Abstract. Humans have an innate ability to extract important information from data sets or images, but also to solve particular problems on which computers may struggle. Serious gaming, and the related crowdsourcing term, are new computational paradigms attempting to process enormous amounts of scientific data exploiting these innate humans capabilities. In this paper we present Space Hopper, a “serious gaming” experiment aimed at improving interplanetary spacecraft trajectories design techniques and, at the same time, at proving that part of an interplanetary trajectory design can be crowdsourced. Space Hopper exploits and collects data on the users’ problem-solving skills and spatio-temporal reasoning to help formulate a “human-inspired” tree search algorithm allowing efficient traversal of vast trees.

Keywords: crowdsourcing, serious gaming, trajectory design, tree search algorithms, self-adaptive differential evolution

1 Introduction

The design of spacecrafts interplanetary trajectories has been the exclusive domain of mission analysis experts. However, recent advances in the field of trajectory optimization, coupled to the growth of computational power, makes it possible to automatize a good deal of the necessary design steps and thus decrease the knowledge necessary produce good designs. Sophisticated modern search strategies (e.g. based on evolutionary computations) can explore vast solution spaces and locate good interplanetary trajectories able to compete with human-competitive results [10] taking the domain expert almost completely out of the loop.

With the serious game Space Hopper, we want to experiment with a radically new approach to interplanetary trajectory design which neither solely relies on computational power nor on human expert knowledge. Instead, we make use of the human spatio-temporal reasoning, intuition, curiosity and general problem-solving skills, as recorded during a game session, to improve the search of available trajectory options. By providing an appealing and easy user interface, humans with little or no knowledge about orbital mechanics, mission analysis or space engineering are able to design real trajectories enabling specific scientific goals to be pursued. In this first game prototype, the exploration of the Jovian system is considered.

From this perspective, Space Hopper can be seen as a serious game in which players from all over the world compete to find the best-possible flyable interplanetary trajectories. Unlike several previous scientific crowd-
sourcing games looking for explicit solutions to their specific problem, Space Hopper also extracts information about the individual actions of all players on their way to discovering their final solutions. By applying machine learning techniques to this sequential data we aim to model the human decision making progress for this particular task. Out of this model, we want to infer a human-inspired general search strategy that can be transferred to other complex interplanetary trajectory problems in order to improve already known fully-automated search algorithms.

The structure of this paper is as follows: in Section 2 we give a short survey about related work with respect to serious gaming and crowdsourcing applied to science. Section 3 deals with the domain of interplanetary trajectory optimization and introduces the problem to be solved by the non-expert human players. Section 4 will give details about the user interface, the gaming experience and the underlying technologies used to develop Space Hopper. In Section 5 we describe the underlying data model and outline how we intend to use the collected information. We conclude with discussing future work and how crowdsourcing experiments can help advance the field of space exploration in Section 6.

2 Related Work

Using games for other purposes than mere entertainment is widely known under the broad term serious gaming [24, 15], first introduced by Clark Abt [1] in 1970. While these games may still be entertaining, they traditionally attempt to educate, inform or train the player. The objective of a serious game is usually linked to a specific sequence of amino acids. While this problem is computationally expensive, the game makes use of the inherent reasoning, learning and pattern recognition skills of humans, allowing for a way faster computation. A major achievement of the players of Foldit was helping to fully model the Mason-Pfizer monkey virus enzyme, a virus which causes AIDS among monkeys. Bioengineers have been trying to decipher the structure of this virus for 15 years whereas the online player community required just 10 days to fully model it in 3D [11].

There are several other scientific discovery games to mention. In the citizen science project Galaxy Zoo Supernovae [21] players were shown the wield-field images to search for stars in order to classify super nova candidates. More projects involve humans to classify bat calls, galaxy types, cancer cells, discovering exoplanets, cyclones and many more. In the online game Eye-Wire ³, the players help to trace the spatial structure of neural connections throughout the retina of mice.

Most of the time, scientific discovery games exploit

1See Kickstarter (http://www.kickstarter.com) and Indiegogo (http://www.indiegogo.com) for two popular platforms.

2See https://www.zooniverse.org

3See https://www.eyewire.org
the advanced image processing capabilities of humans, which are still by far superior than nowadays algorithms. Peekaboom [27] uses these abilities to let its players detect specific objects and relevant regions in pictures. By providing human annotations, there is hope that games with a purpose are able to create ontologies to help weaving the semantic web [20].

3 Background

Space Hopper is based on the problem proposed by NASA’s Jet Propulsion Laboratory (JPL) as part of the sixth edition of the Global Trajectory Optimisation Competition (GTOC6). The participating team were tasked to globally map Jupiter’s Galilean moons [17] (Io, Europa, Ganymede and Callisto) using the multiple gravity assist technique within a four-year window. The surface of each moon is divided into 32 areas (faces) where each face is either a pentagon or a hexagon. This division approximates the moons as truncated icosahedrons (classic soccer balls). Each face is assigned a score from 1 to 3 depending on the scientific interest of the area on a given moon (see Figure 2). Faces on Europa are worth double points due to higher scientific interest of this particular moon. Points are scored by successively mapping faces of each of the four moons. A face $F_i$ is visited if the closest approach vector $r_p$ of a flyby around a chosen moon is passing through that particular face. For a face to be visited, it has to be fully or partially within the "visitable band" an area on the surface of the given moon where the flyby parameters are within the allowable bounds (see Figure 3). A face $F_i$ is visitable if at least one of its vertices lies within the band or its vertices lie on both sides of the band (yet none inside it). Points for visiting face $F_i$ are scored only for the first flyby over that face, no points are added to the overall trajectory score for subsequent flybys over face $F_i$. The maximum cumulative score for a full tour of the Galilean moons is 324.

At starting epoch $t_0$ the spacecraft mass is $M_0 = 2000\text{kg}$, half of which is the propellant, therefore the mass of the spacecraft cannot fall below 1000kg. With each flyby the overall spacecraft mass is reduced based on the thrust used during the Deep Space Manoeuvre (DSM) with a maximum continuous thrust of $\tau_{\text{max}} = 0.1\text{N}$. Furthermore, there also exists a mass penalty for close approaches to Jupiter to account for additional shielding from Jupiter’s strong magnetic field. The penalty is applied if, at any time during the mission, the spacecraft range to Jupiter goes below $2R_J$.

GTOC6 was won by a joint team of University of Rome and Turin Polytechnic with best score of 311 points. The team attempted to fully map one moon before moving onto the next one. However, after the competition a new solution scoring 316 points was found by Advanced Concepts Team (ACT) of European Space Agency. The trajectory is fundamentally different from the winning solution of GTOC6 as it exploits a “moon-hopping” technique allowing rapid transfers between moons rather than fully mapping one moon before moving to the next one. The found trajectory was evolved using PaGMO/PyGMO⁴ and is up to date the best known valid solution for the GTOC6 problem [10].

⁴See http://pagmo.sourceforge.net/pygmo

DOI: 10.2420/AF09.2014.91
The problem was defined with great detail and accuracy thus it required an appropriate representation of the Jupiter system and spacecraft dynamics. Space Hopper uses state-of-the-art methods based on PyKEP, an open-source keplarian calculations toolbox developed by the ACT. The error margin for the calculations is at maximum $10^{-9}$. The parameters of the Galilean moons as well as Jupiter were defined in JPL's problem statement (e.g. Keplerian orbit elements, gravitational parameter, radius, etc.). The time epochs are encoded in Modified Julian Date format (number of days elapsed since 17 November 1858). The moon ephemerides (cartesian position and velocity) are calculated according to values provided by JPL without any perturbations in the orbits of the satellites. The overall accuracy and realistic approach to the problem makes it feasible for a pre-phase A study.

The Grand Tour starts at a point on a sphere of radius $R_J=$10000R$_J$ with Jupiter at the centre. The first leg (called *incipit*) is unique as it does not include a DSM in the transfer to the first visited moon. The parameters of the first leg are encoded into a 4-dimensional array (called a chromosome in evolutionary computing) $x_0 = [t_0, u, v, T_0]$. $t_0$ is the epoch of the spacecraft launch date. $u$ and $v$ are used to calculate the coordinates of spacecraft at epoch $t_0$ on the aforementioned sphere around Jupiter:

$$ r_0 = R_J(\cos \theta \cos \phi \hat{i} + \sin \theta \cos \phi \hat{j} + \sin \phi \hat{k}) $$

where $\theta = 2\pi u, \phi = \cos^{-1} (2v - 1) - \pi/2$, $u$ and $v$ were chosen over directly optimising the values of $\theta$ and $\phi$ to ensure a uniform distribution of sampled points on the sphere. $T_0$ is the duration of the spacecraft’s journey to the first visited moon (in days) with its lower and upper bounds set at 180 and 220 days respectively.

Each subsequent leg of the tour is an interplanetary trajectory between two moons of Jupiter with a single Deep Space Manoeuvre. These legs are also encoded as a 4-dimensional chromosome $x_n = [\beta_n, r_{pn}, \eta_n, T_n]$ where $\beta_n$ is the flyby angle at the periapsis, $r_{pn}$ is the periapsis of the flyby, $\eta_n$ is the timing of the DSM (ratio of days elapsed before the DSM to total leg duration $T_n$). The bounds for $\beta$ and $r_p$ are calculated for each face of the currently visited moon based on the incoming velocity of the spacecraft but have to be inside $[-2\pi, 2\pi]$ and $[50000/R_m, (R_m + 2000000/R_m)]$ respectively where $R_m$ is the radius of the currently visited moon. $\eta$ values have to be between 0 and 1 exclusive.

For each leg its chromosome is optimised using a self-adaptive Differential Evolution algorithm (jDE) [4] to ensure the best results. jDE allows us to skip parameter tuning before solving each leg problem and still guarantees high performance of the optimisation. The flyby parameters for each leg of the trajectory are evolved 200 times over 60 generations with 12 individuals. At each step of the optimisation, an individual is evaluated using an objective function which calculates each legs change in velocity $\Delta V$ during the Deep Space Manoeuvre. The evolution attempts to minimise $\Delta V$ for each leg in order to make the trajectories as close to ballistic as possible.

The objective function of the first leg is computed as a Lambert Problem between $r_0$ and $r_1$ with transfer time $T_0$ where $r_0$ is the position of the spacecraft on the 10000R$_J$ sphere (calculated using the $u$ and $v$ variables) at time $t_0$. $r_1$ is the position of the first visited moon $r_m$ at time $t_0 + T_0$. Each consecutive leg has an objective function defined as $\text{MGA-1DSM}$ [9].

### 4 Game Design

This section describes the current design status and some future planned developments for Space Hopper. We present the details of the different technologies used to create the experiment. Furthermore, the gameplay and interface is explained in order to better understand the application. Lastly, we present the search tree that is used to represent the underlying problem and discuss its content and significance.

![Figure 3: The visitable band shown on an example planet](image)
4.1 Development and design

Space Hopper was designed with adaptability and compatibility in mind. The chosen technologies and methods comply with the established and emerging standards of interactive web development.

HTML5 was chosen as to maximise the number of possible players/contributors as it is the current standard for dynamic websites and online games. Furthermore, it is currently being introduced on mobile devices allowing further expansion of the user base and development in a new direction through the use of touchscreens. HTML5 allows for a more engaging design through combined use with various other programming languages and technologies. An example of this is using inline CSS for secondary style definition of the webpage.

JavaScript is the most crucial element of interactive web pages and applications as it supports (among others) the object-oriented programming style. Such capabilities allow definition and manipulation of composite objects in Space Hopper as well as carrying out various complex astrodynamics calculations thousands of times a second. Furthermore, the differential evolution algorithm is also defined in JavaScript which is currently emerging as a programming language for scientific purposes and has proved to have high performance [2]. However, Space Hopper seamlessly incorporated the algorithm without sacrificing the fluidity or the experience of the experiment. This is mostly due to the implementation of equations being based on the aforementioned open-source keplerian toolbox, PyKEP.

Space Hopper graphics are also supported by JavaScript through the use of Three.js [5], a compact, yet powerful open-source 3D graphics library. It allows creating various visual objects such as trajectory and orbit curves, spheres and even truncated icosahedrons (moon models). Moreover, Three.js manages the crucial scene objects of the HTML canvas including camera settings, movement in three-dimensional space and object texturing.

MySQL is used to store data submitted by the players who choose to do so. The database contains the details of the players choices and the values of crucial variables at each step of trajectory design including backtracks. Each row in the database is a full interplanetary trajectory. PHP is used to transfer the gathered data to and from the MySQL database.

4.2 User interface and Game-flow

Space Hopper can start in two different ways, with or without the incipit phase. This choice affects the difficulty of the later gameplay. Although the two starting points of the game vary only by two legs of the trajectory, designing the rest of the tour is very different. Designing the full tour (including the incipit phase) is a much greater challenge because the first trajectory legs are quite eccentric and have a very large semi-major axis. The players have to adjust various variables such as leg duration and the perijove of the leg trajectories in order lower semi-major axis and achieve short-duration transfers between moons with $\Delta V \approx 0$. At the very beginning the player has to choose two moons to fly by. Afterwards the incipit trajectory optimisation takes place. On the other hand, the second type of gameplay (post-incipit phase) sets up the initial conditions so that it is very easy for the players to achieve $\Delta V \approx 0$ trajectories without having to spend extensive time adjusting the flyby variables.

Following the initial phase of Space Hopper, the game continues with the spacecraft at a particular moon (either the second of the moons chosen during the incipit phase or Ganymede for the non-incipit level). The player has to select one of the moons on the faceted surface of the currently visited moon. Depending on the velocity of the spacecraft and the moon ephemerides, for each flyby, on average 12 out of 32 of these faces are visible. The player has to choose very carefully as each decision has a major impact on the following legs of the trajectory, even though the immediate gain from the face value might not be the highest.

After selecting the face to fly by, the player is presented again with the full view of the Jovian system.
so that he can select the next moon to visit. Again, the automated leg optimisation follows. This optimisation depends on the previously selected face and the general user-adjustable parameters of the trajectory leg. The user can adjust the duration of the following leg (in days), the perijove of the flyby and the bounds for the face to ensure scoring of the chosen face (there is approximately 90% chance of scoring the selected face due to the non-rectangular shape of the faces).

The process of face and moon selecting is then repeated until either all of the faces are visited, the mission time has expired or the spacecraft mass has fallen below 1000kg.

When the game has finished, a prompt appears showing the details of the designed trajectory (total score, cumulative ∆V, remaining mass, etc.). The user can then submit the score to be saved in the ACT database and analysed later.

4.3 Search tree

The entire problem has been formulated as a tree-search. While usually tree traversal algorithms are very efficient, the search space grows exponentially in this case and thus becomes very quickly too vast to be fully explored in order to find an optimal solution. In Space Hopper, the average number of the visitable faces, at any point in the search, is 15 for each of the 4 moons. Therefore, the search tree has an average branching factor of 60 which means that at the nth step the search tree is of size 60^n. Although there exist many efficient tree-search algorithms [31, 25, 16, 8], this problem is still too difficult for any of them to return a good solution in a reasonable time period due to sheer number of possible choices and necessary calculations at each step of the search to determine the overall fitness of the trajectory.

In the tree, each of the nodes is a full interplanetary trajectory between two moons which contains crucial information about the position of the spacecraft, m (i.e. which moon of Jupiter it is visiting), at epoch t, its velocity (v_m), and the accumulative velocity change (∆V) as well as a list of all the previously visited moon faces (f).

Humans are an integral part of Space Hopper’s tree search. At each step, users provide crucial information by making informed decisions about which face and moon to visit and by manipulating the flyby parameters.

Space Hopper’s approach to tree-search differs even more from the classical methods as each step can be re-optimised multiple times to achieve lower ∆V as the evolutionary algorithm responsible for finding the best flyby parameters can often return a non-optimal solution. Re-optimisation can also mean the players adjusting the flyby bounds to improve the optimisation of the trajectory through jDE.

5 Data Mining

In order to analyse the human players’ reasoning, data needs to be collected. All of the decisions made by users are recorded to ensure maximum feasibility when formulating heuristics for the novel tree-search algorithm. The submitted solutions are the full Jupiter Grand Tour trajectories which, as mentioned before, are formulated as a search tree where each node contains details about the state of the spacecraft at that time (currently visited moon, velocity) and global values of the tour (epoch, previously visited faces and the cumulative change in velocity of the spacecraft).

We attempt to formulate a set of heuristics for a new "human-inspired" algorithm based on the players’ understanding of the system and their decisions. We therefore need to track the players’ gameplay and understand how their choices affect general outcomes in terms of score, ∆V, time, spacecraft mass, etc. As a result, backtracks in the search are also recorded. Thus the search tree also contains branches (parts of the tour) which were considered and explored but ultimately discarded.

5.1 Future Work

The second part of the project is the most important one as it will require very sophisticated machine learning algorithms for recognising patterns in the collected data submitted by the best online players. The resulting data should provide us with very interesting mix of Breadth-first (BFS) and Depth-first (DFS) search algorithms which comes naturally to humans and can prove beneficial to the advancement of efficient and powerful tree-search algorithms for trajectory design. Also, learning from re-optimisations of legs and user backtracks in the search should give us better techniques for assessing the feasibility of fitness scores for different types of trajectories.

The resulting "human-inspired" tree-search algorithm will then be compared and contrasted with various well-established and efficient algorithms (such as Breadth-first, Depth-first, Best-first, A* or Lazy-race).
This should give us an indication of how the human abilities fare against the best of decades of Artificial Intelligence research.

6 Conclusion

This paper introduced Space Hopper, a novel online, crowdsourcing experiment which attempts to improve automated trajectory design methods by analysing how humans try to tackle the complex Jupiter Grand Tour problem. The problem is currently too computationally expensive even for the state-of-the-art algorithms but humans have a set of certain problem-solving skills which could prove helpful in advancing search algorithms for vast search trees with a high branching factor. Space Hopper is released online and features a 3D game-like look & feel to maximise the user base. Current trajectory design methods have reached a level where non-experts can design complex yet feasible trajectories.

References


