To simplify the task we release/collect probes only at asteroids. After experimenting we chose a flight path (Earth, a1, a2, a3) to reach three different asteroids with the mothership. Four asteroids would either not be reachable with the given rocket fuel, or reduce the time windows for probe release/return too much. Two asteroids would cost a primary performance point, since all probes would have to start at the same asteroid. To reduce the search space we chose fixed a3 = Iainbanks, because of its high connectivity (see definition below). a1 and a2 were chosen randomly with bias to asteroids highly connected, and a repeated evolutionary optimization process using CMA-ES was applied, aiming at low fuel consumption and large time windows for probe release/pick up. This resulted in a1 = Pire and a2 = Ovaitt, because this combination provides large probe time windows and high connectivity of the asteroids. We start two probes at a1 = Pire, one at a2 = Ovaitt and collect all three at a3 = Iainbanks. To carry two probes to Ovaitt would have violated the fuel consumption constraint for the mothership.

Probes
To evaluate the effectiveness of a given search method, it helps to encapsulate the domain knowledge from the search using a small well defined interface. The interface we used consists of two functions:

\[ \text{validLeg}(a1, a2, t1, t2, m) \text{ and } \text{convertLeg}(\text{leg}) \]

\text{validLeg} assigns for given asteroids a1, a2, times t1, t2 and mass m, if possible a Lambert-arc-chain (Sims Flanagan) representation of a trajectory leg, probably convertible into a continuous thrust leg. \text{convertLeg} does exactly this conversion. Beside efficiency there are two important properties of \text{validLeg}: The probability \( p_{fp} \) a resulting leg is not convertible (false positive) and \( p_{fn} \), the probability a rejected leg is convertible (false negative).

There is a tradeoff between these probabilities and the achievable efficiency of \text{validLeg}. We chose \( p_{fp} \) at about 0.5% and didn't check \( p_{fn} \) yet. Low \( p_{fp} \) means that \text{validLeg} becomes quite expensive. But this way we can store and maintain the whole search tree since its size is limited by our CPU resources. After we found good convertible trajectories we finally increased \( p_{fp} \) to further improve our result, but only the secondary performance index could be raised. Our implementation of \text{validLeg} uses BOBYQA, \text{convertLeg} is based on CMA-ES. Both of these Java implementations were contributed by the author to Apache Commons, but we used enhanced versions written in C called via JNI from Java. Note that \text{validLeg} (contrary to \text{convertLeg}) is deterministic, but the global search using \text{validLeg} is stochastic in most aspects.

A branch in our search tree consists of a trajectory leg for a1, a2, t1, t2, m computed by \text{validLeg}(a1, a2, t1, t2, m) together with the path leading to this leg. A leaf node is
branched by selecting a suitable set of target asteroids \{ta_1, ta_2, \ldots\} "reachable" from a2, and different time values minimizing the leg duration and fulfilling the 30 day requirement. Then \textit{validLeg} is applied to a2,ta_i. The leaf node to be branched next is randomly chosen. The selection is biased towards "promising" nodes by assigning each node a fitness value and chose better nodes with higher probability. The "breadth" of the search is determined by the selection bias and the number of child legs generated at a branch. This stochastic selection search is easily scalable: If multiple threads or machines are available, search results of different threads/machines can easily be joined during or after search. We often took advantage of the option to temporarily halt the search, change parameters and restart to "reshape" the search using logged information.

The efficiency of this method highly depends on the right selection of target asteroids and the fitness function used for the node selection bias. Fitness is chosen only dependent from the time the branch used so far and the depth reached. This aspect of the search was inspired by the "lazy race tree search" from [1]. Since we always maintain the whole search tree, using fuel consumption as part of the fitness function is not necessary. We can select low fuel branches from the resulting tree.

It helps to choose well connected target asteroids with a high probability \textit{validLeg} will succeed. To be able to predict this probability (which we call "connectivity") we generated a \(16256 \times 16256\) sized connectivity-matrix. In a repeated process we chose a1,a2,t1,t2,m randomly with a bias towards shorter transfer times and applied \textit{validLeg}. In the first five days of the competition we applied \textit{validLeg} about \(5 \times 10^{11}\) times on three PCs generating about 70 GB of compressed data, which was used to fill the connectivity-matrix. For the connectivity value assigned to \((a1,a2)\) the number of generated valid legs was used, where the transfers were weighted preferring low duration and high mass. The sum of the row and column values for an asteroid defines its overall connectivity. The process was repeated several times, using the overall connectivity of target asteroids as additional weight. This way "connectivity" propagates from asteroid to asteroid. Iainbanks was chosen as probe target, because it ranked highest.

We used three separate searches, one starting from Pire, one starting from Ovaitt and one starting a backward search from Iainbanks. The resulting Pire and Ovaitt search trees were separately merged with the Iainbanks tree, resulting in trajectories from Pire and Ovaitt to Iainbanks. The search was focused on Asteroids near Iainbanks (in terms of the connectivity-matrix) to increase the probability a forward and backward branch meet at an asteroid to form a valid trajectory. It took about 5 hours using three PCs to find the first three convertible 8-leg trajectories, two from Pire and one from Ovaitt, resulting in \((1+7+0) + (0+7+1) + (1+7+0) = 24\) points, since multiple visits count only once. Unfortunately a 9-leg trajectory was missed by about 20kg fuel, but many 8-leg trajectories with lower fuel consumption were found. The resulting trajectories were "tweaked" using a fuel preserving variant of \textit{convertLeg}.

\textit{Literature}