. The specific part of the objective function f directly depending on τ_c, f_τ, is then evaluated for both the values τ_{m-1} and τ_m and finally a piece-wise interpolation is performed:

$$f_\tau(\tau_c) = f_\tau(\tau_{m-1}) + (\tau_c - \tau_{m-1}) \frac{f_\tau(\tau_m) - f_\tau(\tau_{m-1})}{\tau_m - \tau_{m-1}} \quad (9)$$

The parameter τ_c has been seen to converge, during the optimisation, to a discrete value $\tau \in \{\tau_1, \tau_2, ..., \tau_n\}$.

V. Experimental setup

Both the algorithms SCGA and MP–AIDEA have been used to optimise the functions introduced in Section II. For all the instances 50 independent runs have been performed to have statistically significant results. A budget of $5e5$ objective function evaluations have been used for both the test cases. Because of the available computational resources, a rigorous parameter tuning of the algorithms was not possible. However, the impact of the hyperparameters settings have been mitigated by testing the two algorithms with a number of different settings. The settings used for this study are summarised in Table VII. However, in sake of simplicity and clear visualisation, only the results of the best performing settings (bold in Table VII) have been deeply analysed.

TABLE VII: Algorithms parameters settings.

	Parameter	f_1	f_2
MP-AIDEA	Populations	[**1**, 2]	[**1**, 2]
	Agents	[**30**, 45]	[**30**, 45]
	Max Local restart	[**0**, 10, adaptive]	[0, **5**, adaptive]
	CR	adaptive	adaptive
	F	adaptive	adaptive
	DE strategy	DE/Rand and DE/CurrentToBest	DE/Rand and DE/CurrentToBest
	prob_DE_strategy	0.5	0.5
	fmincon	interiorpoint	interiorpoint
	delta_local	0.1, adaptive	0.1, adaptive
	delta_global	0.1	0.1
	ρ	0.25	0.25
SCGA	popSize	[15, **30**, 60]	[**60**, 70, 80]
	Tournament size [%]	[5, 10, **20**, 30]	[5, 10, **20**, 30]
	localOptGenerations	$round(900/popSize)$	$round(1800/popSize)$
	mutRate size	0.05	0.05
	percCross size	0.3	0.1
	percMut	0.1	0.1
	localOpt	$\lceil 1800/popSize \rceil$	$\lceil 1800/popSize \rceil$

Since MP–AIDEA cannot handle integer, categorical and inactive variables, but only continuous variables, a mask function has been interposed between the optimiser an the

objective function to map the continuous to the mixed-discrete formulation as stated in Section IV-A. This implies a higher number of computations of the mass in the subsystems where categorical or integer variable appear. However, this different computational cost has been neglected in the comparison of the two algorithms.

VI. Results

In this section the results obtained employing SCGA to minimise f_1 and f_2 and the ones obtained using MP–AIDEA are presented, compared, and discussed, with the main focus being on the bests and worsts configurations of the tested strategies. In both the optimisations of f_1 and f_2 SCGA converges faster than MP–AIDEA using either the *best* either the *worst* settings as can be seen from Figs. 3a and 3c. This suggests that SCGA is more robust to the settings chosen than MP–AIDEA when facing problems of this type. Therefore, it represents a wiser choice when the characteristics of the objective function are unknown. This can be clearly seen analysing Fig. 2 that shows, through box-plots, the dispersion of the results of all the instances tested to optimise f_2. As one can see, all the instances of SCGA obtain similar results, contrary to what happens to the ones of MP–AIDEA.

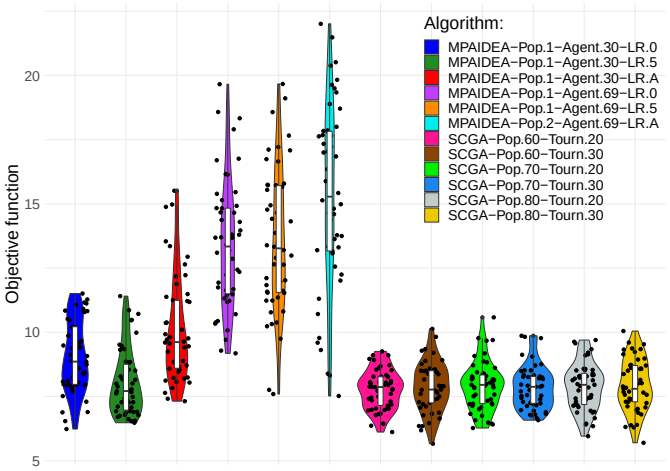


Fig. 2: Box-plot representation of the optimal solutions found minimising f_2 with all the instances of SCGA and MP–AIDEA.

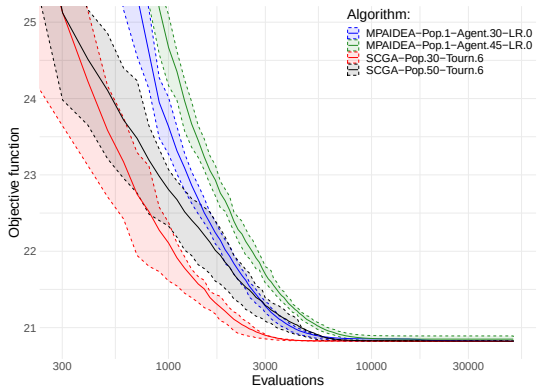
In both cases, SCGA needs less than a fifth of the available budget to find solutions relatively close to the best-known solutions. When optimising f_1, MP–AIDEA converges slightly slower than SCGA. Contrary, when optimising f_2, this converges significantly slower than SCGA. This is due to the greater complexity of the function that emphasises the differences between the two solvers. By looking more closely to the later phase of the optimisation of f_2, it can be seen that the two instances of SCGA perform very similarly and find solutions lower than 10 after 1000 evaluations. Contrary, MP–AIDEA, as well as being slower than SCGA, behaves very differently depending on the settings used. With the bests settings, MP–AIDEA reaches the performance values of SCGA when half of the budget is exhausted, while with the worsts settings, it does not converges within the available budget and don't find solutions lower than 12.5. The statistical validity of these comparisons is tested using the non-parametric Wilcoxon signed-rank test [28]. This test is used to compare the output of experiments (in this case the final best results of the optimisation runs) and assess whether the difference in their mean ranks is significant. Conventionally, the difference between sets of data are considered as significant if the output of the test, the so-called p-value, is lower than 0.05.

The results showed in Table VIII indicate that, when optimising f_1, the difference between the best solutions found by MP–AIDEA, either using the best either the worst configuration, and the ones found by SCGA, using both the configurations, are significant. So can be said that SCGA outperformed MP–AIDEA with both configurations. Whereas, when optimising f_2, the difference the best solutions found by MP–AIDEA using the best configuration and the ones found by SCGA is not significant. So SCGA does not outperform MP–AIDEA using the best settings when minimising f_2 in terms of best solutions found. Though, can be seen from Fig. 3d that finds these results considerably faster.

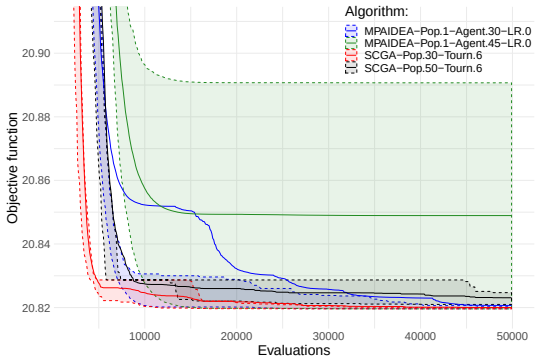
Interesting insights about the behaviour of SCGA and the reasons why it outperforms MP–AIDEA can be found looking at Fig. 4. In this figure the evolution of the best found solution in every generation of the run in which SCGA found the best known solution of f_2 is represented. Specifically, the evolution of each variable is shown separately. Every dot represents the assumed value. Because of the value assumed by the certain variables, namely $d(19), d(26)$ and $d(49)$, some variables do not appear in best solution. This is the case of $d(16), d(27) - d(32), d(33), d(41) - d(47)$ and $d(49) - d(61)$. Firstly, it can be seen that, due to the dedicated operators, the optimum values of categorical variables are usually quickly found. In fact, after every re-initialisation of the population following the local optimisations, SCGA quickly rediscover the optimum values (denoted by the black dotted line) of the categorical variables. Secondly, many variables are indeed deactivated and do not appear in the best solutions found by SCGA. This increases the operations efficiency and so the performance of the optimiser. In fact, they do not contribute to the objective function evaluation and are then redundant and misleading. If classical optimisation strategies, as MP–AIDEA, are used, they are always part of the solution encoding event if they do not contribute to the objective function evaluation and are then redundant. The presence of unnecessary variables, not only brings to useless numerical operations, but can worsen

TABLE VIII: Non-parametric Wilcoxon signed-rank test. Statistically significant differences (pvalue<0.05) are denoted by bold font.

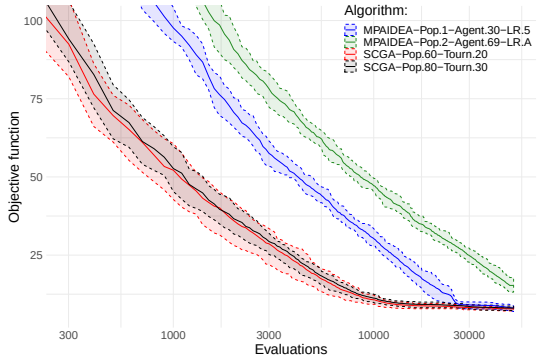
Algorithm	MPAIDEA-Best		MPAIDEA-Worst		SCGA-Best	
	f_1	f_2	f_1	f_2	f_1	f_2
MPAIDEA-Worst	**3,43E-02**	**2,98E-15**	–	–	–	–
SCGA-Best	**1,48E-02**	7,91E-01	**1,41E-02**	**8,58E-16**	–	–
SCGA-Worst	**1,09E-07**	7,64E-01	**4,71E-02**	**3,88E-15**	**4,97E-11**	**4,89E-01**



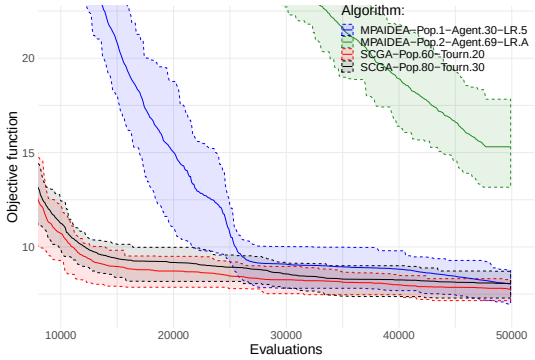
(a) f_1. Complete optimisation history (x axis in log-scale).



(b) f_1. Zoomed in to show final iterations.



(c) f_2. Complete optimisation history (x axis in log-scale).



(d) f_2. Zoomed in to show final iterations.

Fig. 3: Best found solution history of the best and worst configurations of both the tested algorithms. The solid lines represent the mean value of all the 50 independents runs. The dotted lines and the shaded areas show the 25-th and 75-th percentiles.

the behaviour of the algorithm. An example of this is the following. MP–AIDEA, as commented in Section IV, bases its main strength on the the succession of local and global restarts that autonomously balances the exploration and exploitation of the search space. This mechanism heavily riles on the measure of the *diversity* of the population. In case of inactive variables, the optimiser might erroneously consider a population that is already converged as still very variegated. Being these variables ineffective, they will not naturally converge to a value.

VII. Conclusions

This paper presents a new methodology for preliminary spacecraft design using Structured-Chromosome Genetic Algorithms. Adopting the complex network representation, two models of a Cube-Sat in Low Earth Orbit distinguished by two different levels of complexity have been created to model the overall mass of the spacecraft. This performance has been chosen as quantity of interest of an optimisation that was carried out employing two different approaches. In the first, the problem has been framed under a hierarchical problem formulation composed by variables of different types, continuous (numerical) and categorical (nominal). The presence of a hierarchy between variables permitted to dynamically activate and deactivate sets of variables as consequence of the values assumed by the variables creating the first level of the hierarchical structure. The SCGA has been employed as search algorithm. This in fact, contrary to standard optimisers, is able to efficiently manage complex variables encoding with dynamically search spaces. In the second case, the problem has been reformulated to be treated by standard optimisers. Specifically, Multi–Population Adaptive Inflationary Differential Evolution Algorithm has been used to solve it.

The actual potential and benefits of the proposed method have been assessed by comparing the results obtained with the two approaches. Particularly, the results demonstrate the enhancements due to the employment of an optimisers able to reduce the inactive search-space and to treat efficiently different types of variables. SCGA proved to be faster and reliable than standard optimisers and less sensitive to the choice of the settings when coping with problems presenting configurational decisions. As a future work, more complex and detailed models can be implemented and tested. Furthermore, uncertain parameters can be introduced to model more realistic scenarios. From an optimisation standpoint, a restarting strategy will be investigated and tested in SCGA.

References

[1] Carlos Ortega Absil, Gianluca Filippi, Annalisa Riccardi, Massimiliano Vasile, Carlos Ortega, and Absil Phd. A Variance-Based Estimation of the Resilience Indices in the Preliminary Design Optimisation of

Fig. 4: Evolution of the best found result over the optimisation. Colours denote categorical (●) and continuous (●) variables. The optimum values are indicated by (- - -).

Engineering Systems Under Epistemic Uncertainty. Technical report, 2017.

[2] Simone Alicino and Massimiliano Vasile. Evidence-Based Preliminary Design of Spacecraft 6th International Conference on Systems & Concurrent Engineering for Space Applications -. Number October, Vaihingen Campus, University of Stuttgart Germany, 2014.

[3] Kenneth Burke. Fuel Cells for Space Science Applications. (November 2003), 2012.

[4] Charles N. Calvano and Philip John. Systems engineering in an age of complexity. *Systems Engineering*, 7(1):25–34, jan 2004.

[5] Marilena Di Carlo, Massimiliano Vasile, and Edmondo Minisci. MULTI-POPULATION ADAPTIVE INFLATIONARY DIFFERENTIAL EVOLUTION. Technical report.

[6] Shadi A Darani and Ossama Abdelkhalik. Space trajectory optimization using hidden genes genetic algorithms. *Journal of Spacecraft and Rockets*, 55(3):764–774, 2017.

[7] Marilena Di Carlo, Massimiliano Vasile, and Edmondo Minisci. Adaptive multi-population inflationary differential evolution. *Soft Computing*, 2019.

[8] Gianluca Filippi, Mariapia Marchi, Massimiliano Vasile, and Paolo Vercesi. Evidence-Based Robust Optimisation of Space Systems with Evidence Network Models. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, jul 2018.

[9] Gianluca Filippi and Massimiliano Vasile. A Memetic Approach to the Solution of Constrained Min-Max Problems. In *2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings*, pages 506–513, 2019.

[10] Gianluca Filippi and Massimiliano Vasile. A Multi Layer Evidence Network Model for the Design Process of Space Systems under Epistemic Uncertainty. In *EUROGEN*, Guimaraes, 2019.

[11] Gianluca Filippi and Massimiliano Vasile. Evidence-Based Resilience Engineering of Dynamic Space Systems. In *IAC*, Washington, 2019.

[12] Gianluca Filippi, Massimiliano Vasile, Daniel Krpelik, Peter Zeno Korondi, Mariapia Marchi, and Carlo Poloni. Space systems resilience optimisation under epistemic uncertainty. *Acta Astronautica*, 165:195–210, dec 2019.

[13] Gianluca Filippi, Massimiliano Vasile, and Paolo Vercesi. Evidence-Based Robust Optimisation of Space Systems with Evidence Network Models. Rio De Janeiro, 2018.

[14] Gianluca Filippi, M Vasile, P Z Korondi, M Marchi, and C Poloni. ROBUST DESIGN OPTIMISATION OF DYNAMICAL SPACE SYSTEMS. In *8th International Systems & Concurrent Engineering for Space Applications Conference*, Glasgow, 2018.

[15] Umberto Galimberti. *Psiche e techne: l'uomo nell'età della tecnica*, volume 12. Feltrinelli Editore, 2002.

[16] Lorenzo Gentile. LorenzoGentile/SCGA: SCGA second release. Update., January 2020.

[17] Lorenzo Gentile, Cristian Greco, Edmondo Minisci, Thomas Bartz-Beielstein, and Massimiliano Vasile. An optimization approach for designing optimal tracking campaigns for low-resources deep-space missions. In *70th International Astronautical Congress*, 2019.

[18] Lorenzo Gentile, Cristian Greco, Edmondo Minisci, Thomas Bartz-Beielstein, and Massimiliano Vasile. Structured-chromosome ga optimisation for satellite tracking. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1955–1963, 2019.

[19] Cristian Greco, Lorenzo Gentile, Gianluca Filippi, Edmondo Minisci, Massimiliano Vasile, and Thomas Bartz-Beielstein. Autonomous generation of observation schedules for tracking satellites with structured-chromosome ga optimisation. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 497–505. IEEE, 2019.

[20] Edmondo Minisci and Massimiliano Vasile. Adaptive Inflationary Differential Evolution. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1792–1799. IEEE, jul 2014.

[21] Hui Meen Nyew, Ossama Abdelkhalik, and Nilufer Onder. Structured-chromosome evolutionary algorithms for variable-size autonomous interplanetary trajectory planning optimization. *Journal of Aerospace Information Systems*, 12(3):314–328, 2015.

[22] Matthew Lee Ryerkerk. *Metameric representations in evolutionary algorithms*. PhD thesis, Michigan State University, 2018.

[23] Sarah A Sheard and Ali Mostashari. Principles of complex systems for systems engineering. *Systems Engineering*, 12(4):295–311, sep 2009.

[24] Rainer Storn and Kenneth Price. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[25] Massimiliano Vasile, Gianluca Filippi, Carlos Ortega Absil, and Annalisa Riccardi. Fast belief estimation in evidence network models. In *EUROGEN*, sep 2017.

[26] Massimiliano Vasile, Edmondo Minisci, and Marco Locatelli. An inflationary differential evolution algorithm for space trajectory optimization. *IEEE Transactions on Evolutionary Computation*, 15(2):267–281, 2011.

[27] James R. Wertz and Wiley Larson. *B3 - Space Mission Analysis and Design*. 1999.

[28] Frank Wilcoxon, SK Katti, and Roberta A Wilcox. Critical values and probability levels for the wilcoxon rank sum test and the wilcoxon signed rank test. *Selected tables in mathematical statistics*, 1:171–259, 1970.