POLITECNICO DI TORINO
Repository ISTITUZIONALE

GTOC X: Solution Approach of Team Sapienza-PoliTo

Original
GTOC X: Solution Approach of Team Sapienza-PoliTo / Zavoli, Alessandro; Federici, Lorenzo; Benedikter, Boris; Casalino, Lorenzo; Colasurdo, Guido. - ELETTRONICO. - (2019). (Intervento presentato al convegno AAS/AIAA Astrodynamics Specialist Conference tenutosi a Portland, Maine nel 11-15 Aug. 2019).

Availability:
This version is available at: 11583/2767132 since: 2019-11-14T15:35:11Z

Publisher:
AAS

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

12 October 2024

AAS 19-894

GTOC X: SOLUTION APPROACH OF TEAM SAPIENZA-POLITO

Alessandro Zavoli*, Lorenzo Federici†, Boris Benedikter‡, Lorenzo Casalino§, Guido Colasurdo¶

This paper summarizes the solution approach and the numerical methods developed by the joint team Sapienza University of Rome and Politecnico di Torino (Team Sapienza-PoliTo) in the context of the 10th Global Trajectory Optimization Competition. The proposed method is based on a preliminary partition of the galaxy into several small zones of interest, where partial settlement trees are developed, in order to match a (theoretical) optimal star distribution. A multi-settler stochastic Beam Best-First Search, that exploits a guided multi-star multi-vessel transition logic, is proposed for solving a coverage problem, where the number of stars to capture and their distribution within a zone is assigned. The star-to-star transfers were then optimized through an indirect procedure. A number of refinements, involving *settle time re-optimization*, *explosion*, and *pruning*, were also investigated. The submitted 1013-star solution, as well as an enhanced 1200-point rework, are presented.

INTRODUCTION

This paper provides an overview of the solution approach and the numerical methods developed by the joint team Sapienza University of Rome and Politecnico di Torino (Team Sapienza-PoliTo) in the context of the 10th Global Trajectory Optimization Competition, also known as GTOC X.

The proposed problem, named "Settlers of the Galaxy", represents a unique challenge, where participants are asked to design a settlement tree for colonizing the galaxy, *i.e.*, a pool of one hundred thousand candidate stars, scoring points according to a prescribed merit function $J = J_1 J_2 J_3$, that rewards large settlements, uniformly distributed in space (J_2) and an efficient use of propulsion (J_3). Solutions submitted earlier in the competition are also slightly rewarded (J_1). More precisely:

$$J_2 = \frac{N}{1 + 10^{-4} \cdot N \left(E_r + E_\theta\right)} \tag{1}$$

$$J_3 = \frac{\Delta V_{max}}{\Delta V_{used}} \tag{2}$$

being N the total number of settled stars, E_r and E_θ the uniformity errors in radius r and final polar angle θ_f, respectively, ΔV_{used} the total ΔV employed and ΔV_{max} the maximum admissible ΔV.

A set of rules is provided, which constrains the number of starting vessels, namely Mother Ships and Fast Ships, as well as the number of new vessels, namely Settler Ships, that may depart from any settled star. Maneuverability limits for each type of vessel, *i.e.*, maximum number and magnitude of allowed velocity

*Research Assistant, Department of Mechanical and Aerospace Engineering, Sapienza University of Rome, Via Eudossiana 18 - 00184, Rome, Italy. alessandro.zavoli@uniroma1.it

†PhD student, Department of Mechanical and Aerospace Engineering, Sapienza University of Rome, Via Eudossiana 18 - 00184, Rome, Italy. lorenzo.federici@uniroma1.it

‡PhD student, Department of Mechanical and Aerospace Engineering, Sapienza University of Rome, Via Eudossiana 18 - 00184, Rome, Italy. boris.benedikter@uniroma1.it

§Full Professor, Department of Mechanical and Aerospace Engineering, Turin Polytechnic, Corso Duca degli Abruzzi 24 - 10129, Turin, Italy. lorenzo.casalino@polito.it

¶Full Professor, Department of Mechanical and Aerospace Engineering, Sapienza University of Rome, Via Eudossiana 18 - 00184, Rome, Italy. guido.colasurdo@uniroma1.it

increments, are also assigned. A peculiar, non-keplerian, central-force gravitational model is assumed for the motion of both stars and ships, inspired by the circular motion observed for actual stars in the Milky Way. Further details about the GTOCX problem are reported in the statement paper.[1]

The paper is organized as follows. First, an overview of the global strategy pursued by this team is presented. An indirect method for the ships trajectory optimization is summarized, together with an approximate solution based on the impulsive linear rendezvous problem. A set covering problem is defined, with the aim of attaining an *a priori* chosen star distribution over some zones of the galaxy. A multi-settler stochastic Beam Best-First Search, that exploits a guided multi-star multi-vessel transition logic, is used for solving this novel problem. A number of refinements are then discussed. The submitted 1013-star solution, as well as an enhanced version with a score above 1200 points, are presented.

STRATEGY OVERVIEW

According to the experience acquired in the past GTOC editions,[2–4] the solution approach of Team Sapienza-PoliTo mostly depended on the identified problem features. On the basis of a preliminary analysis of the merit function J, the minimization of the error functions E_r and E_θ was deemed essential for maximizing the score, hence it was assumed as main design goal. Indeed, one should notice that if the error functions E_r and E_θ are small, J increases almost linearly with the number of settled stars N.

A problem reformulation was devised in order not to deal directly with the error functions whilst providing an almost equivalent, yet more practical, way to handle the issue. The galaxy was partitioned according to a 30×32 grid, defined over the (r, θ_f) plane, where the k_r-th horizontal row (or "ring") was defined by $r \in [1 + k_r,\, 2 + k_r]$, with $k_r \in [1,\, 30]$, and the k_θ-th column (or "slice") by $\theta_f \in \left[-\pi + \frac{\pi}{16}(k_\theta - 1),\, -\pi + \frac{\pi}{16}k_\theta\right]$, with $k_\theta \in [1,\, 32]$.

Next, an optimal distribution of stars in each row and column for minimizing the error functions E_r and E_θ was searched for, by assuming the center of each grid cell as a proxy for the actual location of each star inside that cell. This assumption was motivated by the fact that if the star distribution is sufficiently dense and the number of captured stars is sufficiently large, then the local effects would be minimal.

As a result of this analysis, it was found out that the errors could be kept to a minimum by a settlement tree of $512 \times n$ stars, equally-distributed over the 32 columns and linearly-distributed over the 30 rows, with more settled stars at the higher radii than at lower ones. Given the limited team experience in dealing with very large data structures, as well as the short available time, a decision was made to limit the number of settled stars to about 1020, postponing all the possible solution refinements to the last few days of the competition. Randomly sampling stars according to the proposed distribution provided a (reasonable-enough) empirical proof for justifying the proposed problem reformulation. Numerical investigations also suggested that small deviations from the target distribution leading to errors smaller than unity could be accepted without severely hindering the overall score. Conversely, an unsupervised increment of the number of stars would have resulted in a significant score drop.

Given the small number of vessels at Year Zero, the role of Settler Ship multiplication in creating the settlement tree was apparent. Mobility charts were drawn for the Settler Ships in order to provide a graphical tool for assessing the maneuvering capability of a vessel, that is, the location of all reachable arrival stars in the (r, θ_f) plane as a function of departure radius, flight time, and departure epoch.

A spiral occupation was recognized as the only viable option, being a global uniform coverage of the space unfeasible due to the strict time constraints. Settler Ship mobility charts were used for guiding the search of the opening maneuvers, that is, Mother Ships and Fast Ships trajectories departing from Sol at Year Zero. In order to keep the search for the opening trajectories simple enough, Mother Ship trajectories intercepting more than three stars had not been investigated. The target number (1020) of settled stars was so low that two or three "root" settlements per Mother Ship were deemed sufficient for attaining the desired number of settled stars. Opening maneuvers had been concurrently optimized in order to provide "root" settlements in galaxy locations sufficiently far apart to allow a proper "growth" of the spiral settlement tree, without having multiple "sub-trees" originating from different roots competing each other for the same stars. Eventually, 9

non-overlapping zones, that is, sets of contiguous cells of the (r, θ_f) grid, had been identified. In each zone a set-covering problem is posed as finding the optimal settlement tree, for an assigned departure (or "root") star and epoch, that allows to settle a prescribed number of stars per cell within the zone.

A multi-settler stochastic Beam Best-First Search (BBFS) has been developed for solving this unusual problem. The best-first variant was motivated by the need to attain an answer to each set-covering problem as soon as possible, making possible to iteratively adjust all the partial settlement trees. Settlement trees of up to 100 stars were generated in a few minutes on a standard laptop.

A settlement tree with 1013 stars was found at the end of this first phase. A Genetic Algorithm was adopted to slightly refine the transfer times, leading to the submitted solution. An enhanced solution with 1220 stars and an overall score above 1200 points was also investigated by adopting a greedy strategy aimed at increasing the number of settled stars (hence the cost factor J_3) without destroying the spatial distribution (hence with almost constant J_2) and then removing unnecessary colonies. Unfortunately, the competition ended before we were able to submit it.

TRAJECTORY OPTIMIZATION IN THE MILKY WAY

Dynamical Model

A non-Keplerian dynamical model governs the motion of both stars and ships. A central-force law is assumed, which approximately models the circular motion observed for actual stars in our Milky Way. The equations of motion are formulated as:

$$\dot{\boldsymbol{r}} = \boldsymbol{v} \tag{3}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{g} = -\frac{\boldsymbol{r}}{r} f(r) \tag{4}$$

where $\boldsymbol{r}$ and $\boldsymbol{v}$ are, respectively, the position and the velocity vector, and $\boldsymbol{g}$ is the acceleration vector directed towards the galactic center, of modulus $f(r) = v_c^2/r$, with v_c the circular orbit speed for a body at radius r from the galactic center, expressed as the inverse of an 8th-order polynomial function of known coefficients $k_0, \ldots, k_8$, that is:

$$v_c(r) = \frac{1}{\sum_{i=0}^{8} k_i r^i} \tag{5}$$

Indirect Optimization

Only impulsive maneuvers are admitted. According to the vessel type, up to five impulses can be performed. The "basic" maneuver is a two-impulse transfer between two stars with fixed departure and arrival time. The problem at hand is equivalent to Lambert's problem in Keplerian dynamics: the initial velocity components (three unknowns) must be determined in order to match the target position at the arrival time (three boundary conditions). The problem has a single solution and is not subject to optimization. It is worth noting that the trajectories improve when the time of flight is increased, as the competition overall time constraint forces times of flight that are much shorter than the optimal one (e.g., the time competing to the equivalent of a Hohmann transfer).

Despite their simplicity, two-impulse maneuvers are usually optimal or close to optimality. However, impulses may be larger than allowable limits, either on the single impulse magnitude and/or on the overall ΔV value. Trajectories exceeding limits must be discarded for Fast Ships, which can perform at most two impulses. Instead, more impulses are available for Mother Ships (up to 3) and Settler Ships (up to 5), and multiple-impulse trajectories are sought when necessary.

A multiple-impulse trajectory requires the definition of an optimal control problem that can be solved through an indirect approach.[5–7] For the dynamical system of Eqs. (3)-(4), the Hamiltonian $H = \boldsymbol{\lambda}_{\boldsymbol{r}}^T \boldsymbol{v} + \boldsymbol{\lambda}_{\boldsymbol{v}}^T \boldsymbol{g}$ is defined and the Euler-Lagrange equations are derived for the adjoint variables:

$$\dot{\boldsymbol{\lambda}}_{\boldsymbol{r}}^T = -\frac{\partial H}{\partial \boldsymbol{r}} \qquad \dot{\boldsymbol{\lambda}}_{\boldsymbol{v}}^T = -\frac{\partial H}{\partial \boldsymbol{v}} \tag{6}$$

The problem is completed by the boundary conditions on state variables, that must ensure to reach the target and fulfill the ΔV limitations, and the boundary conditions for optimality.

For Settler Ships, up to 4 impulses are considered, since no trajectory requiring 5 impulses has been found. The sum of the impulses magnitude is minimized; constraints on the initial and/or final impulse magnitude are introduced when limits are exceeded. The boundary conditions for optimality require impulses parallel to the velocity adjoint vector (primer vector) and the same primer magnitude at non-constrained impulses.[8] The primer magnitude must be maximum at intermediate impulses when the time is not constrained; however, this would lead to a violation of the minimum timing between maneuvers and the corresponding times must therefore be constrained.

Although Mother Ships can have encounters with multiple stars, they are treated with a similar approach to minimize the overall ΔV. Two classes of trajectories have been exploited in the solution: impulse-intercept-impulse-intercept-impulse-intercept and impulse-impulse-intercept-impulse-intercept. Again constraints on ΔVs are introduced when limits are exceeded. Impulses are always at the minimum allowable time. Intercept times are left free and the conditions for optimality requires that the scalar product of primer and Settlement Pod ΔV is zero.

Approximate Solution: the Minimum-Time Linear Impulsive Rendezvous Problem

A proper optimization of each vessel trajectory is mandatory, as saving ΔV positively impact on J_3, hence on the total score. However, the preliminary definition of the (partial) settlement trees requires the evaluation and the comparison of a possibly enormous amount of transfers connecting any pair of stars, with different departure and arrival epochs. The need for a fast computational tool, able to provide a rough evaluation of transfer time and cost, is thus apparent.

The competition overall time constraint dominates the problem; in order to achieve a final "populous" settlement tree, short (and fast) trajectories between relatively close stars are very likely. This considerations suggest the use of the solution of an impulsive linearized rendezvous problem as a way for rapidly estimate the transfer ΔVs as a function of transfer duration.

Linearized relative motion, that is, Hill-Clohessy-Wiltshire equations,[9] are attained by assuming as reference body the target star, which moves on a circular orbit of radius $\tilde{r}_f$ with angular velocity $\tilde{\omega}_f = v_c(\tilde{r}_f)/\tilde{r}_f$. The natural (linearized) motion of a vessel, relative to the target, can be written in compact matrix form as:

$$\begin{bmatrix} \delta \boldsymbol{r} \\ \delta \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} M(\tau) & N(\tau) \\ S(\tau) & T(\tau) \end{bmatrix} \begin{bmatrix} \delta \boldsymbol{r}_0 \\ \delta \boldsymbol{v}_0 \end{bmatrix} \tag{7}$$

where $\delta \boldsymbol{r}$, $\delta \boldsymbol{v}$ are the position and velocity vector of the vessel in the relative reference frame, respectively, $\tau = (t_f - t_0)$ is the transfer time, and $M(\tau)$, $N(\tau)$, $S(\tau)$, $T(\tau)$ are 3×3 matrices composing the state transition matrix, here reported for the sake of completeness:

$$M = \begin{bmatrix} 4 - 3\cos\tilde{\omega}_f\tau & 0 & 0 \\ 6(\sin\tilde{\omega}_f\tau - \tilde{\omega}_f\tau) & 1 & 0 \\ 0 & 0 & \cos\tilde{\omega}_f\tau \end{bmatrix} \quad N = \begin{bmatrix} \frac{1}{\tilde{\omega}_f}\sin\tilde{\omega}_f\tau & \frac{2}{\tilde{\omega}_f}(1-\cos\tilde{\omega}_f\tau) & 0 \\ -\frac{2}{\tilde{\omega}_f}(1-\cos\tilde{\omega}_f\tau) & \frac{4}{\tilde{\omega}_f}\sin\tilde{\omega}_f\tau - 3\tau & 0 \\ 0 & 0 & \frac{1}{\tilde{\omega}_f}\sin\tilde{\omega}_f\tau \end{bmatrix} \tag{8}$$

$$S = \begin{bmatrix} 3\tilde{\omega}_f\sin\tilde{\omega}_f\tau & 0 & 0 \\ -6\tilde{\omega}_f\sin\tilde{\omega}_f\tau & 0 & 0 \\ 0 & 0 & -\tilde{\omega}_f\sin\tilde{\omega}_f\tau \end{bmatrix} \quad T = \begin{bmatrix} \cos\tilde{\omega}_f\tau & 2\sin\tilde{\omega}_f\tau & 0 \\ -2\sin\tilde{\omega}_f\tau & 4\cos\tilde{\omega}_f\tau - 3 & 0 \\ 0 & 0 & \cos\tilde{\omega}_f\tau \end{bmatrix} \tag{9}$$

Optimal impulsive linear rendezvous problems have been deeply studied.[10] Two-impulse solutions are particularly simple to compute, as no iterative procedure is involved; only a few matrix operations are required. The rendezvous condition requires null relative displacement and velocity at the final time, that is $\delta \boldsymbol{r}_f = 0$ and $\delta \boldsymbol{v}_f = 0$. Let $\delta \boldsymbol{r}_0$, $\delta \boldsymbol{v}_0$ be known initial conditions. The departure velocity $\delta \boldsymbol{v}_0^+$ required to intercept the target at time τ and the corresponding velocity at the intercept $\delta \boldsymbol{v}_f^-$ are evaluated as:

$$\delta \boldsymbol{v}_0^+ = -N^{-1}(\tau)\left(M(\tau)\delta \boldsymbol{r}_0\right) \tag{10}$$

$$\delta \boldsymbol{v}_f^- = S(\tau)\delta \boldsymbol{r}_0 + T(\tau)\delta \boldsymbol{v}_0^+ \tag{11}$$

The total transfer cost is readily computed as :

$$\Delta V_{tot} = \Delta V_0 + \Delta V_f = \left\|\delta \boldsymbol{v}_0^+ - \delta \boldsymbol{v}_0\right\| + \left\|\delta \boldsymbol{v}_f^-\right\| \tag{12}$$

Discrepancies between full-dynamics and linear solutions were minimal in most cases of interest (below $2 \div 5$ km/s). Indeed, as for the full-dynamics problem, impulse magnitude limits are likely to be violated. Therefore, time-minimum two-impulse transfer that satisfy the constraints on both single and overall impulse magnitude have been searched for. A simple line search on a time grid with 1 Myr step provides reasonable values of the minimum admissible rendezvous time τ for any assigned pair of stars and departure epoch.

Eventually, given a starting star s and a starting epoch t_s, a minimum-time Hill neighborhood $\mathcal{N}_s = \mathcal{N}(s, t_s)$ has been defined as the set of all stars that can be reached in the available time window departing from s at time t_s, while respecting the vessel ΔV limits on both single impulse magnitude ($\Delta V_i < \Delta V_i^{max}$) and their sum ($\Delta V_{tot} < \Delta V_{tot}^{max}$). In order to account for possible improvements that would inevitably come with the full-optimization procedure, ΔV constraints are (moderately) relaxed, with the tolerance increasing together with the flight time. Minimum-time hill neighborhoods have been extensively used in the construction of partial settlements trees. The adopted settings, based on a trial and error procedure, are summarized in Table 1.

Table 1: Acceptance thresholds for the generation of minimum-time Hill neighborhoods.

	ΔV_i^{max} [km/s]	ΔV_{tot}^{max} [km/s]
$\tau <$ 4 Myr	170	340
$<$ 9 Myr	175	350
$<$ 15 Myr	190	360
otherwise	300	400

SEARCHING FOR SETTLEMENT TREES

A major role in the problem solution is played by the creation of (partial) settlements trees, that is, a web of vessel trajectories connecting pairs of stars, with origin at one given *root* star, either settled by a Mother Ship or a Fast Ship. The search for settlements trees is aimed at providing a proper spatial distribution, that keeps the error functions sufficiently low, while ensuring the feasibility of all transfers. To attain this result, a number of *zones*, that is, sets of contiguous cells of the 30×32 grid defined over the (r, θ_f) plane, are identified. Each zone Z encompasses a prescribed number of adjacent rows $K_r^Z = \{k_{r,min}^Z, \ldots, k_{r,max}^Z\}$ and columns $K_\theta^Z = \{k_{\theta,min}^Z, \ldots, k_{\theta,max}^Z\}$.

In each zone Z, a *set-covering problem* is issued. The goal is to find, if it exists, a settlement tree, starting at a given root star and epoch, that realizes a spatial occupancy as close as possible to a target one. The target *occupancy distribution* in a zone could be provided either on a cell-by-cell basis (i.e., by imposing the number of stars to settle in each cell), or on a row and column basis (if the stars to settle are imposed in each row and column). Having in mind the (preliminary studied) global 1020-star occupancy, the row/column target formulation appears simpler to devise and implement. Unfortunately, this formulation results in a very broad, hence extremely slow, search. Instead, since a same row/column distribution can be achieved with several (different) cell-by-cell distributions, the use of a formulation by cells narrows down the search, speeding up the solution process. In addition, a cell-by-cell distribution allows to completely remove all transfers ending to stars belonging to a cell where the occupancy requirement has already be filled (both in case the target occupancy is zero, or the target number of stars in that cell has already been settled), so simplifying further the search. These reasons led us to prefer an objective distribution by cells instead of by rows and columns, in each of the identified zones. As a minor drawback, a few iterations over the target distribution matrix of each zone were needed, in order to produce an actual overall distribution complying with the 1020-star desired row/column occupancy.

Set covering problem definition

The set-covering problem poses as a "standard" *search problem*,[11] where the objective is to look for a sequence of *actions*, *i.e.*, transitions from a state to the next one (or *successor*), that, starting from a known initial state, reaches a *goal* state, that is, an optimal state according to a given *performance measure*.

The set-covering problem can be formally defined as a search problem, as soon as the following components are specified: (i) the state space of the problem, that is, the set of all states reachable from the initial state by any sequence of actions, (ii) a performance measure, to compare the states each other and push the search towards the goal, and (iii) the transition model, which allows to generate the successors of a given state.

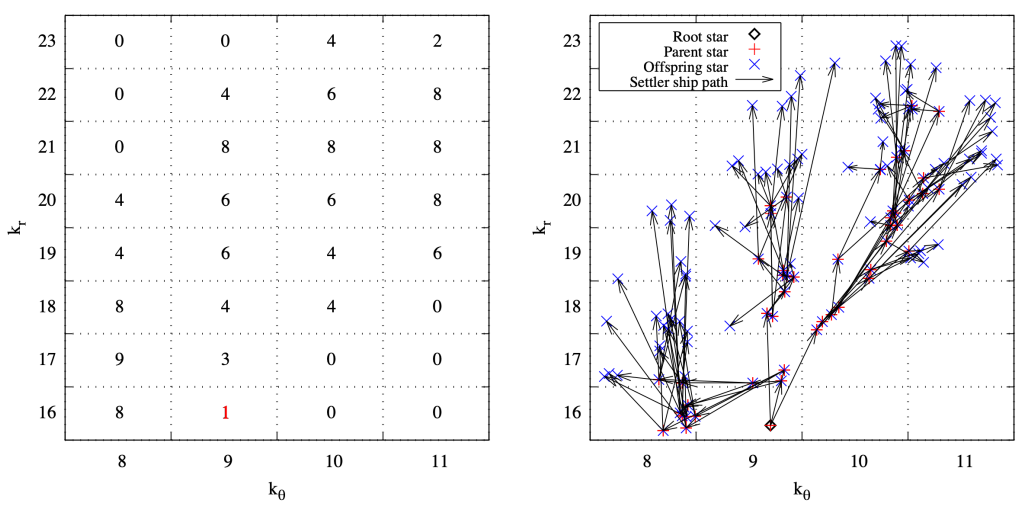
State Space. Each state in the state space represents a settlement tree, *i.e.*, an un-ordered list of stars $\mathcal{S} = \{s_0, s_1, \ldots, s_n\}$ settled at times $t_0, t_1, \ldots t_n$. The k-th star s_k is labeled with information about: i) the star ID, ID_k, ii) the "parent" star, s_p, where the in-bound Settler Ship comes from (with $p \in [0, n-1]$), iii) a maximum of three "offspring" stars, $\mathcal{O}_k = \bigcup_{i=1}^{l \leq 3} s_{o_i}$, reached by the out-bound Settler Ships (with $o_i \in [1, n]\ \forall i$), and iv) the star settlement epoch t_k. Since only the minimum (2-Myr) waiting time has been considered at each settled star, the evaluation of the transfer time from a parent star to any of its offspring is straightforward, once the settlement epochs are known. The initial state of the search procedure is the settlement tree made up only by the root star s_0, settled at a known epoch t_0.

Performance Measure. The explored states are ranked according to the following cost function (to be minimized):

$$\phi = \|H - G\| + \frac{t_n}{100} \tag{13}$$

H and G are two matrices such that, for each cell $(k_r,\ k_\theta)$ composing the zone Z under search, $H(k_r,\ k_\theta)$ (respectively, $G(k_r,\ k_\theta)$) reports the actual (respectively, target) number of stars settled (respectively, to settle) in that cell, and t_n is the final settling time in Myr. Being $\|H - G\|$ an integer value and $\frac{t_n}{100} < 1$, the final settling time becomes important only once the target distribution has been achieved.

A visual representation in the $(r,\ \theta_f)$ plane of the best settlement tree (*i.e.*, state) obtained in the zone labeled $M11$, with target distribution matrix G reported in Figure 1a, is proposed in Figure 1b, where time information is neglected for the sake of clarity.



(a) Distribution matrix G. The red number indicates the root star.

(b) Settlement tree obtained as a result of the search procedure.

Figure 1: Zone $M11$: target distribution (a), correspondent settlement tree (b).

Transition Model. New states, or *successors*, are built upon previously evaluated ones by using an incremental mechanism, that is, by settling additional stars using the available vessels from any star composing the (already constructed) tree.

Let s_k be a "fertile" settled star belonging to the *ancestor* state $\mathcal{S}$, *i.e.*, a settled star with at least one available vessel. A reachable minimum-time neighborhood $\mathcal{N}_k^\star$ of s_k is defined as the set of all (unsettled) stars, in the considered zone, reachable from s_k in the least possible time which produces admissible ΔVs. In order to keep the computational time reasonably low, the (true) neighborhood, obtained by solving the optimization problem involving, for each arrival star, the determination of a minimum-time, ΔV-constrained, n-impulse transfer in the full dynamical model, is replaced by the minimum-time Hill neighborhood $\mathcal{N}_k$, where the two-impulse solution of the linear rendezvous problem is used. Neighborhoods are computed offline, with departure and arrival epochs discretized over a uniform time-grid with a 1-Myr step size. For the sake of simplicity, non-settled stars located in cells where current occupancy matches the target distribution G are not included in any neighborhood, thus speeding up the computation.

Several transition models may be defined for the problem under investigation. At a first attempt, a *one-vessel* transition model was considered, that is, any successor state $\mathcal{S}'$ is obtained by settling one additional star s_o that belongs to the neighborhood of one of the n_f fertile stars of the ancestor $\mathcal{S}$:

$$\mathcal{S}' = \mathcal{S} \cup \{s_o \in \mathcal{N}_k \mid s_k \in \mathcal{S} \wedge |\mathcal{O}_k| < 3\} \tag{14}$$

Up to $n_f\,|\mathcal{N}_k|$ different successor states per ancestor $\mathcal{S}$ may be constructed with this model, being $|\mathcal{N}_k|$ the number of unsettled stars composing the minimum-time Hill neighborhood of a fertile star $s_k \in \mathcal{S}$.

While particularly simple, this transition model was not really effective: indeed, the probability to obtain duplicate (successor) states starting from different ancestor states is quite high; as a consequence, the search algorithm has to keep in memory a great number of states very similar (if not equal) each other, and must perform a high number of steps (*i.e.*, transitions) in the search space in order to reach the target distribution, thus resulting in a very slow procedure.

Therefore, a *multi-star multi-vessel* transition model has been devised, that is, in a single transition multiple (up to three) vessels may depart from any number of fertile stars in $\mathcal{S}$, "concurrently" settling multiple stars that will be added to $\mathcal{S}$ in order to generate the successor state $\mathcal{S}'$:

$$\mathcal{S}' = \mathcal{S} \cup \bigcup_{(i \le n_f)} \bigcup_{(j \le 3-|\mathcal{O}_{k_i}|)} \left\{s_{o_j} \in \mathcal{N}_{k_i} \mid s_{k_i} \in \mathcal{S} \wedge |\mathcal{O}_{k_i}| < 3\right\} \tag{15}$$

The number of successors that can be generated in this way is much larger, as it depends on all possible combinations of departure and arrival stars, letting aside the travel time, which is assumed to be the minimum. This number increases exponentially as the settlement tree grows; it also increases as the star density increases, that is, at lower radii (less then 5 kpc).

Both memory and time constraints prohibit the generation of all possible successors. A constructive heuristic is proposed for generating up to $b_{f,max}$ successors from a single ancestor, by using a well-defined set of rules for the selection of: a) fertile (departure) stars, b) the number of out-bound vessels from each departure star, and c) the neighbor (arrival) star for each vessel.

An iterative procedure has been devised to built the set of successors. Let $\mathcal{L}^i$ be a set containing the successors generated at the i-th iteration ($\mathcal{L}^0 = \mathcal{S}$ at the first iteration).

1. First, the fertile stars in the ancestor state $\mathcal{S}$ are ranked by using as "fitness" the number of missing stars (*i.e.*, still to settle) in their own cell (k_r, k_θ) and in the 8 adjacent cells:

$$f = \sum_{p=-1}^{1} \sum_{q=-1}^{1} \left[G(k_r + p, k_\theta + q) - H(k_r + p, k_\theta + q)\right] \tag{16}$$

Then, a fertile star s_k is selected among the candidates (and removed from the list) by means of a biased sampling without replacement, proportional to f. In such a way, it is promoted the creation of branches directed towards the most empty cells of the zone.

2. Now, $m_k \left|\mathcal{L}^i\right|$ new successors states are created, by appending m_k differently generated set of offspring of star s_k to each state in $\mathcal{L}^i$. The parameter m_k is set according to the formula:

$$m_k \quad = \begin{cases} 1.2\,\hat{m} & \text{if } \; f_k > \bar{f} \\ \hat{m} & \text{otherwise} \end{cases} \tag{17}$$

with $\hat{m} \in [\hat{m}_0, \hat{m}_0 + 4]$ an user-defined value, either constant or randomly sampled at any iteration, where the value $\hat{m}_0 = \sqrt[n_f]{b_{f,max}}$ ensures a symmetrical successor tree; $\bar{f}$, instead, is the average fitness of the fertile stars. The idea is to stress the exploration of empty cells by allowing a 20% more successors states to be generated from stars with a fitness larger than the average.

3. The number of offspring β_k of the star s_k to insert in each new successor is randomly sampled in $[1, 3 - |\mathcal{O}_k|]$, according to an uneven probability distribution that aims at increasing the number of settled stars as soon as possible.

4. Finally, each offspring star s_o must be selected within the neighborhood $\mathcal{N}_k$ of the parent star s_k. Each offspring star s_o is chosen either as the closest in time to the parent star s_k, or as the one that realizes the maximum displacement in θ_f per unit time. Randomly alternating these two rules allows to combine large, but time consuming, horizontal movements, and "quick reproduction" phases corresponding, in most cases, to vertical movements in the (r, θ_f) plane. As a result, the following *throw and multiply* rule for offspring selection is defined:

$$s_o \quad = \begin{cases} \arg\min_{s_h \in \mathcal{N}_k} (t_h - t_k) & \text{if } \; p_r < 0.5 \\ \arg\max_{s_h \in \mathcal{N}_k} \frac{|\Delta\theta_f|}{t_h - t_k} & \text{otherwise} \end{cases} \tag{18}$$

being p_r a random number between 0 and 1 and $\Delta\theta_f$ the difference in final polar angle between star s_h and s_k. If the stars closest in time are more than one, one of them is randomly picked when $p_r < 0.5$. At this point, s_o is removed from $\mathcal{N}_k$ and another offspring star is selected with the same rule, until β_k offspring are generated. The β_k offspring are appended to one of the $m_k \left|\mathcal{L}^i\right|$ new successors under construction, and the points 3. and 4. are repeated for each of them.

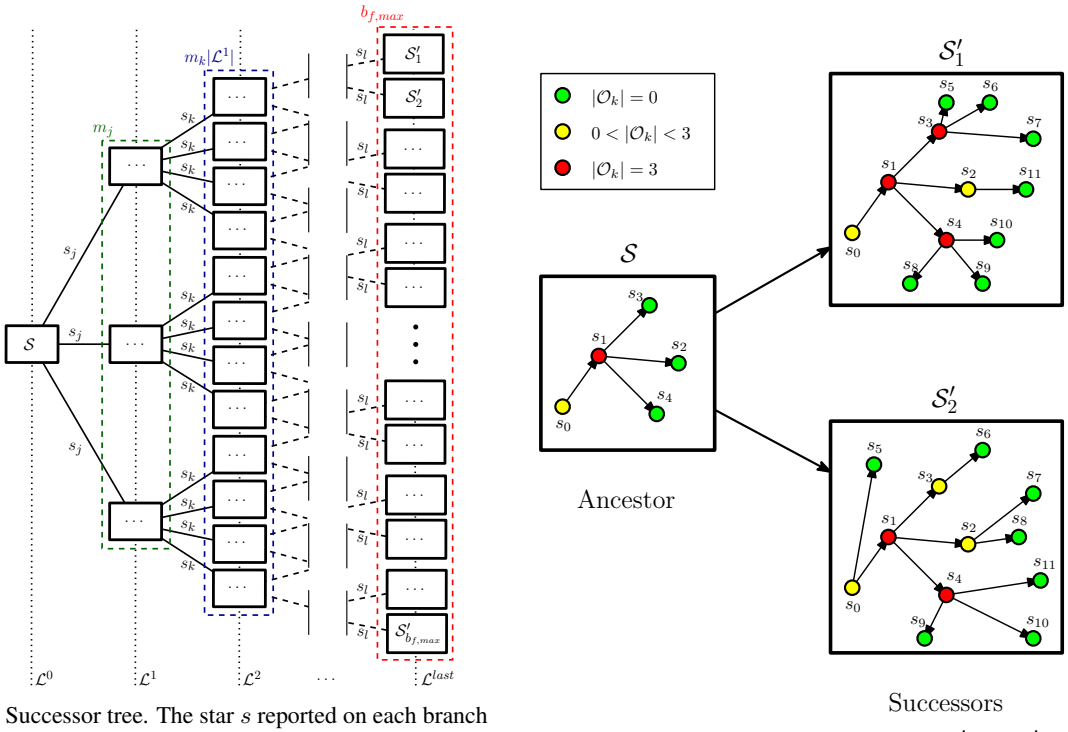
The procedure is repeated until: i) it is no longer possible to create new successors (that is, all neighbourhoods are empty or there is no available fertile star) or ii) a number of successors equal to $b_{f,max}$ has been generated. In both cases, all the successors in the last evaluated iteration, $\mathcal{L}^{last}$, are retained.

An example of the successor tree generated in a single transition from the ancestor state $\mathcal{S}$ is shown in Figure 2a; the settlement trees corresponding to the ancestor state $\mathcal{S}$ and two of its successors, $\mathcal{S}'_1$ and $\mathcal{S}'_2$, are shown in Figure 2b.

Beam Best-First Search

A multi-settler stochastic Beam Best-First Search (BBFS) has been developed for solving the unusual search problem defined in the previous section, as a generalization of the (single-spacecraft) "Stochastic Beam" algorithm proposed by Simões et al.[12] for handling the multi-rendezvous problem released in GTOC 5. Solution methodologies based on the beam search have been demonstrated to be successful also in other GTOCs, characterized by a wide search space.[13–16]

Being the beam search a *tree-search* algorithm, it tries to approach the goal gradually building a so-called *search tree*, with the initial state as root: the *nodes* of the tree correspond to the explored states in the state space of the problem, while the *branches* correspond to state-to-state transitions.



(a) Successor tree. The star s reported on each branch identifies the fertile star selected for offspring generation in the corresponding iteration.

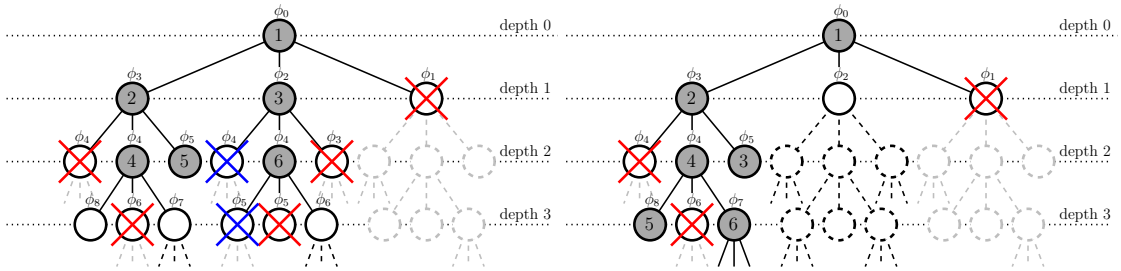
(b) Ancestor state $\mathcal{S}$ and two successors $\mathcal{S}'_1$ and $\mathcal{S}'_2$.

Figure 2: Transition model: successor tree (a), ancestor and successor states (b).

At each iteration, the algorithm picks the first *leaf* state in the *frontier*, that is, the list of all the generated, but yet unexplored, nodes at any given point of the search, and *expands* it by generating a set of *successor* states. In the proposed beam search variant, termed *best-first*, the frontier is modeled as a priority queue ordered by ϕ. So, at any iteration, the best state is used for generating $b_{f,max}$ successors states. A *probabilistic branching* is now exploited so as to avoid to fill the frontier only with the newly generated nodes: with a probability p, the best b_f successor states, according to ϕ, are retained and appended to the frontier; otherwise, the b_f successors are chosen by means of a biased sampling without replacement proportional to ϕ. Subsequently, the frontier, now containing a (possibly large) number of states evaluated at different iterations, is sorted according to the cost function ϕ and the best b_w are kept in memory, while the others are discarded. The now-exhausted ancestor state is removed from the search-tree frontier, and a new iteration may begin. The algorithm terminates as soon as successors can no longer be generated from any of the states in the frontier, or a maximum, pre-defined, number of nodes have been explored. The best-so-far state is eventually elected as problem solution.

The best-first (or uniform-cost[11]) variant has been here adopted, in place of the "standard" breadth-first search, in order to provide good-quality answers to each set-covering problem as soon as possible: indeed, as shown in Figure 3, the best-first variant is able to cross freely the different levels of the tree and quicker generate "deeper" nodes (*i.e.*, states with a larger number of settled stars), that we expect have a better solution score. In this way, settlement trees of up to 100 stars were generated in a few minutes on a standard laptop.

The hyper-parameters of the BBFS procedure were tuned "by hand" during the competition, by trying different values and comparing the correspondent results. At the end, the following values resulted to be the best ones for the great part of the analyzed zones: $b_w = 20000$, $b_{f,max} = 20000$, $b_f = 1000$. In order to mitigate potential issues due to the "non-exhaustiveness" and stochastic nature of the BBFS, several runs were performed in parallel, and the best-found solution saved.



(a) Beam (breadth-first) Search. (b) Beam Best-First Search.

Figure 3: Comparison between six iterations of a breadth-first and best-first version of the Beam Search on the same problem, by supposing $b_f = 2$ and $b_w = 3$. In particular, grey-filled nodes have been explored (with the number indicating the order); white-filled nodes have been generated but not yet explored; black-dashed nodes are yet to be generated; grey-dashed nodes are the successors of a (crossed) node pruned by the beam procedure. More precisely, red-crossed nodes are excluded by the probabilistic branching (b_f), while blue-crossed nodes by the frontier re-sizing (b_w). The values of the cost function ϕ are labeled in decreasing order: $\phi_j < \phi_i$ if $j > i$.

The proposed procedure has been repeated over all zones, iteratively changing the target occupancy distribution according to the attained results, in order to have an overall number of settled stars in each row and column as close as possible to the one stated by the global, 1020-star, target distribution devised beforehand. Eventually, a settlement tree with 1013 stars was found at the end of this process.

IDEAS FOR SOLUTION IMPROVEMENT

A number of refinement strategies were investigated during the competition in order to improve the BBFS solution, as soon as a 1020-star settlement tree were available. The three most effective are here presented.

Concurrent optimization of transfer times

Once the layout of the settlement tree has been defined, a large score boost can be obtained by a proper optimization of the intermediate departure and arrival times as well as by moving to 90 Myr the settle epoch of all terminal stars (i.e., stars without offspring).

A rudimentary sequential quadratic programming optimization has been carried out. Let $\delta = \delta(j)$ be the additional transfer time admitted for the j-th transfer leg composing a settlement tree of N legs (no special sort of the transfers is required). Modified departure and arrival times $t_f(j)$, $t_0(j)$ are evaluated as

$$t_f(j) = \bar{t}_f(j) + \delta(j) \tag{19}$$

$$t_0(j) = \bar{t}_0(j) + \sum_{i=1}^{N} M(i,j)\delta(i) \tag{20}$$

where over-lined variables refers to the unperturbed quantities, and $M(i,j) \in \{0,1\}$ is an $N \times N$ matrix indicating if the arrival star of the i-th transfer arc is the same as the departure star of the j-th arc ($M(i,j) = 1$) or not ($M(i,j) = 0$). A quadratic approximation of the total used ΔV is built using the derivatives of each transfer cost $\Delta V(j)$ with respect to its departure and arrival time:

$$\begin{aligned}\Delta V(j) = \overline{\Delta V}(j) &+ \frac{\partial \Delta V}{\partial t_0}\left(t_0(j) - \bar{t}_0(j)\right) + \frac{1}{2}\frac{\partial^2 \Delta V}{\partial t_0^2}\left(t_0(j) - \bar{t}_0(j)\right)^2 \\ &+ \frac{\partial \Delta V}{\partial t_f}\left(t_f(j) - \bar{t}_f(j)\right) + \frac{1}{2}\frac{\partial^2 \Delta V}{\partial t_f^2}\left(t_f(j) - \bar{t}_f(j)\right)^2\end{aligned} \tag{21}$$

A (local) minimization problem is then formulated as:

$$\min_{\delta_{min} \leq \delta \leq \delta_{max}} \sum_{j=1}^{N} \Delta V(j) \tag{22}$$

$$\text{s.t.} \quad \Delta V(j) < \Delta V_{tot}^{max} \quad \forall\, j \in \{1, \ldots, N\} \tag{23}$$

where δ_{min}, δ_{max} are sufficiently close bounds that ensure the validity of the quadratic approximation; $\delta_{min} = -0.2$ Myr and $\delta_{max} = 1$ Myr have been selected according to the author experience. This procedure is repeated, each time updating the reference value with the newly obtained ones. A few iterations are usually sufficient to acquire a significant reduction of the overall mission ΔV.

Explosion

The overall settlement tree, obtained as the sum of the solutions of the different set covering problems, may still grow in size (*i.e.*, in number of stars) without hindering the distribution score J_2. In fact, with respect to the previously discussed set covering problem, this settlement tree already covers all the cells of interest. Therefore, it is now possible to work on a simpler cell-by-cell basis, while keeping the distribution as uniform as desired.

The explosion algorithm is here presented. Let $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ be the set of settled stars so far. Each star s is characterized by a certain number of available Settler Ships $n_A(s) = 3 - |\mathcal{O}(s)|$ that can still depart and reach new stars. Let $x(k_r, k_\theta)$ be an integer variable indicating the number of additional stars to settle in cell (k_r, k_θ). This variable is non-negative and upper bounded by the maximum number of stars $x_{\text{max}}(k_r, k_\theta)$ that can be settled in the cell. Working on a cell-by-cell basis, that is, only transfers within a cell are allowed, a rough estimate of this upper bound is given by the sum of the available settlers in that cell:

$$x_{\text{max}}(k_r, k_\theta) = \sum_{s \in (k_r, k_\theta)} n_A(s) \tag{24}$$

If newly settled stars are reached early in time a new set of Settler Ships is available for further colonization of the galaxy, thus generating an entire new "army". Therefore, a more accurate estimate of the maximum number of new stars per cell, given the problem time/impulse magnitude constraints, is given by:

$$x_{\text{max}}(k_r, k_\theta) = \sum_{s \in (k_r, k_\theta)} \left(n_A(s) \sum_{k=0}^{n_{\text{gen}}} 3^k \right) \tag{25}$$

where n_{gen} is the number of following generations. It can be computed as:

$$n_{\text{gen}} = \left\lfloor \frac{t_{end} - t(s)}{\Delta t + t_w} \right\rfloor \tag{26}$$

where $t(s)$ denotes the settling time of the departure star s, t_w is the waiting time before a Settler Ship can depart from a settled star, t_{end} denotes the mission final time and Δt is the average transfer time (assumed equal to 6 Myr).

The (putative) optimal number of additional stars $x(k_r, k_\theta)$ to settle in each cell is found as the solution of an Integer Linear Programming (ILP) problem that aims at minimizing the row/column error on the novel star distribution with respect to the desired target one, given i) the current star distribution $N_0(k_r, k_\theta)$, ii) a target row/column star distribution $N_r^{des}(k_r), N_\theta^{des}(k_\theta)$, and iii) a rough estimate of the maximum number of stars the could be settled in a given cell (k_r, k_θ) by working on a cell-by-cell basis. The problem can be stated as:

$$\min_{0 < X < X_{max}} \sum_{k_r=1}^{30} \Delta_r(k_r) + \sum_{k_\theta=1}^{32} \Delta_\theta(k_\theta) \tag{27}$$

where $\Delta_r(k_r)$, $\Delta_\theta(k_\theta)$ are auxiliary error functions defined as the difference between the settled stars (both before and after the explosion) and the desired ones in a ring or, respectively, in a slice:

$$\Delta_r(k_r) = \left| \sum_{k_\theta=1}^{32} \left(x\left(k_r, k_\theta\right) + N_0\left(k_r, k_\theta\right) \right) - N_r^{\text{des}}(k_r) \right| \tag{28}$$

$$\Delta_\theta(k_\theta) = \left| \sum_{k_r=1}^{30} \left(x(k_r, k_\theta) + N_0(k_r, k_\theta) \right) - N_\theta^{\text{des}}(k_\theta) \right| \tag{29}$$

By repeatedly solving for a target distribution that is a multiple of the theoretical optimal 512-star target distribution, one hopes to find a refined target distribution that features a much larger number of stars without breaking the initial regularity. Once the putative optimal number of new stars to settle in each cell has been obtained, a greedy search is carried out on a cell-by-cell basis in order to find the actual settlement tree that realizes that distribution. All possible combinations of minimum-time transfers between already settled stars and available stars in that cell are considered. Fertile stars are sorted by their settle times; those with lower settle time are used first in order to generate as soon as possible new vessels that could possibly reach other stars before the end of the mission time.

Pruning

In most cases, the attained settlement tree is not as uniform as expected. This may happen for a variety of reasons. As an example, the BBFS may have favoured a group of stars very close each other, or the explosive algorithm my have failed in uniformly expanding the original settlement tree. In either case, a non-uniform star distribution could severely hinder the overall score. Therefore, in order to re-gain a more uniform star allocation, the settlement tree undergoes a (greedy) removal process, *i.e.*, a "pruning": for each star, the increase in the merit index J_2 associated with its elimination is computed; the star leading to the largest score improvement is actually removed. This procedure is iteratively repeated as long as the score function keeps improving. This step should always be followed by a concurrent optimization of all transfers time, as reducing the number of stars may also reduce the overall propellant consumption.

NUMERICAL RESULTS

Some numerical results that document the adopted procedure are here presented. At the beginning of the competition a large effort has been devoted to the analysis of the mobility of Fast Ships and Settler Ships. Figure 4 highlights the stars that is possible to reach with a Fast Ship as a function of the allowed travel time. Mobility charts as those shown in Figure 6 were instead used for understanding the capability of a Settler Ship to reach nearby stars, with respect to the allowed travel time. From this analysis, it was apparent the scarce lateral mobility at higher radii, somehow opposed by the great lateral mobility at lower radii, and the usually good radial mobility at any radii. This analysis was also fundamental for the definition of the *throw and multiply* strategy exploited in the BBFS procedure.

After a long reasoning, nine (almost) non-overlapping zones in the plane (r, θ_f) were figured out. Each zone refers to a "root" star either provided by a Fast Ship or by a Mother Ship: one of the Mother Ships intercepts three stars and the others two stars each; their trajectories are reported in Figure 5. Independent settlement trees departed from each root star. These zones are listed in Table 2, and shown in Figure 7b in the (r, θ_f) plane. The following nomenclature has been used to label each zone: the initial letter indicates if the star is encountered by a Mother Ship (M) or a Fast Ship (F), respectively; the first number uniquely identifies the ship, and, the second number (only for Mother Ships) corresponds to the settled star number (ordered by settling time).

Figure 7a provides a visual representation of the initial (tentative) 1020-star distribution on the (r, θ_f) grid. As more information about the problem were gathered, the target occupancy distribution was updated, up to the very last day, bringing to the final distribution shown in Figure 7b.

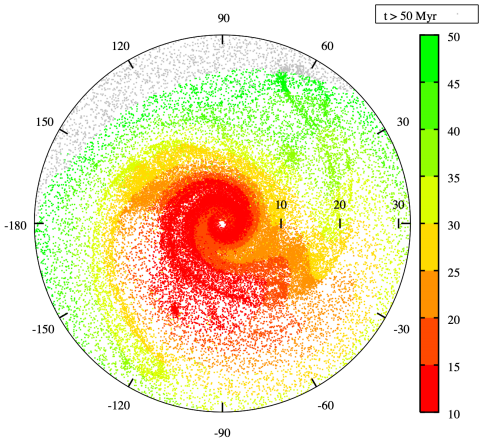


Figure 4: Reachable stars in the galaxy as a function of the transfer time for the Fast Ships.

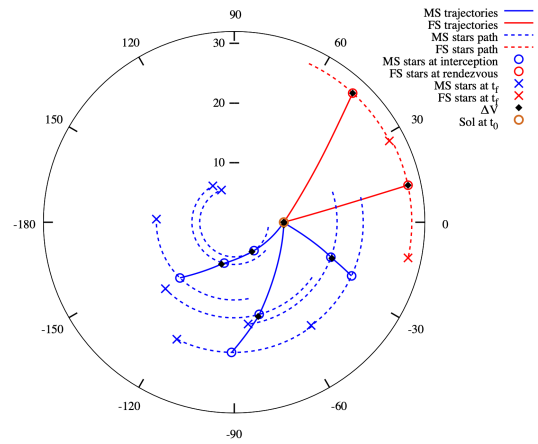
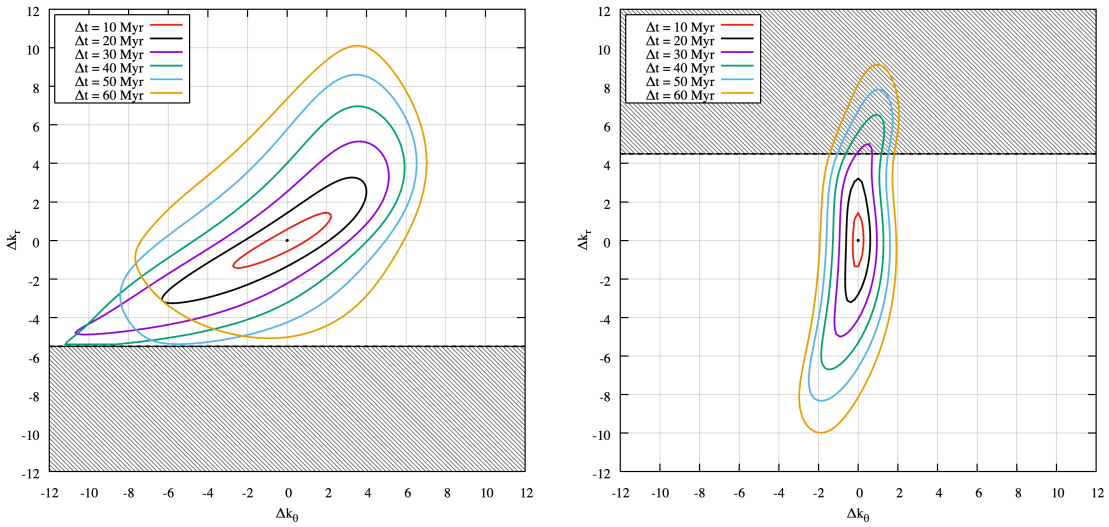


Figure 5: Trajectories of Mother Ships and Fast Ships: view from Galactic North.



(a) $r_0 = 7.5\,kpc$.

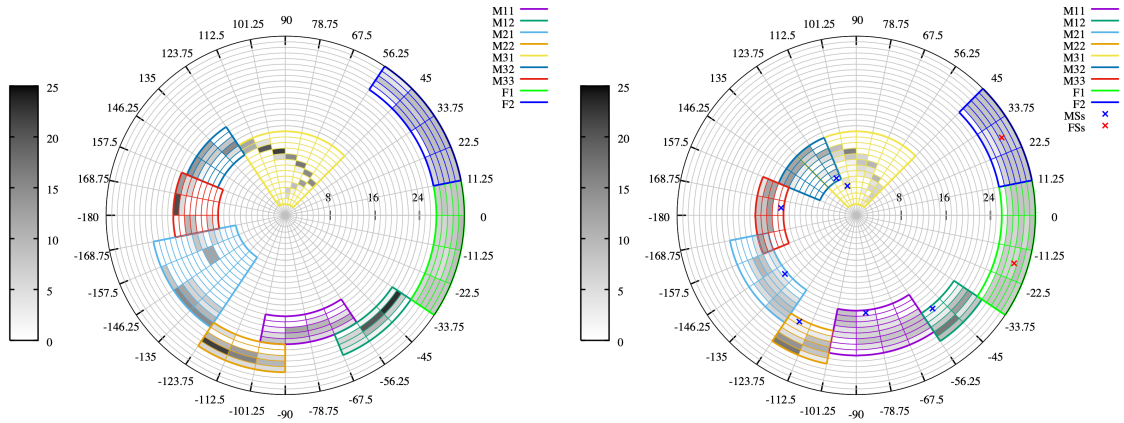
(b) $r_0 = 27.5\,kpc$.

Figure 6: Mobility maps. Contour lines are represented for a vessel departing from two different radii at $t_0 = 30$ Myr, as a function of the transfer time Δt. Patterned zones indicate regions beyond the given boundaries of the galaxy.

.

Z	$M11$	$M12$	$M21$	$M22$	$M31$	$M32$	$M33$	$F1$	$F2$
k_r	$8 \div 11$	$12 \div 13$	$2 \div 5$	$6 \div 7$	$21 \div 28$	$27 \div 30$	$31 \div 2$	$14 \div 17$	$18 \div 20$
k_θ	$16 \div 23$	$20 \div 25$	$15 \div 21$	$20 \div 25$	$1 \div 13$	$6 \div 13$	$12 \div 16$	$25 \div 30$	$25 \div 30$

Table 2: Zone names and ranges.

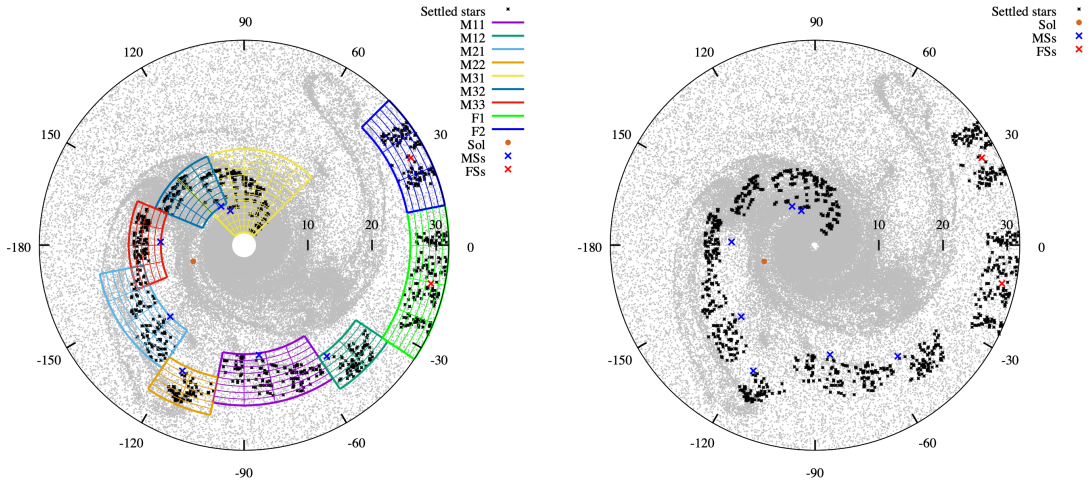


(a) Target distribution in the (r, θ_f) plane.

(b) Obtained distribution in the (r, θ_f) plane.

Figure 7: Identified zones and cells occupation: target distribution (a), obtained distribution (b). The grey-scale refers to the number of stars to settle in any cell. Blue and red crosses identify the cells where the "root" stars encountered by Mother Ships and Fast Ships, respectively, are located in the final solution.

At the end, the final submitted solution, shown in Figure 8, is characterized by 1013 stars, "uniformity" merit index $J_2 = 630.069$ and "propulsive" merit index $J_3 = 1.504$, resulting in an overall score $J = 946.451$ and in the 7th place in the leaderboard.



(a) Submitted solution: view in the (r, θ_f) plane.

(b) Submitted solution: view from Galactic North.

Figure 8: Final submitted settlement tree. The nine search zones are highlighted.

Enhanced solution

Due to the competition limited time, the submitted solution did not take advantage of the presented explosion and pruning techniques. If these two ideas are applied to the same settlement tree, by first increasing as most as possible the number of settled stars (explosion) and then removing unnecessary settlements until J_2 reaches a local optimum (pruning), one attains a significantly better solution. Figure 9 presents the starting 1013-star settlement tree, the "exploded" solution with 1377 stars, and the "pruned", final, solution with 1220 stars, with merit index $J_2 = 772.761$ and $J_3 = 1.5531$, that is an overall score $J = 1200.145$. This result

suggests that the proposed procedure is able to attain results competitive with the other top teams.

Solution	N	E_r	E_θ	J_2	ΔV_{used} [10^3 km/s]	ΔV_{max} [10^3 km/s]	J_3	$J = J_2 * J_3$
Submitted	1013	1.88968	4.11965	630.069	271.42	408.20	1.504	946.451
Exploded*	1375	3.67621	3.50581	691.814	405.65	550.50	1.3571	938.839
Enhanced	1220	1.47322	3.27067	772.761	314.54	488.50	1.5531	1200.145

* ΔV not optimized.

Table 3: Summary of the best found settlement trees.

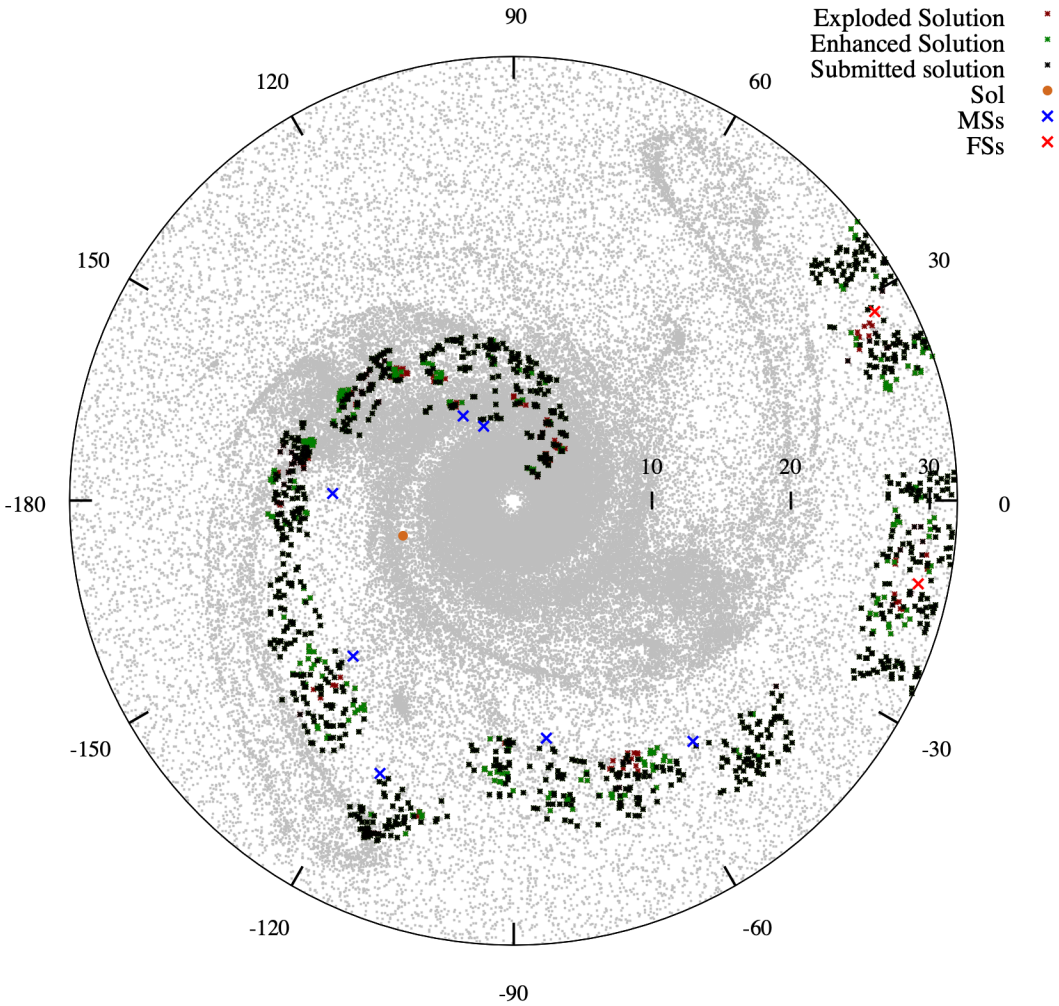


Figure 9: Comparison between the obtained solutions.

CONCLUSIONS

Team Sapienza-PoliTo proposed a solution strategy mainly devoted to maximize the "uniformity" of the final settlement tree. In order to reduce the search space dimension, the efforts have been focused on the achievement of an *a priori* chosen star distribution, constructed after a careful analysis of the problem merit function and constraints. The search procedure has been further simplified by partitioning the overall distribution into smaller, independent, zones, each to be explored departing from one of the root stars encountered by the initial ships. An *ad-hoc* formulated Beam Search procedure has been exploited to built suitable settlement

trees in each of the identified zones, leading to an overall, 1013-star, solution, with a distribution quite close to the desired one. As final step, a concurrent optimization of transfers (by means of an indirect procedure) and transfer times was performed, leading to the final, 7th ranked, submitted solution. A number of refinement procedures have been developed in the very last days of the competition, but completed immediately after. Thanks to such procedures, a refined 1220-star solution was then obtained, that would (probably) have ranked fifth if it had been submitted until the competition term.

ACKNOWLEDGEMENTS

The authors are willing to acknowledge the restless work and the commitment of the team members here listed in alphabetical order: Andrea Ferrero, Dario Pastrone, Filippo Masseni and Luigi Mascolo (PoliTo), Francesco Simeoni (independent) and Danilo Zona (Sapienza).

REFERENCES

[1] A. E. Petropoulos, E. D. Gustafson, G. J. Whiffen, and B. D. Anderson, "GTOC X: Settlers of the Galaxy Problem Description and Summary of the Results," *Paper AAS 19-891, Astrodynamics Specialist Conference*, Portland, Maine, 11-15 Aug. 2019.

[2] L. Casalino, D. Pastrone, F. Simeoni, G. Colasurdo, and A. Zavoli, "GTOC5: Results from the Politecnico di Torino and Universita di Roma Sapienza," *Acta Futura*, No. 08, 2014, p. 2936, 10.2420/AF08.2014.29.

[3] G. Colasurdo, A. Zavoli, A. Longo, L. Casalino, and F. Simeoni, "Tour of Jupiter Galilean moons: Winning solution of GTOC6," *Acta Astronautica*, Vol. 102, 2014, pp. 190–199. cited By 4, 10.1016/j.actaastro.2014.06.003.

[4] L. Casalino, G. Colasurdo, A. Zavoli, and M. Berga, "GTOC8: Results and methods of Team 22," *Advances in the Astronautical Sciences*, Vol. 158, 2016, pp. 4291–4300.

[5] A. E. Bryson, Y.-C. Ho, and G. M. Siouris, *Applied Optimal Control: Optimization, Estimation, and Control*, Vol. 9. Institute of Electrical and Electronics Engineers (IEEE), 1979, 10.1109/tsmc.1979.4310229.

[6] A. Zavoli, F. Simeoni, L. Casalino, and G. Colasurdo, "Optimal cooperative deployment of a two-satellite formation into a highly elliptic orbit," *Advances in the Astronautical Sciences*, Vol. 142, 2012, pp. 3647–3663.

[7] L. Casalino, G. Colasurdo, and D. Pastrone, "Optimal Low-Thrust Escape Trajectories Using Gravity Assist," *Journal of Guidance, Control, and Dynamics*, Vol. 22, sep 1999, pp. 637–642, 10.2514/2.4451.

[8] L. Casalino, G. Colasurdo, and D. Pastrone, "Optimization Procedure for Preliminary Design of Opposition-Class Mars Missions," *Journal of Guidance, Control, and Dynamics*, Vol. 21, jan 1998, pp. 134–140, 10.2514/2.4209.

[9] W. Clohessy, "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, Vol. 27, No. 9, 1960, pp. 653–658.

[10] J. E. Prussing, *Optimal multiple-impulse orbital rendezvous.* PhD thesis, Massachusetts Institute of Technology, 1967.

[11] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.

[12] L. F. Simões, D. Izzo, E. Haasdijk, and A. Eiben, "Multi-rendezvous spacecraft trajectory optimization with beam P-ACO," *European Conference on Evolutionary Computation in Combinatorial Optimization*, Springer, 2017, pp. 141–156.

[13] I. Grigoriev and M. Zapletin, "Choosing promising sequences of asteroids.," *Automation & Remote Control*, Vol. 74, No. 8, 2013.

[14] D. Izzo, L. F. Simões, M. Märtens, G. C. d. Croon, A. Heritier, and C. H. Yam, "Search for a grand tour of the jupiter galilean moons," *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, ACM, 2013, pp. 1301–1308.

[15] D. Izzo, D. Hennes, L. F. Simões, and M. Märtens, "Designing complex interplanetary trajectories for the global trajectory optimization competitions," *Space Engineering*, pp. 151–176, Springer, 2016.

[16] A. Petropoulos, D. Grebow, D. Jones, G. Lantoine, A. Nicholas, J. Roa, J. Senent, J. Stuart, N. Arora, T. Pavlak, *et al.*, "GTOC9: Methods and results from the Jet Propulsion Laboratory team," 2017.