A Genetic Algorithm for Beam Placement in High-Throughput Satellite Constellations

Nils Pachler, Edward F. Crawley, Bruce G. Cameron
Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, MA, USA
{pachler, crawley, bcameron}@mit.edu

***Abstract*—The new era of satellite communications is approaching as the next generation of constellations is set to launch during the next few years. Digital payloads, steerable beams, and larger constellations will allow operators to serve better and more demand than ever before. However, better service comes at the cost of operational complexity. Manual distribution of satellite resources becomes unfeasible and automatic tools have to be developed to deal with the additional variables.**

From all the different sub-problems that constitute the resource allocation problem in satellite communications, this paper focuses on analyzing the *beam placement* problem in high-dimensional scenarios using graph theory, and, driven by the shown NP-hardness, solving it using a genetic algorithm (GA). To optimize the problem from a system perspective, we develop novel problem-specific metrics (number of beams and number of frequency slots per beam) that intend to reflect a trade-off in the general resource allocation problem.

The results show a fast convergence towards the Pareto-Front that allows us to solve a new-era constellation with thousands of users in around 50 generations. The full system-level analysis proves that the problem-specific metrics developed in this work represent a system-level trade-off independently from allocation algorithms chosen for the other satellite resources. On top of that, when we compare our approach with a previously published heuristic, we show that we reduce both power usage, by between 20% and 40%, and unmet demand, by between 50% and 100%.

***Index Terms*—Satellite communications, beam placement, resource allocation, NP-hard**

I. Introduction

A. Motivation

During the last few years, many established satellite operators (e.g., SES, Telesat) as well as new competitors (e.g., SpaceX, Amazon) have filed applications for constellations to provide global communication services. The new satellites rely on many technological uptakes (e.g., digital payloads and steerable beams) that will allow operators to serve more demand than ever before. However, the new flexibilities involve a higher degree of operational complexity. To optimize their capacity utilization, operators will require automatic tools able to allocate the available satellite resources, such as power and frequency, to the users in real time.

The resource allocation (RA) problem in satellite communications consists of two subproblems: 1, assigning users to beams, and 2, assigning the necessary resources to those beams to fulfil the demand requirements. Due to the large number of users and increased flexibility in the number, position, and characteristics of the beams, the RA is becoming a high-dimensional problem difficult to solve with traditional optimization methods, which are impractical for real scenarios with tens of thousands of users and thousands of beams.

In this work, we present a multi-objective formulation for the beam placement problem (i.e., assign users to beams and place them) and an iterative method to find optimized solutions using a generic algorithm. Finally, we use previously presented algorithms [1] to allocate the other satellite resources (i.e., frequency and power) and show how the novel metrics developed in this work reflect a system-level trade-off independently of the power and frequency algorithms chosen.

B. Literature review

Many works have been recently published regarding the RA for satellite communications. Most of the research focuses on the power allocation [2], [3] and frequency assignment [1], [4] subproblems. Both problems are known to be **NP-hard** [5], [6]. The joint power and frequency allocation has also been studied using different metaheuristic techniques, such as genetic algorithm [7] and particle swarm optimization [8].

Regarding the beam placement problem, different approaches have been proposed: using linear optimization techniques [9], self-organizing feature maps [10], and problem linearizations [11], authors show that their result give optimal solutions in low-beams scenarios (<500 beams). Regarding higher dimensional approaches (>500 beams), in a previous publication [1] we used a heuristic method to reach for the minimum number of beams in a graph-based approach. In this work, we complete the graph formulation presented in [1] by developing novel metrics to better reflect the system trade-offs. In addition, we present a metaheuristic approach to solve the new formulation.

The idea of applying genetic algorithms for graph-based optimization is not new, as it has already been applied to many common graph-based problems: bin packaging problem [12], partitional clustering [13], graph coloring [14], and maximum clique [15]. All these problems are known to be **NP-hard** and, therefore, conventional optimization techniques tend to perform poorly.

978-0-7381-2565-7/21/$31.00 ©2021 IEEE

C. Objective

The objectives of this work are threefold: formulate the beam placement problem as a multi-objective optimization using a graph-based representation, solving it using a genetic algorithm, and assess the quality of the formulation and solution from a system's perspective.

D. Overview

The remainder of the paper is organized as follows: Section II presents the multi-objective formulation for the beam placement problem and proves NP-hardness, Section III introduces the necessary genetic operators to solve the explained formulation using a genetic algorithm, Section IV shows the results of this work in terms of algorithm convergence and system-level performance, and finally, Section V remarks the conclusions of this work.

II. Beam placement formulation

This section presents the beam placement problem formulation as a multi-objective optimization using graph theory and analyzes its complexity. The formulation extends the one presented in [1]. Throughout the work, we assume that we only use a predefined circular beam shape with half cone angle $\frac{}{2}$.

A. Beam placement as a graph problem

First, we define as user any fixed ground station that connects to the satellite constellation to receive or transmit data. The beam placement problem consists of dividing a set of users U into a collection of sub-sets V_i that satisfies the spatiotemporal constraints. Each sub-set V_i is served by one beam, which is described as a highly concentrated satellite signal that covers a part of the Earth. The spatiotemporal constraints on the beam placement problem arise from the geographic limitations of the beam's footprint over the Earth, which restricts the possible users that can be served with that beam. In addition, if the satellite is not geosynchronous, the footprint changes over time.

Given a sub-set of users V_i, their positions $p_g = (x_g; y_g; z_g)$ $8g\ 2\ V_i$, and a satellite position $p_s = (x_s; y_s; z_s)$, we define a satellite cone $(p_c; \ )$ as a cone with origin at p_s, cone angle , and whose axis passes through the point p_c. From this, we define the instant minimum containing cone as the satellite cone that minimizes the maximum satellite angle between the center and each of the users at a specific time t. We define the satellite angle as:

$$_g(t) = \arccos\left(\frac{v_{sg}\ \ v_{sc}}{jv_{sg}jjv_{sc}j}\right) \quad (1)$$

Where $v_{sg} = p_g \ \ p_s$ and $v_{sc} = p_c \ \ p_s$. Note that depends on time as the position of the satellite (p_s) changes over time in non-geosynchronous orbits (NGSO). Then, the instant minimum containing cone can be expressed as:

$$(t) = \min_{p_c} (\max_{g2V} \ _g(t)) \quad (2)$$

Fig. 1: Angle (t) for a sub-set containing two users at different points in time.

Given the dynamics of NGSO, the users on Earth are only visible to the satellites for a portion of the time. To accommodate this flexibility, we discretize time and define T as the collection of time-steps where the sub-set V_i is assigned to satellite s. Then, we can compute the minimum containing cone over all time as:

$$= \max_{t2T} (\min_{p_c} (\max_{g2V} \ _g(t))) \quad (3)$$

As an example, Figure 1 shows a graphical representation of the instant minimum containing cone for a beam covering two users for two separate time instances. As shown, (t) may change over time and is defined as the maximum for all times in T (for this example, $= \max(\ (t_1); \ (t_2))$).

As we only consider circular beams with a fixed shape , once is computed, we can guarantee that users in V_i can be served by the same beam if, and only if, . Given the full set of users U, the beam placement problem consists of finding k subsets (or beams) V_i such that:

$$V_i \setminus V_j = ; \ \text{if } i \ 6\ j \quad 8i; j \quad (4)$$

$$\bigcup_{i=1}^{k} V_i = U \quad (5)$$

$$_i = \max_{t2T} (\min_{p_c} (\max_{g2V_i} \ _g)) \quad 8i \quad (6)$$

B. Objective function

Although any solution that satisfies the constraints is valid for the beam placement problem, not all of them perform optimally in the global resource allocation problem. To optimize the global problem, we need to minimize the amount of resources, mainly power and frequency, that will be consumed by the beams in the solution. To that end, we define the demand of a beam (d_i) as the sum of the demand of each user (d_g): $d_i = \sum_{g2V_i} d_g$. The bandwidth needed to serve a beam can be computed as the ratio between the demand and the spectral efficiency of the beam ($_i$): $b_i = \frac{d_i}{_i}$. Note that $_i$ depends on the modulation and coding (MODCOD) scheme used, which ultimately depends on the power supplied to that beam. This relation implies that power and frequency are inversely proportional when it comes to data-rate: to serve the same demand, we can opt to increase power and decrease frequency or vice-versa. However, this is not captured in the problem definition, as we only want to optimize resources. To resolve this ambiguity, we assume that optimizing resources is equivalent to minimizing frequency usage, while power plays a secondary role for the beam placement problem. This assumption evolves from the increased power flexibility of modern systems compared to the limitations in frequency.

In new constellations, frequency can usually only be allocated by prefixed slots. In that case, we compute the number of slots needed per beam as $_i = \frac{b_i}{c}$ where c is the bandwidth per slot, given by the characteristics of the satellites. The total number of slots consumed is then $= \sum_{i=1}^{k} {}_i$.

In this context, optimizing the usage of resources is equivalent to minimizing the amount of slots consumed, without actually reducing the amount of demand served. The only way to do that is to increase the number of users per beam with the hope that the total number of chunks consumed decreases. However, beams serving more users tend to require more frequency. Given that the frequency is a limited resource, higher demanding beams are more difficult to allocate in the frequency assignment as they require more consecutive slots which imposes additional allocation constraints. Therefore, we want to minimize the amount of slots consumed but keeping the slots per beam low. This is equivalent of minimizing while maximizing the number of beams k. The beam placement optimization problem can be formulated as:

$$
\begin{aligned}
\min \quad & \\
\max \quad & k \\
\text{s.t.} \quad & V_i \cap V_j = \emptyset \quad \text{if } i \neq j \quad \forall i; j \\
& \bigcup_{i=0}^{k} V_i = U \\
& {}_i \qquad \forall i
\end{aligned}
\tag{7}
$$

Note that we keep the two objective separate because the relative importance of each objective can only be assessed based on the results of the full system allocation, which is not known a priori. In a configuration with excess of capacity, having more beams is beneficial to achieve higher flexibility and reduced power losses. On the other hand, in a saturated architecture, having a reduced collection of beams will help increase the maximum amount of users served.

C. Approximations

Given the inherent non-linearities, dependencies with other resource allocation subproblems, and high-dimensionality, the formulation presented is complex to resolve as is. For this reason, we introduce several approximations to make the problem tractable and independent.

1) *Spectral efficiency:* In order to make the beam placement problem independent from the other RA sub-problems, we will assume that all beams will use the same spectral efficiency. As we want to maximize the demand served, we will assume that $= {}_{max}$ where ${}_{max}$ is the maximum spectral efficiency allowed by the modulation and codification techniques. In this work, we use the standard MODCOD schemes from DVB-S2X defined in [16].

2) *Beam center:* From the constraint definition, we note that determining which users can pertain to a specific beam depends on the beam center, which depends on the users served by that beam. To avoid this circular dependency, we will approximate the center of the beam as a weighted sum of the positions of the users (with $w_g = \sum_{h \in V_i} |p_h - p_g|$):

$$
p_{c;i} = \frac{\sum_{g \in V_i} w_g p_g}{\sum_{g \in V_i} w_g}
\tag{8}
$$

D. Complexity

The objective of this subsection is to prove NP-hardness of the beam placement problem by reducing a specific scenario of the formulation to the *minimum clique cover* problem, which has been proven to be NP-hard [17]. In order to analyze complexity and for this subsection only, we will analyze the subproblem:

$$
\begin{aligned}
\min \quad & \\
\text{s.t.} \quad & V_i \cap V_j = \emptyset \quad \text{if } i \neq j \quad \forall i; j \\
& \bigcup_{i=0}^{k} V_i = U \\
& {}_i \qquad \forall i
\end{aligned}
\tag{9}
$$

In the specific case where $\frac{d_g}{c} \quad 1 \; \forall g \in U$, the problem reduces to:

$$
\min \sum_{i=1}^{k} \frac{\sum_{g \in V_i} d_g}{c} = \sum_{i=1}^{k} 1 = k
\tag{10}
$$

Together with these assumptions, we will approximate the constraint ${}_i$ as a pairwise constraint between the users. We state that two users g and h can be in the same beam if:

$$
\begin{aligned}
{}_{gh} &= \arccos\left(\frac{v_{sg} \quad v_{sh}}{|v_{sg}||v_{sh}|}\right) \\
{}_{gh} &= \max_{t \in T} {}_{gh} \quad 2
\end{aligned}
\tag{11}
$$

The original constraint can be understood as the union of the l-dimensional constraints with $l \in \{2;:::;n\}$ where n is the number of users (i.e., the union of the pairwise constraint between two users, the three-way constraint between three users, etc.). The pairwise constraint approximation allows us to construct a graph where there is an edge between nodes g and h if ${}_{gh} \quad 2$. The approximated problem can be formulated as:

$$
\begin{aligned}
\min \quad & k \\
\text{s.t.} \quad & V_i \cap V_j = \emptyset \quad \text{if } i \neq j \quad \forall i; j \\
& \bigcup_{i=0}^{k} V_i = U \\
& {}_{gh} \quad 2 \quad \forall g; h \in V_i
\end{aligned}
\tag{12}
$$

Because the edge condition is entirely dependent on the internal characteristics of the set of users U, this is equivalent to the *minimum clique cover* problem, as we want to minimize the union of cliques on a graph. Henceforth, we proved that some solutions in the original problem are NP-hard and, thus, the original problem as a whole is NP-hard.

Step 1: Choose parents			Step 2: Select crossing points		
P1	ABBCAD	ABCD	P1	ABBCAD	A jBC jD
P2	WXYYZZ	ZYXW	P2	WXYYZZ	ZY jX jW
Step 3: Create offspring			**Step 4: Apply cross**		
P1	ABBCAD	A jBC jD	P1	ABBCAD	A jBC jD
P2	WXYYZZ	ZY jX jW	P2	WXYYZZ	ZY jX jW
O1	ABBCAD	ABCD	O1	ABBCAD	AXBCD
O2	WXYYZZ	ZYXW	O2	WXYYZZ	ZYBCXW
Step 5: Eliminate beams			**Step 6: Repare genes**		
P1	ABBCAD	A jBC jD	P1	ABBCAD	A jBC jD
P2	WXYYZZ	ZY jX jW	P2	WXYYZZ	ZY jX jW
O1	AX ?CAD	AXCD	O1	AXDCAD	AXCD
O2	WBBCZZ	ZBCW	O2	WBBCZZ	ZBCW

Fig. 3: Example of the crossing operator as presented in [12]. P1 and P2 represent the parents and O1 and O2 represent the offspring. Both parts of the genes are represented in order.

Fig. 2: Individual gene coding example for 5 users. The rst part of the individual has as many genes as users (5), while the second part has as many genes as beams (3).

III. ALGORITHM IMPLEMENTATION

This section presents the genetic algorithm used to solve the problem. First, we introduce the concept of genetic algorithms. Then, we present and explain the different genetic operators, which are speci c for graph operations.

A. Genetic Algorithm

Genetic Algorithms (GA) are a subclass of Evolutionary Algorithms (EA), which are based on population evolution to obtain iteratively better and better solutions [18]. Speci cally in GA, the different individuals of the population have the problem variables coded in their genes and evolve towards better individuals using mainly two operators: crossing and mutation. At each iteration, only the best individuals are kept by using a selection process. Since GA have proved to be successful when dealing with other high-dimensional, NP-hard RA sub-problems [7] and common graph-based problems [12]–[15], we adopt this technique to nd optimal solutions for our formulation.

B. Gene codi cation

The gene codi cation and the population crossing operator are closely related with the ones presented in [12]. A full individual is composed of two distinct parts: the mapping of users to beams, in which we code each user with one gene which represents a unique identi er of the beam that is serving that user, and the beams, which is of variable length, has as many genes as beams (k genes), and each gene just contains the unique identi er of the beam. In total, each individual has $n + k$ genes (number of users plus number of beams). Although the same information can be conveyed with n genes, the operators are applied on the second part of the individual, as it represents a smaller solution space. An example with 5 users and 3 beams is shown in Figure 2.

C. Crossing

Crossing is based on the combination of two individuals to create new offspring. Figure 3 shows a 6-user example of the crossing operator used in this work. The main idea behind the crossover operator is explained as follows: 1) two parents are selected from the population, 2) for each parent, two crossover points are randomly selected, 3) at rst, offspring is created as a copy of the parents, 4) the beams between crossover points and all affected users are crossed between offspring (i.e., copied from P1 to O2 and P2 to O1), 5) all modi ed beams that were not crossed are eliminated, and 6) all users that have been left without beams are assigned to existent or new beams. This nal step ensures that the resulting offspring is feasible, as the users left without beams are only assigned to beams that satisfy the formulation's constraints.

D. Mutating

For the mutation operation, we consider three different operators: create beam, absorb users, and destroy beam. To accelerate the search of the GA, mutation is applied several times to the same individual in a single iteration. To focus the algorithm's effort into either maximizing beams or minimizing frequency slots, we de ne two search directions: with a probability p_{dir}, the mutation will only apply create and absorb operations, tending to reduce the number of beams. With probability $1 - p_{dir}$, the mutation will only apply destroy operations, tending to increase the number of beams.

1) *Create:* We create a new beam and assign a random user to it. After that and respecting the constraints, we add other users to that beam based on some probability p_{abs}.

2) *Absorb:* We select a beam and, respecting the constraints, we add close users based on some probability p_{abs}.

3) *Destroy:* We select a beam and destroy it. All the non-assigned users are either absorbed into other beams, or new beams are created to serve them.

Each operator ensures that the new individual created respects all the constraints by only assigning users to valid beams (similarly to the crossing operator, if no beam satis es the constraints, a new beam is created for that user).

E. Selection

The selection of the best population is based on the NSGA-II [19]. Given the GA population, this algorithm searches for the Pareto-Front solutions. If there are more solutions in the Pareto-Front than required population, it selects only a few of them based on crowding distance. If not, it selects the solutions in the Pareto-Front, extracts those solutions, and iterates for the remaining set of solutions, until we obtain as many solutions as required population.

Parameter	Value
Generations	50
Population size	50
Crossing probability	80%
Genes crossed	10%
Mutation probability	20%
Mutated genes	5%
Absorb probability (p_{abs})	25%
Direction probability (p_{dir})	50%

TABLE I: GA parameter selection

IV. RESULTS

This section presents the results of this work: rst, we describe the characteristics of the model; second, we discuss the GA convergence on the beam placement problem; third, we obtain system-level metrics by applying power and frequency allocation algorithms to each beam placement solution and assess the performance from a system's perspective; and fourth, we compare our approach with previous heuristics.

A. Scenario description

For all cases presented below we used a demand model provided by SES which contains information about position and maximum demand for tens of thousands of users distributed around the world. Regarding the constellation characteristics, we set the constellation O3b mPower as a reference [20], which consists of ten satellites at 8062 kilometers operating in equatorial orbits. The half cone angle $\frac{}{2}$ is set to 1.

Regarding the GA, we ran it for 50 generations with 50 individuals. The speci c parameters are summarized in Table I. For all scenarios, GA individuals are initialized at the maximum number of beams (one user per beam).

B. GA convergence

Figure 4 shows the results of the GA's Pareto-Front at 5, 10, 50, and 100 generations. Both metrics have been normalized against the maximum values. Following the metrics presented in Section II, the objective of the approach is to maximize the number of beams while minimizing the number of frequency slots. As can be seen, the results show a signi cant improvement in the Pareto-Front solutions between 5 and 10 generations. Between 10 and 50 generations, however, the gain is reduced, only being relevant in the low-beams, low-frequency region. Even more, from 50 to 100 generations there is almost no enhancement in the Pareto-Front solutions' optimality, as seen in the inset graph. From this, we can conclude that the algorithm rapidly converges and 50 generations are enough to obtain an optimized set of solutions. Table II shows the exact numbers for the point with minimum number of frequency slots for all four experiments considered.

Fig. 4: Convergence of the GA's Pareto-Front for 5, 10, 50, and 100 generations. Note that normalized number of beams equal to 1 (top-right corner) implies one beam per user

Generations	5	10	50	100
Normalized k	0.586	0.288	0.177	0.163
Normalized	0.709	0.546	0.504	0.500

TABLE II: Minimum point for the Genetic Algorithm at 5, 10, 50, and 100 generations. Note that k represents the number of beams and represents the number of frequency slots

C. Full system analysis & Baseline comparison

To get a detailed insight on how the problem-speci c metrics developed in this work translate to the general RA problem, we used two of the frequency assignment algorithms compared in [1]: the frequency assignment heuristic (FAH) approach developed in that work and the random baseline approach. While the FAH technique was shown to perform better in the scenarios considered, the random method gives higher statistical con dence on the complexity of the allocation (i.e., for the random assignment, the results are worse the higher the complexity, which cannot be proved for the FAH approach). Regarding power allocation, we have considered a scenario with no power limitations. Given this condition, the power can be found directly by solving the link budget equations.

The system metrics used are twofold: Transmission Power (by adding the necessary power to serve all the allocated beams) and Unmet Demand (UD, i.e., demand not served). These metrics are commonly used in literature to assess optimality regarding resource allocation [7], [8], [21]. As a baseline, we also included previous results from [1] regarding the beam placement heuristic (BPH) developed in that work.

Figure 5 plots the system metrics for the global resource allocation when using the heuristic and random frequency assignments. In Figure 5a, we can observe that the new approach manages to serve all the demand, while reducing the power around 40%. In the same trend, Figure 5b shows a drastic reduction in UD (around 50%), while also reducing the power consumption by 20%. The fact that the problem-speci c Pareto-Front shown in Figure 4 directly translates into a system-level Pareto-Front in Figure 5 con rms that the metrics developed represent a global trade-off: solutions with higher number of frequency slots will tend to have higher UD and lower power, and vice-versa. Although aiming for the low UD regions is appealing from an operations perspective, the multi-objective approach is necessary due to the 0 UD point. I.e., assuming there exists a solution with 0 UD, reducing the number of beams (or frequency slots) with respect to that solution will only increase power usage. As this point depends on the scenario and frequency assignment and power allocation algorithms, the multi-objective approach is needed to obtain a