Allocating Power and Bandwidth in Multibeam Satellite Systems using Particle Swarm Optimization

Nils Pachler, Juan Jose Garau Luis, Markus Guerster, Edward Crawley, Bruce Cameron
Massachusetts Institute of Technology
77 Massachusetts Avenue 33-409
Cambridge, MA 02139
{pachler, garau, guerster, crawley, bcameron}@mit.edu

***Abstract*—In the recent years, communications satellites' payloads have been evolving from static subsystems to highly flexible components. Modern satellites are able to provide four orders of magnitude higher throughput than their predecessors fourty years ago, going from a few Mbps to several hundreds of Gbps. This enhancement in performance is aligned with an increasing highly-variable demand. In order to dynamically and efficiently manage the satellite's resources, an automatic tool is needed.**

This work presents an implementation of a new metaheuristic algorithm based on Particle Swarm Optimization (PSO) to solve the joint power and bandwidth allocation problem. We formulate this problem as a multi-objective approach that considers the different constraints of a communication satellite system. The evaluation function corresponds to a full-RF link budget model that accounts for adaptive coding and modulation techniques as well as multiple types of losses. We benchmark the algorithm using a realistic traffic model provided by a satellite communications operator and under time restrictions present in an operations environment.

The results show a fast convergence of the PSO algorithm, reaching an *admissible* solution in seconds, but fails to reach the global optimum and gets stuck in local optima. This motivates the creation of a hybrid metaheuristic combining the presented PSO with a Genetic Algorithm (GA). We show that this approach dominates the PSO-only both in terms of power consumption and service rate. Furthermore, we also show that the hybrid implementation outperforms a GA-only algorithm for low run-time executions. The hybrid provides up to an 85% power reduction and up to 5% better service rate in this case.

TABLE OF CONTENTS

1. INTRODUCTION

Motivation

Over the last decades, space communications systems have been evolving from static, prefixed payloads, which offered low capacity to the owner, to flexible high throughput devices. While the first generations of communications satellites were only able to manage a few phone lines (few Mbps), a rapid evolution of on-board technology and an increase in power generation led to the development of the so-called High-Throughput Satellites (HTS), able to provide tens and even hundreds of Gbps. The next generation of communication satellites, boosted by digital payloads, advanced modulation and codification techniques, and multi-beam antennas, is expected to surpass the Tbps barrier. This enhancement in performance aims to cover an increasing highly-variable demand. According to recent reports [1] [2], the demand growth is expected to continue over the next years, mainly due to the aim of covering rural areas and the extension of the market to new mobility sectors, such as planes or ships.

These new features, although necessary to satisfy the demand's growth, imply a much higher complexity in the satellite's resource management policies. In the early stages of this industry, the resources such as frequency and power were assigned once, prior to the launch of the satellite. This approach is highly efficient when users consume exactly the predefined amount of data rate at every time. However, the reality is that data demand is bursty, i.e., users have a highly fluctuating consumption. The static allocation of resources implies an over-conservative management of the satellite's resources, specially when the demand is not maximum – which happens frequently.

Current satellite operators are planning to launch new multi-beam constellations capable of managing thousands of fully re-configurable spot-beams while being able to change the frequency and power allocation. Although manual resource allocation was well suited for static management, it is unfeasible for the new generation of satellite communications. An automatic tool has to be developed to fully exploit these novel capabilities. The problem underlying the development of this tool is known as the dynamic resource management (DRM) problem.

Literature review

In the recent years, the DRM problem has become a well-studied topic, both in academia and industry. Within the DRM problem, literature often identifies four resources to allocate [3]: radio-transmitted power, radio-transmission frequency, beam pointing and beam shape. The frequency assignment problem can be further subdivided into the beam's frequency allocation and the bandwidth allocation. In this section we cover the work for the power and bandwidth allocation problems, as well as the research in the joint approach.

The power allocation problem consists of assigning the power level for each transmitting beam within a satellite. The problem is known to be NP-hard and hard to approximate when

978-1-7281-2734-7/20/$31.00

the satellite's power generation is not enough to satisfy the power demand [4], as it can be reduced to a special case of the sum rate maximization problem, which has both characteristics. For this reason, conventional approximation algorithms lead to poor solutions. In the recent years, many approaches have been developed to solve this problem. Authors in [5] propose a convex optimization technique to improve the maximum capacity of the system compared with a uniform and proportional allocation. Work [6] tries to find a better solution with an heuristic based on Lagrange multipliers. [4] uses a hybrid between the Simmulated Annealing and Genetic Algorithm metaheuristics to solve a multi-objective formulation of the problem, while [7] uses a PSO Approach to solve a single-objective formulation, showing the reduced computational complexity of this algorithm to solve the problem. The authors in [8] rely on a Deep Reinforcement Learning-based (DRL) technique to find a solution. All these studies try to distribute the available power pool into different carriers and beams to meet users' demand.

The bandwidth allocation problem is a sub-problem of the frequency asssignment problem and consists of assigning a sub-slot of the beam's frequency to each user. As the power allocation problem, the frequency assignment is known to be NP-hard and hard to approximate [9]. This problem has also been solved using heuristic approaches [10] and convex optimization [11]. These approaches show improvements with respect to a classic water filling method in terms of user fairness or total capacity. Regarding artificial intelligence and machine learning approaches, [12] proposes a DRL methodology, while [13] uses a hybrid neural network combined with a Genetic Algorithm. Both of these algorithms aim to minimize the co-channel interference in a dual-satellite system.

The joint problem has also been studied in the recent literature. Authors in [14] propose an algorithm to optimize resource allocation based on *Signal to Interference plus Noise Ratio* constraints. The results show an improved fairness among users compared to solely power allocation. This problem has also been studied in [15] following a Genetic Algorithm (GA) approach. Both works show significant improvements in power efficiency when allocating joint power and bandwidth compared to only power allocation. None of these works, however, tests the algorithms under high dimensionality and time restricted conditions, crucial for real applications, as they are based on the management of *hundreds of beams* to serve a *highly fluctuating demand*. A fast convergence algorithm is necessary to be able to dynamically adapt the system to this demand.

Literature gap

Based on the literature review, the power and bandwidth allocation have been deeply studied sub-problems. The joint problem has also been researched, but the number of studies is lower. Many approaches rely on metaheuristics implementations to obtain an *admissible* solution in a reasonable time. However, the algorithms proposed for the joint problem have a convergence time that might not be practical for real operational scenarios. For this reason, we consider in this work an algorithm commonly known for its speed and applicability in real case applications [16–18], the Particle Swarm Optimization (PSO). The PSO showed improved results compared to other heuristics in the power allocation problem, which suggests a good potential for the joint power and bandwidth allocation problem.

Objective

The specific purpose of work is to solve the joint power and bandwidth allocation problem for multibeam HTS by formulating the problem as a multi-objective approach and solving it using a PSO-based algorithm. To evaluate the performance of the implementation, we use three different time horizons that aim to test the algorithm in a realistic operational environment under time-restricted conditions.

Overview

This paper is structured as follows: Section 2 introduces the link budget equations and model used in the simulations; Section 3 formulates the problem, including the variables, constraints and metrics which are explained in the different subsections; Section 4 details the PSO implementation of this work; Section 5 presents the results of the algorithm and the comparison with a GA baseline; and Section 6 presents the overall conclusions and possible future work.

2. LINK BUDGET MODEL

This section introduces the equations, proceedings and assumptions relevant to the link budget modelling. This work is based on the model presented in [15].

Satellite communications systems are governed by the link budget equation, which gives the carrier to noise ratio at the receiver's demodulator based on a series of variables that involve the transmitter, receiver, and transmitting medium. Its main use is to compute the necessary transmission power (P_{TX}) to reach the receiver antenna with a sufficient power (C) to decode the signal. The equation used in this work for a single beam is:

$$C = P_{TX} + G_{TX} + G_{RX} - OBO - L\ [dB] \quad (1)$$

$$L = L_{FS} + L_{TX} + L_{RX} + L_p + L_{\text{atm}}\ [dB] \quad (2)$$

where G_{TX} and G_{RX} are the gains in the transmitting (TX) and receiving (RX) antennas, OBO is the power amplifier output back-off and L is the aggregation of the system's losses. These losses include the free space loss (L_{FS}), the losses in the transmitting and receiving chains (L_{TX}, L_{RX}), the losses due to pointing mismatch (L_p), and atmospheric losses (L_{atm}).

To estimate the OBO we followed the method explained in [15]: generate a sequence of 100,000 symbols and assume the OBO equals the peak-to-average power ratio (ratio between the 99.9th percentile power and the average power).

To compute the carrier to noise ratio ($\frac{C}{N_0}$), we first calculate the noise power using the system's temperature (T_{sys}):

$$N_0 = 10 \cdot \log_{10}(k \cdot T_{\text{sys}})\ [dB] \quad (3)$$

where k is the Boltzmann constant. T_{sys} is computed using the Friss formula for noise temperature, which gives the temperature of a system based on the aggregation of the different subsystems and their gains. In this work, we consider five different inputs that affect the global temperature: the atmospheric temperature (T_{atm}), as the noise contributions from the atmospheric layers; the sky temperature (T_{sky}), as the noise contributions from clear sky; the antenna temperature (T_{ant}), as the noise contributions from the antenna elements and its surroundings; the receptor's waveguide temperature (T_{w}), as

the noise contributions from the waveguide; and low noise block temperature (T_{LNB}), as the noise contributions from the low noise amplifier. The temperature of the system, then, is:

$$T_{\text{sys}} = T_{\text{atm}} + T_{\text{sky}} + T_{\text{ant}} + T_{\text{w}} + T_{\text{LNB}}\ [K] \tag{4}$$

The carrier to noise ratio ($\frac{C}{N_0}$) can be computed using equations 1 and 3. Next, we calculate the bit energy to noise ratio ($\frac{E_b}{N_0}$) as:

$$\frac{E_b}{N_0} = \frac{C}{N_0} \cdot \frac{BW}{R_{\text{off}}} \tag{5}$$

where the BW represents the bandwidth assigned to the beam and R_{off} is the beam's offered data rate, i.e., the data rate that the beam provides to the users. This term is computed using:

$$R_{\text{off}} = \frac{BW}{1+\alpha_r} \cdot \Gamma\left(\frac{E_b}{N_0}\right)\ [bps] \tag{6}$$

where α_r is the roll-off factor and Γ is the spectral efficiency of the MODCOD (modulation and coding scheme), which is a function of the bit energy to noise ratio $\frac{E_b}{N_0}$. In this work, we have assumed that adaptive coding and modulation strategies are used, which means that the MODCOD used on each link maximizes the spectral efficiency, while maintaining a threshold link margin β. The chosen MODCOD has to ensure that:

$$\left.\frac{E_b}{N_0}\right|_{MODCOD} + \beta \leq \frac{E_b}{N_0}\ [dB] \tag{7}$$

where $\left.\frac{E_b}{N}\right|_{MODCOD}$ represents the MODCOD energy to noise threshold.

It is interesting to point out that equations 5 and 7 have a circular dependency, as the OBO depends on the MODCOD. In order to solve this dependency, we assume a MODCOD based on Equation 7 without including the OBO in the $\frac{C}{N_0}$ computation. Then, we iterate over Equations 1-6 to find R_{off}, while ensuring Equation 7.

In this paper, we use the MODCOD schemes defined in [19], which correspond to the extension of the second generation standard developed by the Digital Video Broadcast Project (DVB-S2X). This standard is widely used for broadband services, broadcasting and interactive services for space communications and it defines the channel coding, framing structure and modulation schemes. From this standard, we have only used the non-dominated MODCODs in terms of maximizing spectral efficiency and minimizing $\left.\frac{E_b}{N_0}\right|_{MODCOD}$.

3. PROBLEM FORMULATION

This section introduces the variables, based on a single GEO satellite, a set of users and a set of beams; constraints, restrictions imposed by the payload's limitations; and metrics, based on a multi-objective approach; used in this work.

We consider a GEO satellite with N_b beams, each one with a prefixed pointing and band frequency assigned. We assume $N_{u,b}$ users inside each beam's footprint. Then, the problem is to assign power and bandwidth to each user to maximize the service rate and minimize the usage of resources. Throughout the process, the hardware constraint imposed by the satellite must be constantly fulfilled.

Variables

As defined by the problem's objective, we have 2 parameters to optimize for each user: power and bandwidth. The power has to be allocated from the satellite's total power capacity, while the bandwidth per user is a sub-band of the beam's preassigned frequency band. For this problem, the number of variables is thrice the number of users, as we need 1 power variable and 2 bandwidth variables per user (initial and final frequencies).

To reduce the complexity of the problem, we use two simplifications that lead to a reduced set of variables. First, we assume that the $N_{u,b}$ users can be grouped into different types based on their behaviour, i.e., we group the users into N_c groups for each beam. The objective is then to assign power and bandwidth to each group. In other words, each group is assigned a specific *carrier*, and the parameters (power and frequencies) have to be allocated to the carrier instead. The total number of variables (N_{var}) is, then:

$$N_{\text{var}} = 3 \cdot N_c \cdot N_b \tag{8}$$

To further simplify the problem, we assume that all the frequencies available is to be used. Therefore, to divide a range in N_c groups, we only need $N_c - 1$ dividers.

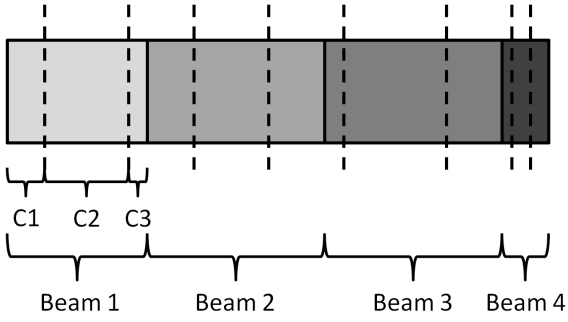

Figure 1. Prefixed division of the frequency pool into 4 beams, marked in different shades, and variable division of the prefixed slots into 3 carriers (C), represented with dashed lines.

Figure 1 graphically shows the division of the total available frequency spectrum into beams and the subdivision of these frequency slots into carriers for an example of 4 beams and 3 carriers per beam. Assuming a specific order in the groups, we only need $N_c - 1$ variables for the bandwidth allocation for each beam. Thus, the number of variables is now:

$$N_{\text{var'}} = \underbrace{N_b \cdot N_c}_{\text{Power allocation}} + \underbrace{N_b \cdot (N_c - 1)}_{\text{Bandwidth allocation}} \tag{9}$$

With these simplification we reduced the problem's dimensionality in a factor of:

$$\frac{N_{\text{var}}}{N_{\text{var'}}} = \frac{3 \cdot N_c \cdot N_b}{N_b \cdot N_c + N_b \cdot (N_c - 1)} = \frac{3N_c}{2N_c - 1} \tag{10}$$

Constraints

We consider three constraints in this work, two for power and one for bandwidth:

• **Maximum Power**: the maximum power provided by the satellite cannot surpass a specific threshold P_{max}, given by the satellite's power generation capacity.

$$\sum_{c=1}^{N_b \cdot N_c} P_c \leq P_{\text{max}} \tag{11}$$

where P_c represents the power assigned to carrier c.

• **Amplifier's Power**: we consider a number of amplifiers N_a, where $N_a \leq N_b$. Each beam is connected to one, and only one, amplifier, while a single amplifier may be connected to different beams. This restricts the total output of the beams, as it cannot surpass the amplifiers limitation $P_{\text{max},a}$.

$$\sum_{c=1}^{N_b \cdot N_c} P_{c,a} \leq P_{\text{max},a} \; \forall \; a \text{ in } \mathcal{A} \tag{12}$$

where $P_{c,a}$ is the power of carrier c if connected to amplifier a, 0 otherwise, and $\mathcal{A}$ represents the set of amplifiers.

• **Minimum bandwidth**: in order to maintain the link between satellite and user, a minimum bandwidth B_{min} for each carrier has to be ensured.

$$B_c \geq B_{\text{min}} \; \forall \text{ c in } \mathcal{C} \tag{13}$$

where B_c represents the bandwidth assigned to carrier c and $\mathcal{C}$ represents the set of carriers.

Metrics

This work uses a multi-objective formulation to assess the goodness of the system both in terms of performance and cost.

The performance metric aims to measure how well the system is satisfying the demand, i.e., how much of the user's requested throughput the system is successfully providing. The chosen metric is the *Unmet Demand* (UD), which aims to measure the non-satisfied demand. This metric was also used in [4, 15] with the name of Unmet System Capacity (USC). The UD is mathematically defined as:

$$\text{UD} = \sum_{c=1}^{N_b \cdot N_c} \max(R_{\text{req},c} - R_{\text{off},c}, 0) \tag{14}$$

where $R_{\text{req},c}$ is the carrier's requested demand, as an aggregation of the individual user's demand, and $R_{\text{off},c}$ is the carrier's offered demand, computed with the link budged equations (Equations 1-7).

The cost metric aims to measure the resource consumption to meet all carrier's demand. As it has been assumed that the all frequency is to be used, the metric for cost will be the total power, defined as:

$$\text{P} = \sum_{c=1}^{N_b \cdot N_c} P_c \tag{15}$$

4. ALGORITHM IMPLEMENTATION

This section details the implementation of the Particle Swarm Optimization (PSO) algorithm used. The different functions and constraint handling, as well as a problem specific heuristic, are explained in the following subsections.

Particle Swarm Optimization

The PSO is a metaheuristic algorithm inspired by the movement of bird flocks. It is a population-based algorithm where the different individuals (called particles) *travel* through the search space to find better positions [20]. A position is a set of coordinates that represents a solution in the search space. The travel action is represented as a velocity at every iteration t (v_t), i.e., a coordinates' change rate. The algorithm moves the particles by giving them an initial position and a velocity at every iteration, aiming to find better positions (better solutions to the problem) based on the current state of the set of particles (also known as swarm).

In general, the PSO particles are kept during the iterations of the algorithm, i.e., new particles are not created. In this work, however, we use the concept of selection, commonly used in evolutionary algorithms but only used in some specific PSO approaches [21], to keep only the best particles after every iteration. The detailed selection function is explained below.

Regarding the velocity at every iteration, many different approaches have been developed throughout the years providing diverse results. However, most of them coincide in the usage of three main pulls: the global pull, the local pull and the inertia.

Global pull

Represents the pull towards the current best position of the swarm. For particle p, it is computed by the formula:

$$G_p = \gamma \cdot g \cdot (x_g - x_p) \tag{16}$$

where γ is a value chosen from a uniform random distribution between 0 and 1, g is the global influence factor, a design parameter, x_g denotes the coordinates of the best solution in the swarm, and x_p denotes the current position of the particle.

PSO as presented in its original paper [20] does not consider multiple objective approaches. The global best position, thus, is assumed to be unique. In multiple objective problems, however, there exists a Pareto Front (PF) of solutions that present a trade-off between the different metrics. From this PF, a *best* particle has to be chosen to *guide* the search. Authors in [22] propose a hypervolume partition and wheel selection technique to find the leader for each particle. The purpose of this division is to maintain the diversity among the population to avoid early convergence. In this work, we prioritize the lower UD region, as it is more interesting from the financial point of view (higher UD usually imply larger economic penalties). For this reason, we assign a fitness (f_p) to each particle in the PF based on their UD:

$$f_p = \frac{\epsilon}{d_p + \epsilon} \tag{17}$$

where ϵ is a design parameter and d_p is computed as:

$$d_p = \frac{\text{UD}_p - \text{UD}_{\text{min}}}{\text{UD}_{\text{max}} - \text{UD}_{\text{min}}} \tag{18}$$

where UD_p is the UD of the particle under consideration and UD_{min} and UD_{max} are the minimum and maximum UD found, respectively. To choose a particle's global best we follow a wheel selection technique, which is based on a random selection given a probability distribution. We assign the

probability of each particle (π_p) depending on this fitness as:

$$\pi_p = \frac{f_p}{\sum_{i=1}^{N_{\text{PF}}} f_i} \tag{19}$$

where N_{PF} is the number of particles in the Pareto Front, as only the particles in the PF are susceptible to be chosen. This technique allows us to avoid the early convergence problems of a single objective formulation while prioritizing the region of interest.

Local pull

Represents the pull towards the best recorded position of the particle. This concept is usually related with the *memory* of the particle: each particle remembers its previous best position and tends to return to it. The formulation for particle p is similar to the global pull's:

$$L_p = \lambda \cdot l \cdot (x_l - x_p) \tag{20}$$

where λ is a value chosen from a uniform random distribution between 0 and 1, l is the local influence factor, a design parameter, x_l denotes the coordinates of the best solution in the particle's memory, and x_p denotes the current position of the particle. Likewise in the case of the global pull, the multi-objective formulation implies non-unique optimal solutions. To solve this issue, we use the single metric (UD) to decide the best solution. In case the two solutions (recorded and current) have the same UD, we prioritize the position with the lowest power.

Inertia

Represents the tendency of a particle to continue in a specific direction, i.e., to keep part of the speed present in the previous iteration. Although not presented in the initial paper, the inertia weight is commonly added to the inertia factor in PSO applications [23]. The inertia pull is computed as:

$$I_p = w \cdot v_{p,t-1} \tag{21}$$

where $v_{p,t-1}$ denotes the velocity of particle p in the previous iteration and w is a design parameter. The velocity of the new iteration ($v_{p,t}$) is computed as the aggregation of the three pulls:

$$v_{p,t} = G_p + L_p + I_p \tag{22}$$

Then, the position is updated:

$$x_{p,t} = x_{p,t-1} + v_{p,t} \tag{23}$$

where $x_{p,t-1}$ and $x_{p,t}$ denote the positions of particle p in the previous and current iterations, respectively.

Particle initialization

In this work, we address the power and bandwidth initialization separately. Regarding power, we consider two valid initialization: all carriers either receive zero or maximum power (carriers receive an equal distribution of power based on the total and amplifiers limitations). This follows the initialization presented in [24], which proposes the creation of two full-swarms with opposed individuals (individuals with opposed initialization) and then does a selection to keep the swarm size constant. The idea behind opposed initialization is to give a better representation of the search space by providing two types of individuals: ones with all the resources allocated and the others with none. This also helps enhancing the exploration capabilities of the algorithm.

On the other hand, each bandwidth split is initialized as a value chosen from a uniform random distribution between 0 and 1. Then, dividers associated with the same beam are sorted. Consequently, the bandwidth allocation for a carrier is given by the difference between two consecutive dividers.

Particle mutation

Although originally developed for evolutionary algorithms, the mutation functions have also been used in PSO applications with satisfactory results [25]. In this work, we use a simple mutation operator to avoid an early convergence for the algorithm. The implementation of the mutation operator is based on two algorithm parameters: first, the mutation probability, which indicates the probability of an specific to undergo the mutation, and second, the variable mutation probability, which is the probability of a single variable (e.g. the power of one carrier) to mutate. The power and bandwidth variables that undergo mutation are redefined as a value from a uniform random distribution between the [0, $P_{max,a}$] and [0, 1], respectively.

Constraint handling

Constraint satisfaction is taken care of after every iteration of the algorithm. Regarding power, we first set all possible negative values to 0. Then, we check the maximum total and amplifier power given by Equations 11 and 12, respectively. In case any of these constraints is not satisfied, we proportionally reduce the power value of the affected carriers to ensure the constraint satisfaction. This is done on a sequential basis for all particles.

For the bandwidth constraint, we only have to ensure that each carrier receives a minimum amount of bandwidth as indicated in Equation 13. To that end, we clip the dividers in the range [0, 1], sort them if necessary, and make sure than the available bandwidth is greater than the minimum threshold for all carriers.

Particle selection

In this work we use the NSGA-II algorithm [26] to keep the best particles of the swarm, i.e., solutions in the PF or close to the PF, and discard the worst particles. After the metric computation, the previous swarm and the new computed swarm are merged, and only half of the population is maintained, using the non-dominated sorting for the selection.

Heuristic

In this work, we use a simple heuristic to guide the PSO algorithm towards the interesting region of the search space. From the problem description, we prioritize the low UD area versus the low power region. Thus, solutions that lead to higher UD are not interesting and slow convergence. To guide the algorithm to low UD we restrict the velocity of the PSO iterations, clipping to zero the velocities that lead to increased power or bandwidth when the UD_c (carrier's UD) is already 0 and the velocities that lead to reduced power or bandwidth when $\text{UD}_c > 0$. We also apply this reasoning to the mutation operator, restricting the mutation search space to the region of interest.

Hybrid PSO-GA

In this work we also present a hybrid algorithm combining the PSO with a GA implementation. The motivation be-

hind this hybrid is to benefit from the fast convergence of PSO and solve its local optimality problems by using GA, which is an asymptotically optimal algorithm. We call this implementation **PSOGA**. After a few iterations of PSO, GA is started with the final PSO population. The population transfer is straightforward, as the particle's position in the PSO algorithm are equivalent to a gene arrays in the GA formulation. In case more individuals are considered in the PSO swarm compared with GA, the lower UD region is prioritized.

Finally, in this work we also present an additional implementation that extends the hybrid with an heuristic by means of which the power variables of the PSO particles are multiplied by a random factor κ, when being transferred to the GA population. We call this implementation **PSOGA+**. This is done to avoid the pull towards the local minima resulting from PSO and to allow the GA a larger search space.

5. RESULTS

This section introduces the traffic model used for the simulations, followed by a summary of the parameters used. Then, the results for different scenarios are presented.

The results are based on a 200 beams distribution on a single GEO satellite over America. We assume that the frequency for each beam is predefined to minimize interference among the beams. The environmental parameters considered in the system are presented in Table 1.

Parameter	Value
T_{atm} [K]	290
T_{sky} [K]	290
T_{ant} [K]	290
T_{w} [K]	290
T_{LNB} [K]	92.29
α_r	0.1
β [dB]	0.5
B_{min}	0.001

Table 1. Environmental parameters

We use a traffic model provided by SES S.A. that has demand data for four types of users per beam, grouped in four carriers (we denote them by A, B, C and D). Based on the number of carriers considered, we define three different scenarios: 2 (A and B), 3 (A, B and C), and 4 (A, B, C and D). The scenario with only one carrier is not considered, as bandwidth can not be split.

For each scenario we compare the performance for the PSO, PSOGA and PSOGA+ implementation, alongside a GA-only implementation that is based on the one presented in [15]. The main change we carried this work is the use of the NSGA-II for the selection function instead of the tournament selection. We also applied the same heuristic as the one presented for the PSO.

For each scenario we consider two demand cases (called low and high demand) that correspond to low and high user activity from the data provided. Since we want to compare the time relative performance, we provide results for 10s, 30s and 5min runs. Each run is independent. In all figures that follow, we show the closest-to-average run from four different executions. We compare the power and UD metrics normalized to the power at minimum UD solution found by all the algorithms and the total demand, respectively.

Parameter	Value
Swarm size	500
Global factor	2
Local factor	2
Inertia weight	0.729844 [27]
Power max speed	2.5%
Bandwidth max speed	5%
Mutation probability	15%
Variables mutated	1/16%
Fitness ϵ	0.01

Table 2. PSO parameter selection

Parameter	Value
Population size	100
Crossing probability	75%
Genes crossed	60%
Alpha blending (crossing)	20%
Mutation probability	15%
% mutated genes	2%

Table 3. GA parameter selection

Parameter	Value
PSO iterations	10
κ	[2-4]

Table 4. PSOGA and PSOGA+ parameter selection

Tables 2, 3 and 4 show the parameters for each of the algorithms compared in this work. For the PSO, the maximum speed parameters are defined as a percentage to the maximum power and bandwidth ($P_{\text{c, a}}$ and 1, respectively). The parameters for each of the individual algorithms in the hybrid implementations are the ones showed in tables 2 and 3.

Table 5 show the full results of this paper for the lowest UD particle comparing the four different implementations under different conditions. As can be observed, GA reaches consistently the lowest UD at the 5 minute mark. In the short run, however, PSO provides dominating solutions compared to the GA-only implementation. Both hybrids dominate PSO in all scenarios and cases. In the short term executions, PSOGA+ consistently achieves lower UD solutions compared with the other implementations, while reaching non-dominated solutions in the long term run compared with GA. Each individual result will be discussed in the following paragraphs.

Scenario 1: 2-carrier

This scenario presents the results for two carriers per beam: A and B. Under these conditions, the low and high demand scenarios have been simulated.

Power												
Algorithm	**2-carrier**				**3-carrier**				**4-carrier**			
	Low		**High**		**Low**		**High**		**Low**		**High**	
	10 s	**5 min**	**10 s**	**5 min**	**10 s**	**5 min**	**10 s**	**5 min**	**10 s**	**5 min**	**10 s**	**5 min**
GA	4.86	0.96	3.94	1.00	2.14	1.00	1.92	1.00	1.75	1.00	1.61	1.00
PSO	4.76	**0.39**	3.89	2.33	2.13	1.38	1.92	0.74	1.75	0.86	1.60	1.08
PSOGA	4.18	0.92	**3.45**	**0.97**	**1.93**	0.42	**1.79**	**0.60**	1.74	0.65	**1.59**	**0.78**
PSOGA+	**4.17**	1.00	3.49	**0.97**	1.94	**0.38**	1.89	0.65	**1.40**	**0.54**	1.60	0.81
Unmet Demand (UD, %)												
Algorithm	**2-carrier**				**3-carrier**				**4-carrier**			
	Low		**High**		**Low**		**High**		**Low**		**High**	
	10 s	**5 min**	**10 s**	**5 min**	**10 s**	**5 min**	**10 s**	**5 min**	**10 s**	**5 min**	**10 s**	**5 min**
GA	5.1	**0.1**	7.5	**3.0**	9.6	**1.6**	21.8	**4.7**	19.5	**2.9**	30.0	**6.0**
PSO	3.6	3.1	5.0	4.7	3.4	2.6	14.0	6.7	13.0	5.8	18.9	9.2
PSOGA	**3.5**	0.3	4.9	3.1	3.6	1.9	12.9	6.8	10.6	4.8	**13.7**	9.4
PSOGA+	3.6	**0.1**	**4.8**	**3.0**	**3.2**	2.0	**12.1**	6.5	**8.1**	4.9	16.2	8.2

Table 5. Results for normalized power and percentage of UD for the lowest UD solution in the three different scenarios and two cases per scenario. The results for the 30s executions have been excluded for simplicity.

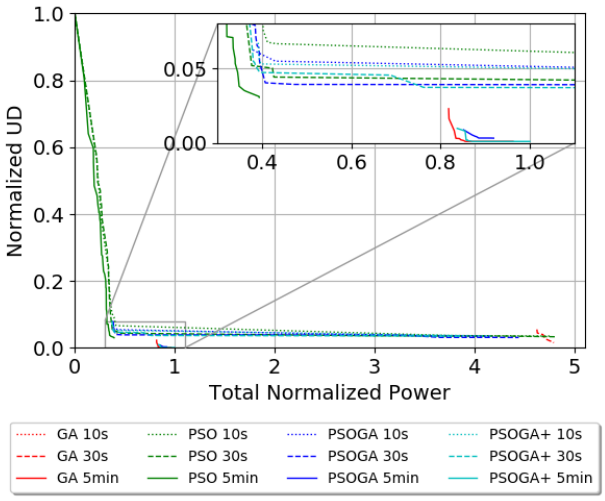

Figure 2. Results for 2-carrier (A and B) and low demand for the four algorithms compared

Low demand—Figure 2 presents the Pareto Front for the 2-carrier and low demand scenario for each of the four implementations compared. The results highlight the behaviour of the non-hybrid algorithms: PSO (green lines) reaches a low power - low UD solution fast (10s to have solutions with a 60% reduction in power versus the power of the lowest UD solution and 7% of UD), but it gets stuck in a local optimum. At the 5 min mark, PSO only gains an extra 4% reduction in UD, while maintaining the same power performance. On the other hand, GA (red lines) starts slowly, having almost 5 times the lowest UD power at the 10s mark. In the long term run, however, the algorithm gets closer to the global optimum, improving the result of PSO, as it reaches almost 0% in UD. Regarding the hybrids, both outperform PSO and GA in the 10s and 30s mark, reaching a power comparable to PSO executions but with only a 6 and a 5% of UD for the PSOGA and PSOGA+ respectively (versus the 7% of PSO). At the 5min mark, PSOGA (blue lines) is unable to outperform GA despite PSO initial boost, while PSOGA+ (cyan lines) reaches a similar UD as the one attained by GA with comparable power.

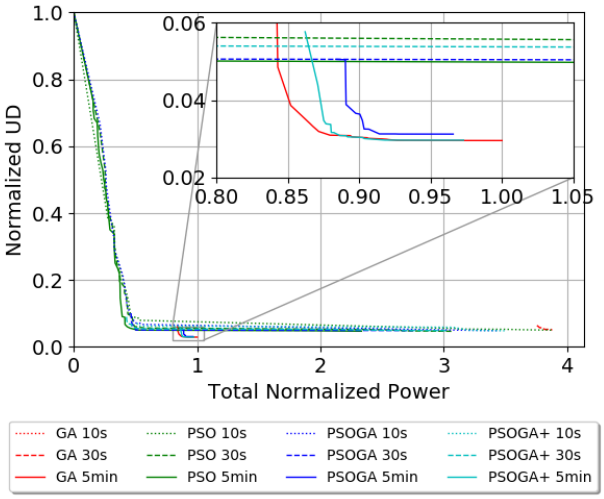

Figure 3. Results for 2-carrier (A and B) and high demand for the four algorithms compared

High demand—Figure 3 presents the PF for the 2-carrier and high demand scenario for each of the four implementations compared. The general behaviour of PSO and GA can be observed once again in these results: PSO rapidly reaches the low power area at the expense of UD while GA starts slowly but ends up converging to a even lower UD after 5 minutes. The hybrids show an improvement in the solution quality in the low run-time case and reach similar solutions in the long term with respect to the GA-only execution, both in terms of power and UD. The PSOGA+, however, reaches lower UD, avoiding the PSO local optima of the PSOGA due to the power changes in the population transfer between PSO and GA executions.

Scenario 2: 3-carrier

This subsection presents the results for the three carrier simulation: A, B, and C. Under these conditions, the low and high demand cases have been simulated.

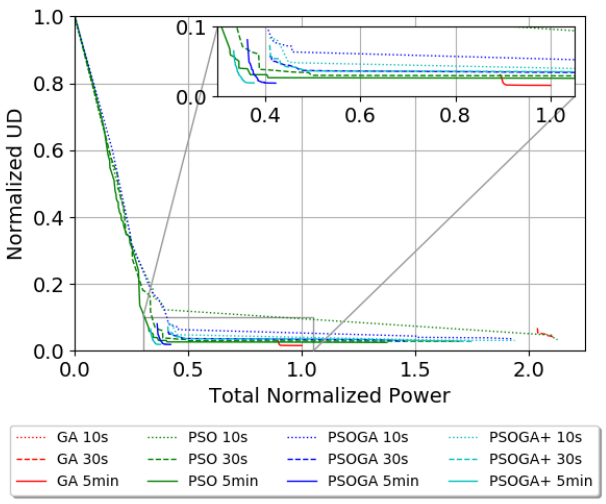

Figure 4. Results for 3-carrier (A, B, and C) and low demand for the four algorithms compared

Low demand—Figure 4 presents the PF for the 3-carrier and low demand scenario for each of the four implementations compared. GA and PSO show clear differences in their final PF. While GA aims for a low UD, giving less importance to power, PSO aims to find an improved trade-off, reaching far lower power solutions, but higher UD. The solutions found by both algorithms are non-dominated. The hybrids try to overcome these differences, clearly outperforming PSO. They also show an improved performance with respect to GA, as they with 60% less power than GA and comparable UD.

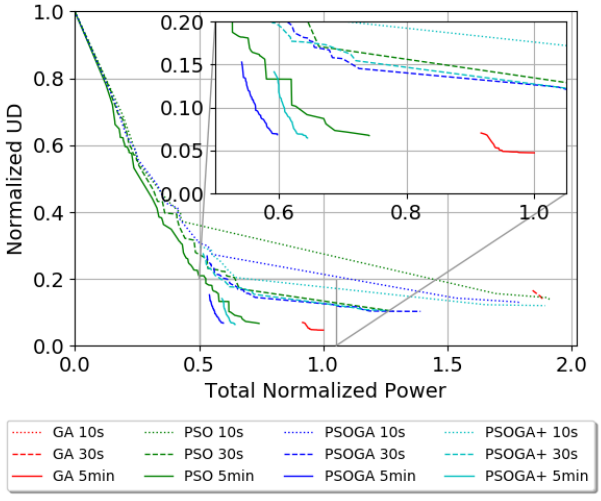

Figure 5. Results for 3-carrier (A, B, and C) and high demand for the four algorithms compared

High demand—Figure 5 presents the Pareto Front for the 3-carrier and high demand scenario for each of the four implementations compared. The same behaviour observed in previous scenarios can be once again identified. In this case, PSO and hybrids reach solutions with similar UD but clearly outperforming GA in the short term run (achieving an almost 10% UD reduction with the same power). In the 5 min mark, both hybrids outperform PSO, but PSOGA reaches lower power solutions than PSOGA+, the latter offering slightly less UD. Both offer non-dominated solutions to GA, which, in turn, is able to find lower UD regions (5% compared to the 6.5 - 6.8% of the PSO and hybrids).

It is noticeable that the Pareto Front angle is more obtuse than the previous scenarios. In those cases, two different behaviours can be observed: in the low power region, the gain in UD heavily varies with the change in power consumption, while in the low UD region much more power is required to have significant changes in UD. In this case, however, the behaviours are not so extreme compared with the previous cases, leading to a more balanced trade-off between UD and power.

Scenario 3: 4-carrier

This scenario presents the results in case of all four carriers. Under these conditions, the low and high demand scenarios have been simulated.

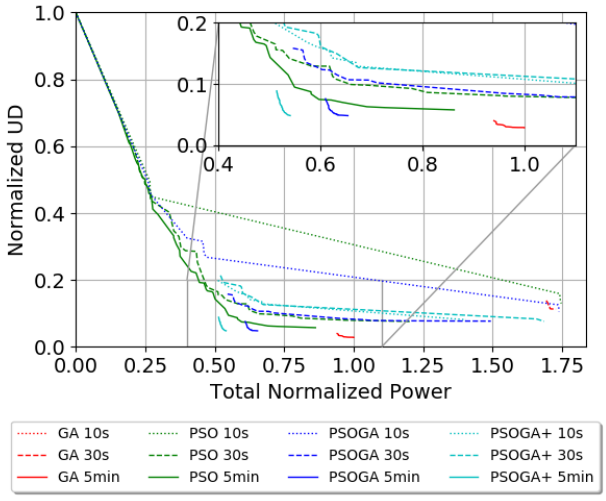

Figure 6. Results for 4-carrier (A, B, C and D) and low demand for the four algorithms compared

Low demand—Figure 6 presents the Pareto Front for the 4-carrier and low demand scenario for each of the four implementations compared. As in all the previous scenarios, GA is consistently able to reach the low UD zone, while PSO gets stuck in a local optimum. Regarding the hybrids, PSOGA is not able to improve the power found by PSO, but reaches a lower UD. On the other hand, PSOGA+ is able to improve in both power and UD with respect to PSO. However, GA still achieves a better UD performance (2% lower in the final iteration), at a power expense (the PSOGA+ achieves a 45% reduction in power). In a real operation environment, if the satellite operator had a 5 minute run-time restriction, it should decide whether the costs of this power reduction could compensate this higher UD. In case the run-time restriction lies within the range of seconds, there is a clear advantage in using PSOGA+.

High demand—Figure 7 presents the Pareto Front for the 4-carrier and high demand scenario for each of the four implementations compared. Once again, the solutions of the algorithms are non-dominated due to the differences in their

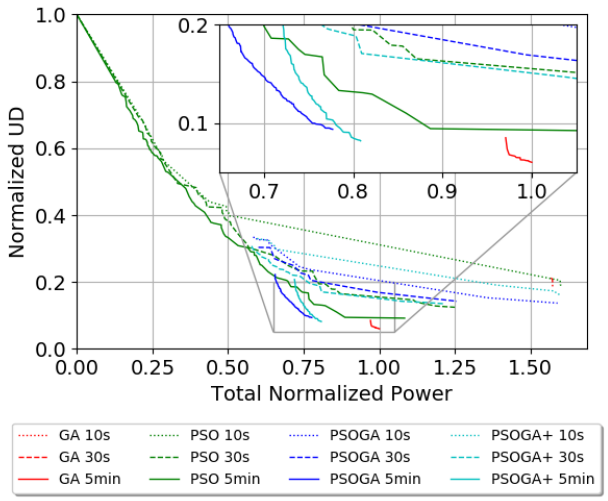

Figure 7. Results for 4-carrier (A, B, C and D) and high demand for the four algorithms compared

search behaviours. In the short run-time executions, PSO and the hybrids reach solutions with up to a 75% reduction in the consumption of power while achieving a 5% reduction in UD compared to GA. In the long term case, however, PSO and the hybrids attain reductions of 20% in power at an expense of between 2% and 3% in UD.

It is important to notice that the trade-off between power and UD reaches a more balanced correlation in this case, being almost linear in the low UD region.

To summarize, PSO showed a fast convergence towards local optima in all scenarios and cases. The results present improvements with respect to a GA-only execution for the low run-time cases, but PSO is unable to outperform GA in the long term run. The hybrids dominated PSO in all cases and provided non-dominated solutions with respect to the GA-only executions in the long term run. In the short term run the hybrid PSOGA+ outperformed all the other algorithms, reaching up to an 85% power reduction and up to 5% less UD than the GA baseline. Therefore this algorithm is suitable for achieving good solutions in highly time-restricted scenarios. On the contrary, when the time is not a limiting factor, the satellite operator should decide whether to use PSOGA+ or the GA-only algorithms, the former providing significantly less power (up to 60% reduction) and the latter attaining less UD (up to 2% less).

6. Conclusions and Future work

In this paper we have presented an implementation of the Particle Swarm Optimization algorithm to the joint power and bandwidth allocation problem for multibeam HTS communications. First, the link budget equation has been detailed and the assumptions considered have been introduced. Then, the problem, including the variables, constraints and metrics, has been formulated. Next, the PSO implementation has been explained and the its functionalities have been covered in detail. Finally, the results of the algorithm on six different scenarios under time restrictions have been presented. The main conclusions of this work are the following:

- The PSO implementation has a fast convergence towards local optima. The results show large improvements with respect to the GA implementation if the computational time is limited. GA, however, consistently outperforms PSO in the long term run.
- In a long run-time case, both hybrid implementations are able to reduce the power consumption (up to 60%), at the expense of an increase in unmet demand (up to 2%), compared with GA-only. Thus, there is no clear dominant solution between GA and the hybrids.
- In short run-time execution cases, PSO and the hybrids reach lower power levels (up to 85% power reduction), while achieving less UD (up to 5% less UD), clearly outperforming GA. With these results, the hybrids are the dominant implementations compared to the GA-only and PSO-only implementations when the computation time is low.

Possible future work may cover:

- Inclusion of the frequency flexibilities for modern constellations: while this paper works under a fixed-frequency model for each beam, modern constellations allow changing the frequency band per beam. This parameter could potentially be included in the algorithm as a decision variable.
- Inclusion of other subproblems from the DRM problem: this work presents the implementation of the joint power and bandwidth allocation problem. Other subproblems inside the DRM, such as the beam placement and beam shape could also be addressed using this algorithm.
- Design robustness and size sensitivity analyses to further understand the behaviour and advantages of all algorithms under different models and scenarios that try to capture the entire variability present in a real satellite communications environment.
- Parameter tuning: while in this work each parameter was optimized individually, without considering interactions, a further work could include a better parameter defining process to increase the performance of each algorithm.

Acknowledgments

The authors would like to thank SES S.A. for their financial and non-financial support during the development of this work. The authors would also like to thank the Higher Interdisciplinary Education Center (CFIS) in the name of Fundació Cellex for their financial support.

References

[1] M. Stanley, "Space: Investment implications of the final frontier," M. Stanley, Tech. Rep., 2017.

[2] S. Networks, "Unleashing the potential of an empowered -world with the launch of o3b mpower," SES, Tech. Rep., 2017.

[3] M. Guerster, J. J. G. Luis, E. Crawley, and B. Cameron, "Problem representation of dynamic resource allocation for flexible high throughput satellities," in *2019 IEEE Aerospace Conference*. IEEE, 2019, pp. 1–8.

[4] A. I. Aravanis, B. S. MR, P.-D. Arapoglou, G. Danoy, P. G. Cottis, and B. Ottersten, "Power allocation in multibeam satellite systems: A two-stage multi-objective optimization," *IEEE Transactions on Wireless Communications*, vol. 14, no. 6, pp. 3171–3182, 2015.

[5] H. Wang, A. Liu, X. Pan, and J. Yang, "Optimization of power allocation for multiusers in multi-spot-beam satellite communication systems," *Mathematical Problems in engineering*, vol. 2014, 2014.

[6] Y. Hong, A. Srinivasan, B. Cheng, L. Hartman, and P. Andreadis, "Optimal power allocation for multiple beam satellite systems," in *2008 IEEE Radio and Wireless Symposium*. IEEE, 2008, pp. 823–826.

[7] F. R. Durand and T. Abrao, "Power allocation in multibeam satellites based on particle swarm optimization," *AEU-International Journal of Electronics and Communications*, vol. 78, pp. 124–133, 2017.

[8] J. J. G. Luis, M. Guerster, I. del Portillo, E. Crawley, and B. Cameron, "Deep reinforcement learning architecture for continuous power allocation in high throughput satellites," *arXiv preprint arXiv:1906.00571*, 2019.

[9] T. Mizuike and Y. Ito, "Optimization of frequency assignment," *IEEE Transactions on Communications*, vol. 37, no. 10, pp. 1031–1041, 1989.

[10] U. Park, H. W. Kim, D. S. Oh, and B.-J. Ku, "A dynamic bandwidth allocation scheme for a multi-spot-beam satellite system," *Etri Journal*, vol. 34, no. 4, pp. 613–616, 2012.

[11] H. Wang, A. Liu, X. Pan, and L. Jia, "Optimal bandwidth allocation for multi-spot-beam satellite communication systems," in *Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*. IEEE, 2013, pp. 2794–2798.

[12] X. Hu, S. Liu, R. Chen, W. Wang, and C. Wang, "A deep reinforcement learning-based framework for dynamic resource allocation in multibeam satellite systems," *IEEE Communications Letters*, vol. 22, no. 8, pp. 1612–1615, 2018.

[13] S. Salcedo-Sanz and C. Bousoño-Calzón, "A hybrid neural-genetic algorithm for the frequency assignment problem in satellite communications," *Applied Intelligence*, vol. 22, no. 3, pp. 207–217, 2005.

[14] J. Lei and M. A. Vazquez-Castro, "Joint power and carrier allocation for the multibeam satellite downlink with individual sinr constraints," in *2010 IEEE International Conference on Communications*. IEEE, 2010, pp. 1–5.

[15] A. Paris, I. Del Portillo, B. Cameron, and E. Crawley, "A genetic algorithm for joint power and bandwidth allocation in multibeam satellite systems," in *2019 IEEE Aerospace Conference*. IEEE, 2019, pp. 1–15.

[16] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, Jun 2007. [Online]. Available: https://doi.org/10.1007/s11721-007-0002-0

[17] R. Poli, "Analysis of the publications on the applications of particle swarm optimisation," *Journal of Artificial Evolution and Applications*, vol. 2008, 2008.

[18] K. E. Parsopoulos and M. N. Vrahatis, *Particle swarm optimization and intelligence: advances and applications*. Information Science Reference New York, 2010.

[19] D. V. B. Project, "Implementation guidelines for the second generation system for broadcasting, interactive services, news gathering and other broadband satellite applications; part 2 - s2 extensions (dvb-s2x)," 2015.

[20] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. Ieee, 1995, pp. 39–43.

[21] P. J. Angeline, "Using selection to improve particle swarm optimization," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360)*. IEEE, 1998, pp. 84–89.

[22] C. C. Coello and M. S. Lechuga, "Mopso: A proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 2. IEEE, 2002, pp. 1051–1056.

[23] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*. IEEE, 1998, pp. 69–73.

[24] H. Jabeen, Z. Jalil, and A. R. Baig, "Opposition based initialization in particle swarm optimization (o-pso)," in *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. ACM, 2009, pp. 2047–2052.

[25] N. Higashi and H. Iba, "Particle swarm optimization with gaussian mutation," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706)*. IEEE, 2003, pp. 72–79.

[26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[27] J. C. Bansal, P. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, "Inertia weight strategies in particle swarm optimization," in *2011 Third world congress on nature and biologically inspired computing*. IEEE, 2011, pp. 633–640.

BIOGRAPHY

Nils Pachler de la Osa *is a dual Bachelor Degree student in Aeronautical Engineerging and Computer Science at the Higher Interdisciplinary Education Centre (CFIS) at the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. He is currently researching at the Massachusetts Institute of Technology (MIT) his degree thesis at the AeroAstro department, under the supervision of Professor Edward Crawley.*

Juan Jose Garau Luis *is a 2nd year graduate student in the Department of Aeronautics and Astronautics at MIT, currently pursuing a M.Sc. degree. In 2017, under the CFIS program, he received two B.Sc. degrees in Telecommunications Engineering and Industrial Engineering from Universitat Politècnica de Catalunya, in Spain. Earlier that year, he was a visiting student at the System Architecture Lab at MIT. He previously worked as a data scientist at Arcvi and at Barcelona Supercomputing Center. His research interests include autonomous systems, Artificial Intelligence, and communication networks.*

Markus Guerster *is a 3rd year graduate student pursuing his Ph.D. degree in the System Architecture Lab of the Department of Aeronautics and Astronautics at MIT. He works on resource allocation for communication satellites with particular focus on a Revenue Management system for operators. He received a M.Sc. degree (2017) with high distinction in Aerospace Engineering as well as a B.Sc. degree (2015) in Mechanical Engineering from the Technical University of Munich.*

Prof. Edward F. Crawley *received an Sc.D. in Aerospace Structures from MIT in 1981. His early research interests centered on structural dynamics, aeroelasticity, and the development of actively controlled and intelligent structures. Recently, Dr. Crawley's research has focused on the domain of the architecture and design of complex systems. From 1996 to 2003 he served as the Department Head of Aeronautics and Astronautics at MIT, leading the strategic realignment of the department. Dr. Crawley is a Fellow of the AIAA and the Royal Aeronautical Society (UK), and is a member of three national academies of engineering. He is the author of numerous journal publications in the AIAA Journal, the ASME Journal, the Journal of Composite Materials, and Acta Astronautica. He received the NASA Public Service Medal. Recently, Prof Crawley was one of the ten members of the presidential committee led by Norman Augustine to study the future of human spaceflight in the US.*

Dr. Bruce Cameron *is a Lecturer in Engineering Systems at MIT and a consultant on platform strategies. At MIT, Dr. Cameron ran the MIT Commonality study, a 16 firm investigation of platforming returns. Dr. Cameron's current clients include Fortune 500 firms in high tech, aerospace, transportation, and consumer goods. Prior to MIT, Bruce worked as an engagement manager at a management consultancy and as a system engineer at MDA Space Systems, and has built hardware currently in orbit. Dr. Cameron received his undergraduate degree from the University of Toronto, and graduate degrees from MIT.*