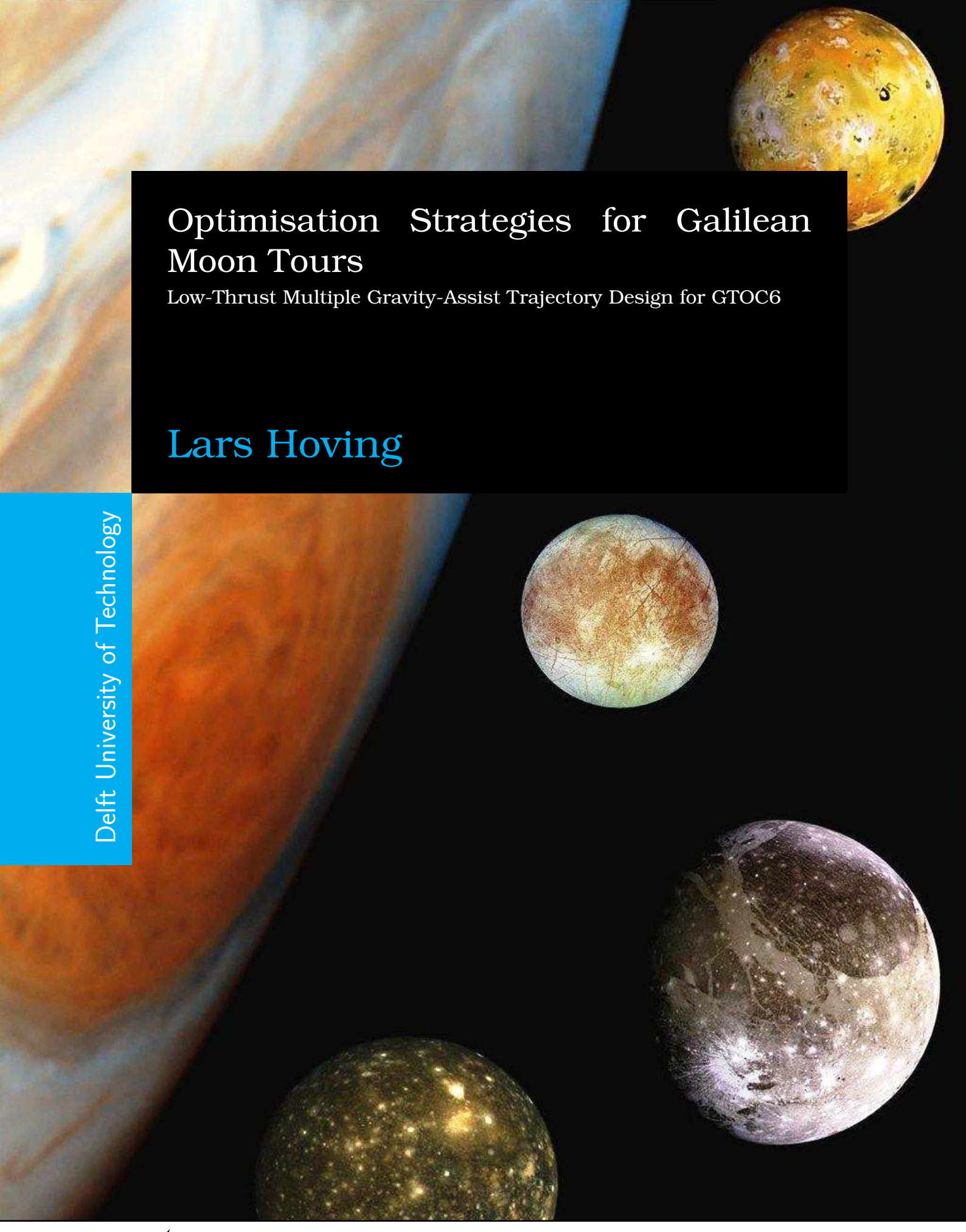

Optimisation Strategies for Galilean Moon Tours

Low-Thrust Multiple Gravity-Assist Trajectory Design for GTOC6

Lars Hoving

Delft University of Technology

TU Delft Delft University of Technology

Optimisation Strategies for Galilean Moon Tours

Low-Thrust Multiple Gravity-Assist Trajectory Design for GTOC6

by

L. Hoving

in partial fulfilment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on Tuesday December 8, 2015 at 2:00 PM.

Supervisor:	Ir. R. Noomen	TU Delft
Thesis committee:	Prof. dr. ir. P.N.A.M. Visser	TU Delft
	Dr. ir. H.G. Visser	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Cover image is adopted from *http://www.realworldglobes.com/gallery/products/jupiter-gm/galilean-moons_01m.jpg*. The image depicts the planet Jupiter with its characteristic big red spot and the Galilean moons Io, Europa, Ganymede and Callisto.

Preface

THIS thesis is the culmination of the research performed at the Department of Astrodynamics and Space Mission Analysis at Delft University of Technology, and is in partial fulfilment of the Master of Science in Aerospace Engineering degree.

The subject of the thesis project can be chosen freely within the boundaries of space exploration. This allows for many different types of topics. However, my main interest is in interplanetary trajectory design. Also, avoiding complexity is not part of my vocabulary and therefore I chose the sixth edition of the Global Trajectory Optimisation Competition as topic. The competition was organised by JPL of NASA and is focussed on global trajectory optimisation in the Galilean moon system of Jupiter. Characteristics of the subject are low-thrust propulsion and multiple gravity-assists which are to be used to map the Galilean moons in a given time frame. The specific research is focussed on optimisation strategies for finding optimal trajectories in the Galilean moon system satisfying the GTOC6 constraints.

The research and the preceding literature study would not have been possible without the support from many people. First of all, I would like to thank my supervisor ir. Ron Noomen for the many joyful and insightful meetings, but also for giving me focus and motivation when I needed it. Furthermore, my thanks also go to my friends on the 9^{th} floor for the fruitful discussions on interplanetary mission design and mainly many other topics for which the purposes were beyond comprehension. Finally, I would like to thank my family and friends for their support and the numerous coffee breaks to keep me going.

Lars Hoving,

November 12, 2015

Abstract

AROUND 1610 Galileo Galilei made his discovery of the four large moons orbiting Jupiter which are referred to now as the Galilean moons. The moons Europa and Ganymede attract a significant amount of scientific interest due to potential present subsurface oceans. As consequence, the design of missions to go there and explore the moon system are increasing. This culminated in the sixth edition of the Global Trajectory Optimisation Competition (GTOC6) which is focussed on solving low-thrust multiple gravity-assist trajectories to map the Galilean moons. The aim of the thesis is to understand the complexity of the GTOC6 problem and to explore and evaluate the quality of various optimisation strategies to solve flyby sequences with low-thrust arcs.

First, insight to the complexity of the problem was gained by analysing the best solution to GTOC6 so far by the Chinese Centre for Space Utilisation (CSU). From the results a clear picture was drawn from what the trajectory model should be capable of. The low-thrust trajectory model is based on the spherical shaping method that is part of the Tudat astrodynamics toolbox. A full analysis of the shaping method was performed to identify the capabilities and shortcomings of the algorithm. One of the main shortcomings is the limited accuracy for trajectories where the departure and arrival conditions differ with several degrees and more for the right ascension of the ascending node (RAAN).

For optimisation use was made of differential evolution (DE). An extensive test was performed to determine the optimal settings. The result was that defining the control parameters randomly during the evolution was the best option with respect to quality and convergence. What followed was defining the optimisation model for a variable number of flybys. Furthermore, a framework was developed with six different optimisation strategies. A sequence of maximum five flybys was set to test the strategies. The strategies define the amount of freedom around the epochs of the flybys for the optimisation. Also the number of flybys that are influenced by this freedom is defined by the strategy.

The goal was to optimise for ΔV for a main sequence of five flybys. Here the main sequence was divided into smaller problems (subsets with less flybys). The optimisation of the main sequence was guided by the solutions of the preceding smaller subsets. Results showed that the initial subset of two flybys did not influence the optimisation of the subsequent subsets at all. Furthermore, two sequences were tested. The first sequence showed large ΔV due to thrust constraint violations and limited accuracy of the spherical shaping method. On the other hand the second sequence showed ballistic solutions to go through all five moons in the sequence.

Finally, from the previous test resulted an optimal strategy that was applied to a sub-problem of GTOC6. Optimisation was set to map the most interesting surfaces of the moons and to minimise ΔV. The resulting trajectories were able to map the surfaces of interest. However, at the cost of more ΔV compared to the previous test which only optimised for ΔV.

Contents

List of Symbols

Symbols and parameters that represent the vector form are printed in boldface in this report. Some symbols have units that depend on the input parameters or can not be described by one unit like the orbital elements vector. These units are denoted with PD which stands for problem dependent.

Symbol	Description	Unit
	Latin letters	
A	Spacecraft-centred reference frame	[-]
a	Dimensionless acceleration parameter	[-]
a	Semi-major axis	[m]
$\hat{\boldsymbol{a}}$	Spacecraft-centred reference frame unit vector	[-]
$\boldsymbol{B}$	Aiming point vector	[m]
B	Body-fixed reference frame	[-]
$\hat{\boldsymbol{b}}$	Body-fixed reference frame unit vector	[-]
C	Constant boundary conditions	[-]
CR	Crossover control parameter DE	[-]
D	Day	[-]
D	Problem dimension, number of decision parameters	[-]
D	Time shaping function	[-]
E	Eccentric anomaly	[rad]
$\mathcal{E}$	Specific (orbital) energy	[m^2/s^2]
e	Eccentricity	[-]
$\boldsymbol{e}$	Eccentricity vector	[-]
e	Truncation error	[PD]
F	Control parameter DE (differential weight)	[-]
F	Hyperbolic anomaly	[rad]
F_{V_i}	Face value	[-]
$\boldsymbol{f}$	Acceleration vector	[m/s^2]
f	Function	[PD]
f	Modified equinoctial element	[-]
f_h	Out-of-plane thrust	[N]
f_N	Acceleration perpendicular to the radius vector	[m/s^2]
f_S	Outward radial acceleration	[m/s^2]
f_W	Acceleration perpendicular to the orbital plane in direction of the angular momentum vector	[m/s^2]
G	Galilean moon reference frame	[-]
G	Gravitational constant (6.670×10^{-11})	[$m^3/kg/s^2$]
$\hat{\boldsymbol{g}}$	Galilean moon reference frame unit vector	[-]
g	Modified equinoctial element	[-]
g_0	Nominal gravitational acceleration at the surface of the Earth (9.80665)	[m/s^2]
H or h	Angular momentum per unit mass	[m^2/s]

Symbol	Description	Unit
$\boldsymbol{H}$ or $\boldsymbol{h}$	Angular momentum vector per unit mass	[m^2/s]
H or h	Step-size	[s]
h	Hour	[-]
h	Modified equinoctial element	[-]
h_{pS}	Fly-by altitude	[m]
I_{sp}	Specific impulse	[s]
i	Inclination	[rad]
J	Objective function	[PD]
J	Jovicentric reference frame	[-]
$\boldsymbol{K}$	Unit vector [0, 0, 1]	[-]
k	Modified equinoctial element	[-]
L	True longitude	[rad]
M	Mean anomaly	[rad]
M	Month	[-]
m	Mass (spacecraft)	[kg]
m	Minutes	[-]
m	Number of spacecraft orbits (resonance)	[-]
m_{min}	Minimum spacecraft mass	[kg]
$m_{pen,\,k+1}$	Mass penalty	[kg]
N	Number of close approaches to Jupiter	[-]
N_{fby}	Number of fly-bys	[-]
N_G	Grand total number of simulation	[-]
N_S	Selection total of simulation	[-]
NP	Number of individuals in population	[-]
n	Mean angular motion	[rad/s]
$\boldsymbol{n}$	Nodal vector	[-]
n	Number of moon orbits (resonance)	[-]
P	Position	[m]
p	Semi-latus rectum	[m]
$R_{CAJ\,min}$	Minimum distance to Jupiter constraint	[m]
R_J	Radius of Jupiter	[m]
R_S	Radius of a moon	[m]
$R_{s.i.}$	Radius of sphere of influence	[m]
$R(\theta)$	Shaping function spherical shaping	[m]
$\boldsymbol{r}$	Position vector	[m]
r	Radial distance	[m]
r_a or r_{ai}	Osculating apoapis radius	[m]
r_p or r_{pi}	Osculating periapsis radius	[m]
$\boldsymbol{r}_{pS}$	Pericentre vector of the fly-by with respect to the moon	[m]
s	Seconds	[-]
s	Semi-parameter	[m]
T	Kinetic energy per unit mass	[m^2/s^2]
T	Orbital period	[s]
T	Normalised time-of-flight	[-]
T	Thrust	[N]
$T_{departure}$	Orbital period of departure orbit	[s]
T_{syn}	Synodic period	[s]
T_{target}	Orbital period of target orbit	[s]
$T(\theta)$	Shaping function spherical shaping	[s]

Symbol	Description	Unit
t	Time	[s]
t_G	Time of the gravity-assist	[s]
t_{G+}	Time just after the gravity-assist	[s]
t_{G-}	Time just before the gravity-assist	[s]
$t^+_{G_{k+1}}$	Time just after gravity-assist $(k+1)$	[s]
$t^-_{G_{k+1}}$	Time just before gravity-assist $(k+1)$	[s]
$\boldsymbol{u}$	Trial vector DE	[PD]
V	Potential energy per unit mass	[m^2/s^2]
V or v	Velocity	[m/s]
$\boldsymbol{v}$	Mutant vector DE	[PD]
$\boldsymbol{v}$	Velocity vector (spacecraft)	[m/s]
v_c	Circular velocity	[m/s]
v_{esc}	Hyperbolic excess velocity	[m/s]
$\boldsymbol{v}_S$	Escape velocity	[m/s]
v_∞	Hyperbolic excess velocity	[m/s]
$\boldsymbol{v}_\infty$	Hyperbolic excess velocity vector	[m/s]
$\boldsymbol{v}_{\infty G+}$	Hyperbolic excess velocity vector just after the gravity-assist	[m/s]
$\boldsymbol{v}_{\infty G-}$	Hyperbolic excess velocity vector just before the gravity-assist	[m/s]
W_S	Face value multiplier	[-]
w	Dimensionless velocity parameter	[-]
x	Cartesian coordinate along the X-axis	[m]
x	Normalised semi-major axis	[-]
$\boldsymbol{x}$	Position vector (spacecraft)	[m]
$\boldsymbol{x}$	Target vector DE	[PD]
$\boldsymbol{x}_S$	Position vector of the moon	[m]
Y	Year	[-]
y	Cartesian coordinate along the Y-axis	[m]
z	Cartesian coordinate along the Z-axis	[m]
	Greek letters	
α	Asymptotic deflection angle	[rad]
α	In-plane orientation angle	[rad]
α	Pump angle	[rad]
β	Angle between major body's velocity and the symmetry line of the hyperbolic trajectory	[rad]
β	Out-of-plane orientation angle	[rad]
β	Rotation angle in B-plane	[rad]
γ	Flight path angle (spacecraft)	[rad]
Δ	Difference	[-]
δ	Change of orbital element with time	[PD]
δ	Turn angle gravity-assist	[rad]
η	Fraction of the transfer time	[-]
η	Approximated value	[PD]
θ	Polar angle	[rad]
θ	True anomaly	[rad]
$\bar{\theta}$	Total polar angle	[rad]
μ	Gravitational parameter	[m^3/s^2]
π	Ratio of a circle's circumference to its diameter ($\sim$3.14159)	[-]
ρ	Dimensionless distance parameter	[-]
τ	Time of the last pericentre passage	[s]

Symbol	Description	Unit
τ	Dimensionless time parameter	[-]
Υ	Reference direction	[-]
$\Phi(\theta)$	Shaping function spherical shaping	[rad]
ϕ	Angle between reference direction and position vector	[rad]
ϕ	Phase angle	[rad]
ϕ	Angle between XY-plane and position vector in spherical coordinate system	[rad]
ϕ	Total rotation angle between incoming and outgoing excess-velocity vector	[rad]
Ψ	Angle between two position vectors	[rad]
Ω	Right ascension of the ascending node	[rad]
ω	Argument of pericentre	[rad]
ω	Rotation rate	[rad/s]
	Other symbols	
#	Number	[-]
∂	Partial derivative	[-]
	Subscripts	
0	Initial	
1, 2, 3	Coordinates along respectively first, second and third axis	
$A \rightarrow J$	Rotation from spacecraft-centred to Jovicentric reference	
a	Apocentre	
a	Arrival	
a_{in}	In-plane acceleration	
ap	Approach	
a_z	Out-of-plane acceleration	
$body$	Body of interest	
d	Departure	
f	Final	
G	Galilean moon reference frame	
G	Generation	
G	Gravity-assist	
$G \rightarrow J$	Rotation from Galilean moon to Jovicentric reference frame	
h	Bi-normal direction	
i	Body i	
i	Element i	
J	Jupiter	
$J \rightarrow A$	Rotation from Jovicentric to spacecraft-centred reference frame	
$J \rightarrow G$	Rotation from Jovicentric to Galilean moon reference frame	
j	Body j	
ji	j with respect to i	
k	Kinetic	
M	Moon	
max	Maximum	
min	Minimum	
n	Dimension	
new	New value	
nom	Nominal	
p	Pericentre	
p	Potential	

Symbol	Description	Unit
pen	Penalty	
$post$	After	
pre	Before	
$prop$	Propulsion	
r	Radial direction	
rel	Relative	
rot	Rotation	
sc	Spacecraft	
$surf$	Surface	
tot	Total	
x, y, z	Coordinates along respectively X-, Y- and Z-axis	
θ	Transversal direction	
ϕ	Out-of-plane direction	
	Superscripts	
$'$	Derivative with respect to the polar angle	
$+$	Just after	
$-$	Just before	
T	Transpose	
$\hat{}$	Unit vector	
$\tilde{}$	Normalised	
$\dot{}$	Derivative with respect to time	
$\ddot{}$	Second derivative with respect to time	

List of Abbreviations

In alphabetical order.

Abbreviation	Description
ACT	Advanced Concepts Team
AN	Ascending Node
AU	Astronomical Unit
B&B	Branch and Bound
BC	Before Christ
BC	Boundary Condition
Ca	Callisto
CPU	Central Processing Unit
CSU	Centre for Space Utilisation
DE	Differential Evolution
DOPRI	Runge-Kutta-Dormand-Prince integrator
DOPRI8	8^{th}-order Runge-Kutta integrator with 7^{th}-order error estimation
DSM	Deep-Space Manoeuvre
ESA	European Space Agency
Eu	Europa
GA	Genetic Algorithm
GA	Gravity-Assist
Ga	Ganymede
GASP	Gravity Assist Space Pruning
GTOC	Global Trajectory Optimisation Competition
GTOC6	Sixth edition of the Global Trajectory Optimisation Competition
HKUST	Hong Kong University of Science and Technology
JD	Julian Date
jDE	Self-Adaptive Differential Evolution
JDN	Julian Date Number
JPL	Jet Propulsion Laboratory
LAN	Longitude of the Ascending Node
MATLAB	Matrix Laboratory
MEE	Modified Equinoctial Elements
MGA	Multiple Gravity-Assist
MJD	Modified Julian Date
MLTGA	Multiple Low-Thrust Gravity-Assist
MOEA/D	Multi-Objective Evolutionary Algorithm Decomposition
NA	Not Available
NaN	Not a Number
NASA	National Aeronautics and Space Administration
PaGMO	Parallel Global Multi-objective Optimiser
PSO	Particle Swarm Optimisation

Abbreviation	Description
RAAN	Right Ascension of the Ascending Node
RFT	Reference Frame Transformation
RK4	4^{th}-order Runge-Kutta integrator
RK7(8)	7^{th}-order Runge-Kutta integrator with 8^{th}-order error estimation
RK8(7)-13M	8^{th}-order Runge-Kutta integrator with 7^{th}-order error estimation
RKDP	Runge-Kutta-Dormand-Prince integrator
RKF	Runge-Kutta-Fehlberg integrator
RKN	Runge-Kutta-Nyström integrator
SOI	Sphere Of Influence
TOF	Time-of-flight
TU	Technical University
Tudat	TU Delft Astrodynamics Toolbox
USA	United States of America

1 Introduction

AROUND 1610 Galileo Galilei made his discovery of the four large moons orbiting Jupiter. Though he named the moons Medicea Sidera after the four Medici brothers, the current names, after the lovers of god Zeus, were given by Simon Marius who discovered the moons around the same time (McFadden et al., 2006). However, the moons Io, Europa, Ganymede and Callisto still have a reference to Galilei as the group of four are called the Galilean moons.

Petropoulos (2012) proposed the sixth global trajectory optimisation problem. The goal of GTOC6 is to find the optimal trajectory to globally map the Galilean moons using fly-bys and low-thrust propulsion. Additional constraints like radiation and mission duration complicate the matter in effect to create a realistic mission scenario. The problem has still not been fully solved and the best solution so far is by the Chinese Centre for Space Utilisation (He and Gao, 2014). Besides solving it GTOC6 brings challenges for new trajectories in complex moon systems using low thrust. With more attention these days for the Galilean moons, like Europe and Ganymede, it is very interesting to do more research in the area of complex low-thrust multiple gravity-assist trajectories.

The aim of this master thesis is to understand the complexity of the GTOC6 problem and to explore various optimisation strategies to solve flyby sequences. It is the combination of flybys with continuously acting thrust that creates a vast and challenging search space for the optimisation process. Instead of solving the problem itself, the focus is on the development of strategies to find optimal solutions in an efficient way. The research question for this thesis is defined as follows:

> *Evaluate the quality of different optimisation strategies for Galilean moon mapping trajectories using one-body dynamics, gravity-assist manoeuvres and low-thrust propulsion.*

For the low-thrust arcs in the trajectory use is made of the spherical shaping algorithm developed by Roegiers (2014) and from the mind of Novak (2012). Constraints are supplied by the GTOC6 problem and effective optimisation techniques are derived from an extensive performed analysis of the current best solution for GTOC6.

The thesis report is divided into four major parts. In part I, "Problem Statement & Heritage", the GTOC6 problem is detailed and the research objective is given. This is discussed in

Chapter 2 followed by Chapter 3 where a thorough analysis of the CSU solution is made to prepare for the optimisation problem.

In the second part "Low-Thrust Multiple Gravity-Assist Trajectory Model" all content related to making the trajectories is given. First the spherical shaping method is explained, followed by in-depth verification of the algorithm in Chapter 4. Secondly, Chapter 5 elaborates on the multiple gravity-assist manoeuvre and the second part finishes with Chapter 6 where details on mapping the moons according to the GTOC6 definition are given.

With part III "Optimisation & Validation" all aspects related to the optimisation algorithm, strategies and techniques are worked out. The former is explained in Chapter 7 and validated in Chapter 8, whereas the latter are elaborated upon in Chapter 9. In that same chapter also the problems are defined with respect to GTOC6 to test optimisation strategies.

Finally the results of the previous test are worked out and explained in Chapter 10 which is in part IV "Results & Conclusions". The part is wrapped up with the conclusions and recommendations.

Part I

Problem Statement & Heritage

2

Problem Statement and Research Objectives

GTOC6 is the sixth edition of the Global Trajectory Optimisation Competition (GTOC) and is the subject for the research in this Master of Science thesis. In the next section the original problem is described. This is followed by the actual research focus and objectives that were derived from the GTOC6 problem. The last section deals with assumptions and constraints that are applied on top of the GTOC6 problem to narrow down the research scope.

2.1 GTOC6 - Original Problem Description

The problem of GTOC6 is the global mapping of Jupiter's Galilean moons. Here the global mapping is performed by a low-thrust propulsion spacecraft utilising multiple close flybys. The description in this section follows from the official GTOC6 problem description (Petropoulos, 2012).

To simplify the problem a few assumptions are made. First the dynamics of the system are simplified by excluding the Sun's gravity and modelling Jupiter as a point mass. The moons around Jupiter are assumed to follow conical orbits. Furthermore the flybys are to be modelled as instantaneous manoeuvres and the moons do not affect the trajectory otherwise. In other words there are no perturbations present, which simplifies the problem significantly. The assumptions and definitions are described in this subsection.

The motion of the spacecraft around Jupiter is modelled with one-body dynamics and the added low thrust (Petropoulos, 2012). This results in Equations (2.1) and (2.2) for respectively the acceleration and mass.

$$\frac{\mathrm{d}^2\boldsymbol{r}}{\mathrm{d}t^2} + \mu\frac{\boldsymbol{r}}{r^3} = \frac{\boldsymbol{T}}{m} \tag{2.1}$$

$$\frac{\mathrm{d}m}{\mathrm{d}t} = -\frac{T}{I_{sp}g_0} \tag{2.2}$$

The departure date of the mission is to be selected between the years 2020 and 2030. Furthermore, the departure position of the spacecraft is in the vicinity of Jupiter. In other words the optimal trajectory does not involve the transfer trajectory to get to Jupiter but only the

mapping trajectory through the moon system of Jupiter. The start position is set to a radial distance of 1000 Jupiter radii R_J with respect to the centre of Jupiter. Additionally, the velocity of the spacecraft at departure is set to 3.4 km/s in arbitrary direction. The duration of the mission is defined as the time lapsed between the initial start date and the time at which the last flyby is performed by the spacecraft. This time-of-flight is set to a maximum of four years.

The four Galilean moons, Jupiter and the spacecraft are defined in a Jovicentric coordinate reference frame that coincides with the Mean Equator and the Prime Meridian of Jupiter at epoch 58 849.0 MJD. The orbital elements of the moons at this epoch can be found in Table A.1 in Appendix A. Physical constants and other constants related to the Jovian system can be found in the same appendix in Tables A.2 and A.3. For the purpose of mapping the moons a body-fixed coordinate frame is used for each moon. The b_1 axis is pointing from the moon to Jupiter and the b_3 axis aligns with the angular momentum vector, which is normal to the moon's orbit plane. The last axis, b_2, is perpendicular to the plane made up by the axis b_1 and b_3 and follows from the right-handed rule. Note that due to tidal locking (Io, Europa and Ganymede) and the equal rotation periods of the orbit and the moon itself (Callisto), the body and its surface are fixed with respect to the coordinate reference frame. With this the moons can be mapped using the body-fixed reference system which is rotating in the Jovicentric reference frame. At time t_G the body-fixed unit vectors are given by Equations (2.3), (2.4) and (2.5).

$$\hat{\boldsymbol{b}}_{\boldsymbol{1}} = -\frac{\boldsymbol{x_M}\left(t_G\right)}{\left|\boldsymbol{x_M}\left(t_G\right)\right|} \tag{2.3}$$

$$\hat{\boldsymbol{b}}_{\boldsymbol{3}} = \frac{\boldsymbol{x_M}\left(t_G\right) \times \boldsymbol{v_M}\left(t_G\right)}{\left|\boldsymbol{x_M}\left(t_G\right) \times \boldsymbol{v_M}\left(t_G\right)\right|} \tag{2.4}$$

$$\hat{\boldsymbol{b}}_{\boldsymbol{2}} = \hat{\boldsymbol{b}}_{\boldsymbol{3}} \times \hat{\boldsymbol{b}}_{\boldsymbol{1}} \tag{2.5}$$

Here $\boldsymbol{x_M}$ and $\boldsymbol{v_M}$ are vectors describing respectively the position and velocity of the moon with respect to the coordinate reference frame.

The success of global mapping of the moons is described by maximising a performance index. To determine this performance index, each moon is divided into a certain number of smaller surfaces defined by the uniform icosahedron, see Figure 2.1. This shape is also known as the football grid (Buckminsterfullerene or bucky-ball) made up by pentagons and hexagons. The coordinates in the body-fixed reference frame of the vertices that make up the hexagons and pentagons can be found in Tables B.1 and B.2 in Appendix B. There are 20 hexagons and 12 pentagons making a total of 32 faces. Each face is a unique part of the surface of the moon and has a certain value F_{V_i}. By performing close flybys over these faces the performance index is increased. The performance index is given in Equation (2.6) and is also called the Total Face Value.

$$J = \sum_{i=1}^{N_{fby}} W_S F_{V_i} \tag{2.6}$$

Here W_S is the face value multiplier of the moon stating the importance of mapping that particular moon, and N_{fby} is the number of close flybys. The values for W_S are 1, 2, 1 and 1 for respectively Io, Europa, Ganymede and Callisto flybys. Note that the moon Europa is of particular scientific interest and therefore has a value of two. Furthermore, the face values F_{V_i} also depend on which moon is being mapped. These values can be found in Table 2.1. A maximum objective score of 324 points can be achieved when all faces are mapped. In Figure

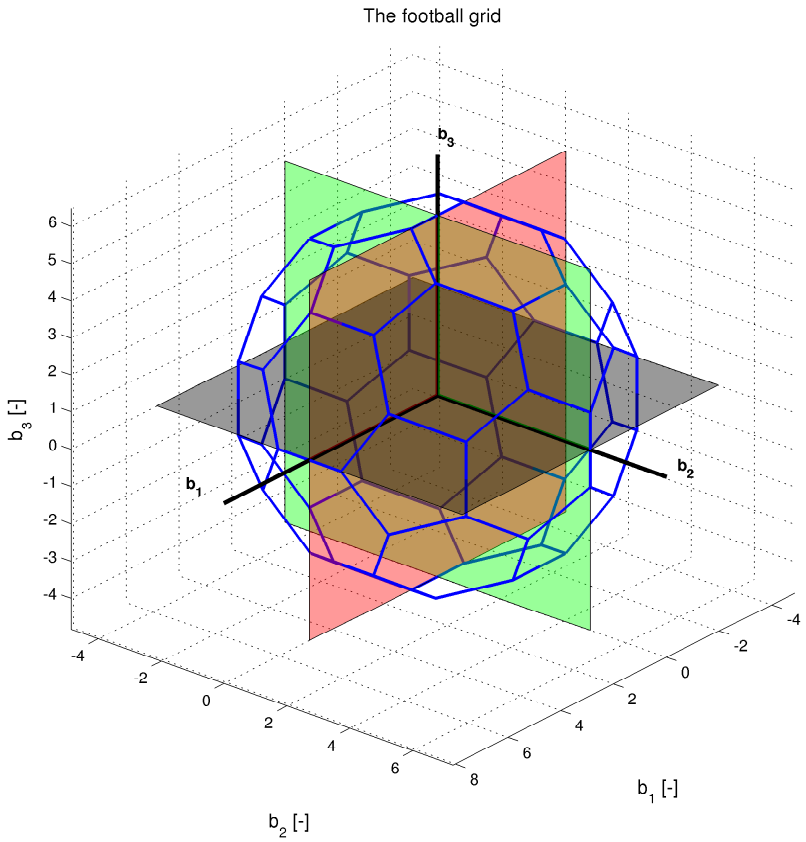

Figure 2.1: *Three-dimensional representation of football grid and body-fixed coordinate axes. Adapted from Petropoulos (2012).*

2.2 a longitude/latitude presentation is given of the football grid in which the faces with their numbers are displayed.

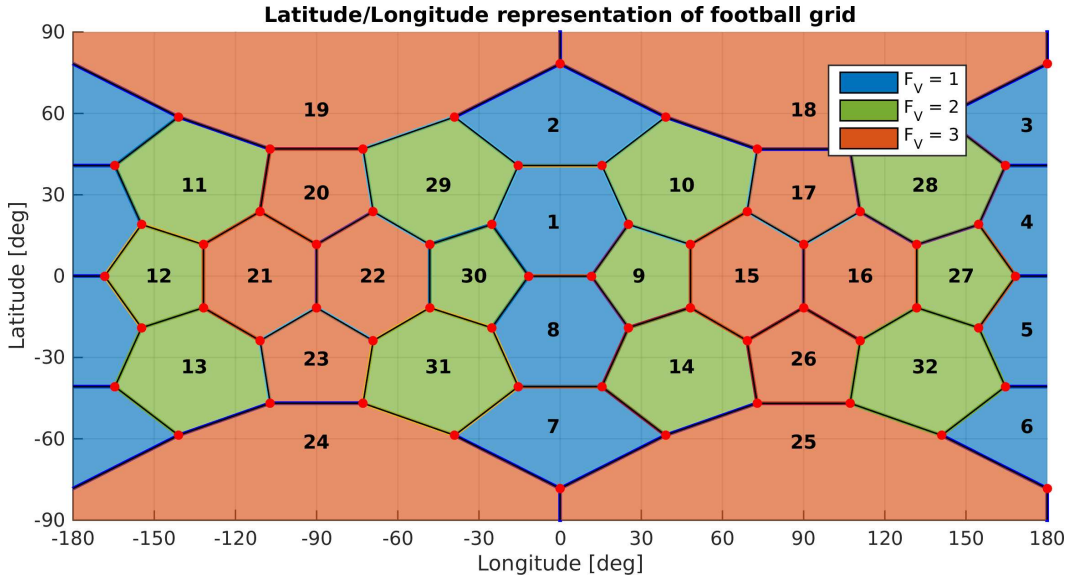

Figure 2.2: *Longitude/latitude representation of football grid. Adapted from Petropoulos (2012).*

Table 2.1: *Face values (Face numbers are illustrated in Figure 2.2).*

Face numbers	F_V for Io, Europa	F_V for Ganymede, Callisto	F_V not new faces
1-8	1	3	0
9-14, 27-32	2	2	0
15-26	3	1	0

The performance index is only increased when a close flyby is executed. A close flyby is defined by the minimum altitude which must be in the range given in Equation (2.7).

$$50\,\text{km} \leq h_{pS} \leq 2000\,\text{km} \tag{2.7}$$

Outside this range the instruments on the spacecraft are assumed not to be able to map the surface of the particular face well as being too close to or too far from the surface of the moon. Furthermore, the pericentre of the flyby needs to be above the surface of the face. In other words, the vector from the centre of the moon to the pericentre is inside the pyramid defined by the centre of the moon and the base of the pyramid. This base is defined by the vertices of either the hexagon or pentagon face. When the vector goes through the boundary of the base, that is the edge or corner (vertex), the face with the highest value connected to the edge or vertex is considered. In case of a close flyby of an old face, that has been counted already, the performance index is not affected. The same holds for when the flyby is outside the range in Equation (2.7) and is therefore not a proper close flyby for scoring points.

Other constraints that apply to the spacecraft and trajectory are related to propulsion, mass and radiation protection. The latter translates into the stringent constraint that the distance of the spacecraft to Jupiter has to be larger than or equal to two times the radius of Jupiter R_J, see Equation (2.8).

$$r \geq R_{CAJ\,min} = 2R_J \tag{2.8}$$

Due to severe radiation that could damage the spacecraft significantly, solutions that violate this constraint are considered catastrophic and therefore invalid.

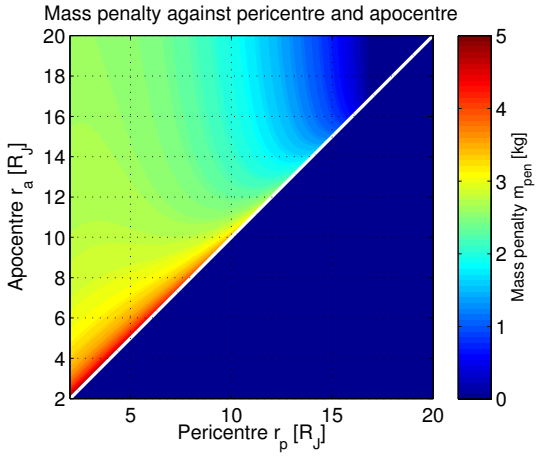

Figure 2.3: *Mass penalty for closed orbits as a function of the osculating periapsis radius r_{pi} and the apoapsis radius r_{ai}. The white line defines the circular orbit boundary.*

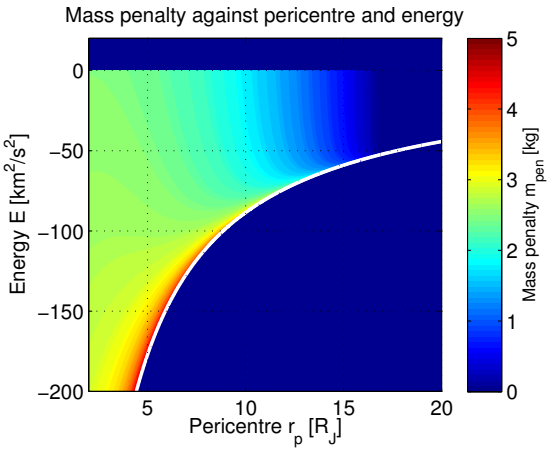

Figure 2.4: *Mass penalty for orbits as a function of the osculating periapsis radius r_{pi} and the energy of the orbit E. The white line defines the circular orbit boundary.*

The spacecraft makes use of low-thrust propulsion which provides a maximum thrust of 0.1 N with a specific impulse of 2000 s. Also the thrust is available in all directions and at all times. The start mass of the spacecraft is 2000 kg and the amount of available propellant is set to 1000 kg. This sets the following mass constraint on the spacecraft.

$$m \geq m_{min} = 1000\,\mathrm{kg} \tag{2.9}$$

In addition to thrusting, the mass of the spacecraft can be decreased by mass penalties to account for necessary radiation protection. Mass penalties are awarded when the spacecraft's trajectory is in close proximity to Jupiter. Close proximity is defined as having the oscillating periapsis between two and 17 Jupiter radii. Below two Jupiter radii was already considered catastrophic with respect to radiation. The relation for the mass penalty due to this violation is given in Equation (2.10).

$$\begin{aligned} m_{pen,\,k+1} = 5 \sum_{i=1}^{N} \left[1 - \left(\frac{\frac{r_{pi}}{R_J} - 2}{15} \right)^2 \right] \cdot \\ \left(1 + \frac{1}{1 + \frac{r_{ai}}{R_J} - \frac{r_{pi}}{R_J}} \right) \left[\left(1 + \mathrm{sign}\left(r_{ai}\right)\right) \frac{1 + \mathrm{sign}\left(17 - \frac{r_{pi}}{R_J}\right)}{4} \right] \end{aligned} \tag{2.10}$$

$$m\left(t^{+}_{G_{k+1}}\right) = m\left(t^{-}_{G_{k+1}}\right) - m_{pen\,,k+1} \tag{2.11}$$

The relation determines the mass penalty between two consecutive flybys. These flybys occur at times t_{G_k} and $t_{G_{k+1}}$ and the penalty $m_{pen,\,k+1}$ is defined for $t_{G_k} \leq t < t_{G_{k+1}}$. In the equation, N is the number of close approaches to Jupiter which occur between the two flybys. A close approach represents a local minimum in the range to the centre of Jupiter. For each i^{th} close approach the osculating periapsis r_{pi} and apoapsis radius r_{ai} with respect to Jupiter are determined based on the position and velocity of the close approach.

Not to mention, as follows from Equation (2.10), the mass penalty is valid for close approaches where the perijove (r_{pi}) is equal or lower than 17 times the radius of Jupiter. Both r_{pi} and r_{ai} influence the amount of mass penalty. Hyperbolic and parabolic conditions are excluded from the mass penalty computation. In Figure 2.3 the mass penalty is shown for various closed

orbits whereas Figure 2.4 shows all kinds of orbits. Note the white lines (diagonal and curved) separating the figures in two parts. They represent the circular orbit boundary and the area south-east of those lines has no physical meaning.

The mass penalty $m_{pen,\,k+1}$ for $t_{G_k} \leq t < t_{G_{k+1}}$ is subtracted from the spacecraft mass immediately after the next flyby at $t_{G_{k+1}}$. The latter is defined as $m\left(t_{G_{k+1}^+}\right)$ whereas the mass penalty during the time interval right before the next flyby is written as $m\left(t_{G_{k+1}^-}\right)$. The subtraction of mass is also described in Equation (2.11). In case a close approach occurs at $t_{G_{k+1}}$ the mass penalty is added to the next flyby at $t_{G_{k+2}}$ and is thus not part of $m\left(t_{G_{k+1}^-}\right)$, but of $m\left(t_{G_{k+2}^-}\right)$ which results in $m\left(t_{G_{k+2}^+}\right)$.

The last part of the problem description is the definition of the flybys using the patched-conic approximation. The spacecraft approaches the target moon and enters the sphere of influence of the moon. Inside the sphere of influence the spacecraft follows a hyperbolic trajectory around the moon and then leaves the sphere of influence. The time spent inside the sphere of influence is neglected in this problem, because this duration is very small compared to the orbital periods on Jovicentric scale. Therefore at time t_G of the flyby the Jovicentric position of the spacecraft has to be equal to the position of the moon up to a 1 km accuracy, see Equation (2.12).

$$|\boldsymbol{x}\left(t_G\right) - \boldsymbol{x_M}\left(t_G\right)| \leq 1\,\mathrm{km} \tag{2.12}$$

The mathematical definition for the position of the spacecraft and the moon around the flyby at time t_G is given in Equation (2.13).

$$\boldsymbol{x}\left(t_{G-}\right) = \boldsymbol{x}\left(t_{G+}\right) = \boldsymbol{x_M}\left(t_{G-}\right) = \boldsymbol{x_M}\left(t_{G+}\right) \tag{2.13}$$

Next, the velocity of the spacecraft immediately before the flyby at t_{G-} determines the incoming hyperbolic excess velocity $\boldsymbol{v_{\infty G-}}$ relative to the moon. Immediately after the flyby at t_{G+} the spacecraft has reached the outgoing hyperbolic excess velocity $\boldsymbol{v_{\infty G+}}$ which has the same magnitude as the incoming hyperbolic excess velocity. Note that the velocity of the spacecraft experiences a discontinuous change when performing a flyby. The computation of the excess velocities has a tolerance of 1 m/s, see Equation 2.14.

$$|\boldsymbol{v_{\infty G+}}| - |\boldsymbol{v_{\infty G-}}| \leq 1\,\mathrm{m/s} \tag{2.14}$$

Furthermore, the hyperbolic excess velocity is constrained to a minimum value defined by Equation (2.15)

$$v_\infty \geq 0.3\,\mathrm{km/s} \tag{2.15}$$

The turn angle δ between incoming and outgoing velocities follows from Equations (2.16), (2.17), (2.18) and (2.19).

$$\boldsymbol{v_{\infty G-}} = \boldsymbol{v}\left(t_{G-}\right) - \boldsymbol{v_M}\left(t_{G-}\right) \tag{2.16}$$

$$\boldsymbol{v_{\infty G+}} = \boldsymbol{v}\left(t_{G+}\right) - \boldsymbol{v_M}\left(t_{G+}\right) \tag{2.17}$$

$$|\boldsymbol{v_{\infty G+}}| = |\boldsymbol{v_{\infty G-}}| = v_\infty \tag{2.18}$$

$$\boldsymbol{v_{\infty G+}} \cdot \boldsymbol{v_{\infty G-}} = v_\infty^2 \cos\delta \tag{2.19}$$

From the turn angle follows the altitude of the hyperbolic at pericentre which is given in Equation (2.20).

$$\sin\left(\frac{\delta}{2}\right) = \frac{\frac{\mu_S}{R_S+h_{pS}}}{v_\infty^2 + \frac{\mu_S}{R_S+h_{pS}}} \tag{2.20}$$

The vector pointing to the pericentre of the hyperbolic trajectory with respect to the moon is given by Equation (2.21).

$$\boldsymbol{r_{pS}} = (R_S + h_{pS})\,\frac{\boldsymbol{v_{\infty G-}} - \boldsymbol{v_{\infty G+}}}{|\boldsymbol{v_{\infty G-}} - \boldsymbol{v_{\infty G+}}|} \tag{2.21}$$

2.2 Research Objectives

The focus of this thesis research is not to find an improved solution to the GTOC6 problem. Although a solution with the maximum score has not been found yet, several prestigious teams have tried and are trying to improve the previous best score and some even succeeding. This shows that the problem is of great interest by many and that the search for the optimum solution has not stopped yet.

The focus of this thesis is on preliminary trajectory optimisation in complex moon systems. More specifically, the main interest is focussed on finding effective ways to optimise moon sequences using low-thrust propulsion and flybys. Especially the latter are of great influence on the trajectory efficiency while the number of flybys and the low-thrust propulsion take care of the complexity of the problem. The GTOC6 problem prescribes the use of one-body dynamics, and that the moons alter the trajectory with an instantaneous flyby and that they do not affect the trajectory otherwise. From this the following main research objective is defined.

Evaluate the quality of different optimisation strategies for Galilean moon mapping trajectories using one-body dynamics, gravity-assist manoeuvres and low-thrust propulsion.

Aspects to be addressed here are:

- Develop astrodynamic tools to design low-thrust multiple gravity-assist trajectories in both the Solar and the Jovian system.
- Develop a global optimisation method to find the global low-thrust arc between two consecutive flybys.
- Extend the global optimisation method to incorporate multiple gravity-assist manoeuvres.
- Design optimisation strategies and develop the algorithm to incorporate the previous described methods into one strategy optimisation algorithm.
- Compare the moon mapping trajectory results with other known solutions.
- Find the optimal strategy for the GTOC6 problem.

2.3 Assumptions and Constraints

Since the focus of the research is on the optimisation strategies instead of finding a complete solution to the GTOC6 problem, an additional constraint is introduced on top of those given in the GTOC6 problem description and thereby reducing the scope of the research. The main constraint is to reduce the flyby sequence from the total problem with 128 or more flybys to a fixed subsequence of maximum five flybys.

3

CSU Solution Analysis

The GTOC6 competition finished by over more than two years ago, but still knows to challenge group of individuals to find the optimal solution that maps all four Galilean moons within the four years mission time. Champion of the official competition are the Italians from Turin Polytechnic and "Sapienza" University of Rome with a score of 311 out of 324. After the competition, the Advanced Concepts Team (ACT) of ESA that finished second, improved their results and came to an astonishing score of 316. Nearly two years later, Yang Gao from the Technology and Engineering Center for Space Utilization (CSU), Chinese Academy of Sciences announced that they surpassed the best solution by four points, totalling at 320. This is just four points short of the maximum. Their solution has not resulted in a published paper yet, but they provided the trajectory files to their solution. With the latter a detailed analysis of the trajectories to expect for the GTOC6 problem was performed to give insight for the optimisation strategies. This analysis and the results will be discussed later in this chapter.

3.1 Methodology of the CSU Solution

In contrast to an evolutionary algorithm that is used often in this problem by the other contesters, the team of CSU went for the approach of human analysis of orbital mechanics. Here two types of orbits are used to create the solution. The first is to find the transfers between different moons in the form of double-moon cycler orbits. The global mapping is performed with single-moon resonant orbits combined with back-flips (He and Gao, 2014). Although not much more information is supplied by He and Gao (2014) on their techniques and methods, a paper will be written soon with more details (according to email conversation). They announce that their solution is a new form of flight mechanics that allows for interspersed mapping with resonance. Also their moon sequence is derived from analysis of orbital mechanics. The complete sequence solution of CSU can be stated in one line as follows

SOI-1Ca-4Ga-4Ca-4Ga-4Eu-32Io-28Eu-15Ga-2Ca-4Ga-2Ca-4Ga-2Ca-4Ga-18Ca

Here SOI is the sphere of influence and is followed by the gravity assists performed by the spacecraft. The digit is the number of flybys at the same moon noted by the two first letters of said moon. Most noteworthy to mention is the fact that most faces of one particular moon are mapped in a few large sequences.

3.2 General Mission Analysis

The paper (Izzo et al., 2013) published by the ACT of ESA on their solution to the GTOC6 problem allowed for considerable insight into the complex moon system problem. First, the mission time is found to be very limited to successfully map all the Galilean moons. An important conclusion from this follows that time is costly and essential and that multiple revolutions (typical for low-thrust trajectories) are not desirable and should be avoided. This is also further illustrated by the fact that capture of the spacecraft should happen fast with as many flybys as possible. The latter will result in a high score within a small time frame.

Furthermore, thrusting is used for small adjustments of the trajectories whereas flybys are utilised for major trajectory changes. However, the amount of change due to either flyby or thrusting is not known and is explored by analysing the solution of CSU (He and Gao, 2014). Next, from the total available mass as propellant, only a small portion is reserved as propellant for thrusting whereas the majority is used for paying the penalty of close encounters with Jupiter due to radiation. The most important take here is that low-thrust propulsion is not used for changing the trajectory and that flybys should be used effectively here. These findings were also supported by Colasurdo et al. (2014).

3.3 Trajectory Analysis

The Chinese team from CSU, although not a participant of the competition, has computed the best solution so far. Note, that the solution has not been verified yet by the organiser of the GTOC6 competition, the Jet Propulsion Laboratory from NASA (according to email conversation Petropoulos said that ”the solution looks solid/valid”). The solution files of the trajectory were acquired from the Chinese CSU and underwent thorough analysis of the characteristics of the trajectories. MATLAB was used to process the data and to generate the results which are shown and discussed in this section.

To analyse the solution of the Chinese, the complete trajectory was rebuilt from their solution files. The solution contains both thrusting arcs and Keplerian arcs where the latter are propagated until the next arc or flyby to match the complete trajectory. During the reconstruction the history of the orbital elements are stored for analysis later on. The complete trajectory solution is shown in Figure 3.1.

In the figure the blue arcs represent the Keplerian arcs whereas the red arcs represent the arcs where low-thrust propulsion is used. Immediately it can be noticed that low-thrust has been used sparsely and that there are more Keplerian arcs than thrust arcs. Also the capture of the spacecraft by Jupiter is clearly visualised in the figure and shows that after several flybys the spacecraft maintains an orbit around Jupiter within the confinements of the Galilean moons. As soon as the spacecraft enters the Galilean moon system it performs two consecutive flybys within two days before making the large comeback orbit for the next flyby. This manoeuvre is efficient and necessary to be captured by Jupiter. This was shown by Gijsen (2014) who performed detailed moon flyby sequences in the Galilean moon system using the Tisserand graph, an energy-based graphical tool to find optimal sequences of gravity assist bodies. According to the author a minimum of two moon flybys is necessary to lower the energy sufficiently to be captured by Jupiter. Note that the Tisserand graph is used with the assumption of no applied thrust during transfers. This is used in this trajectory and therefore only one moon flyby could be sufficient to have the spacecraft be captured.

The results of CSU require a division into the capture and mapping phase, because these two phases are focussed on different aspects of the mission. Capture is focussed on reducing the

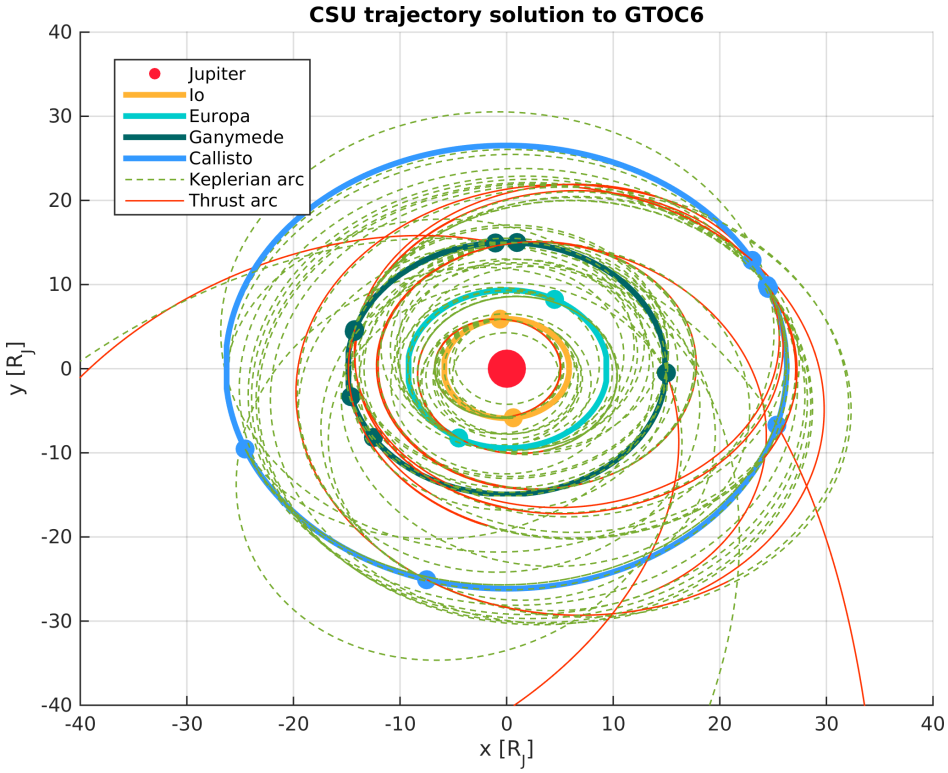

Figure 3.1: *Topview of the CSU GTOC6 trajectory solution.*

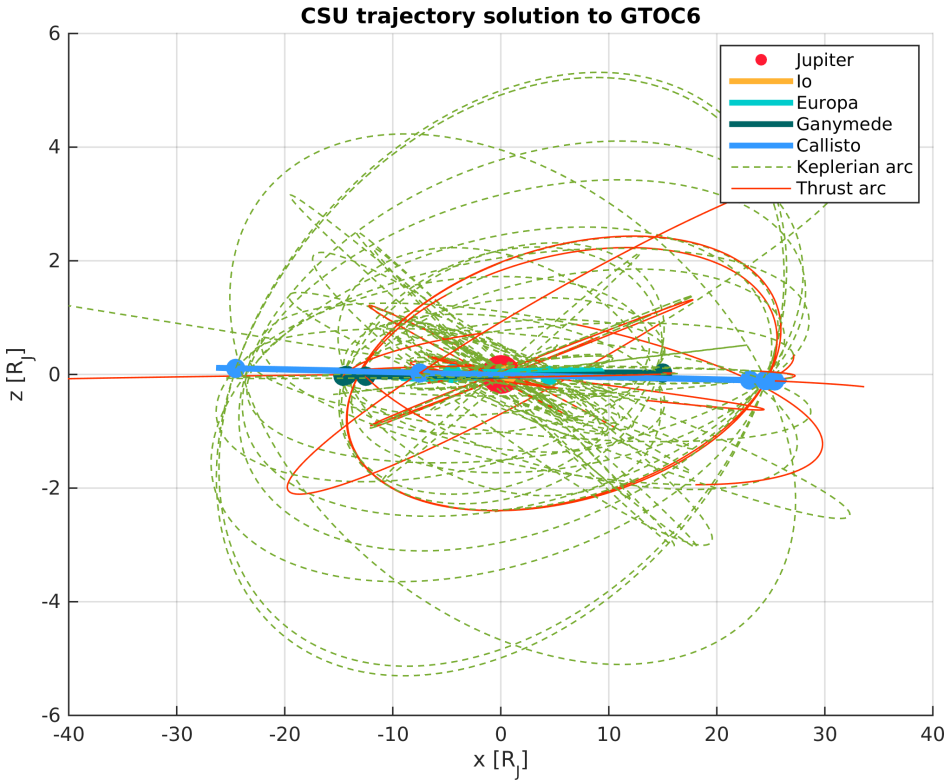

Figure 3.2: *Sideview of the CSU GTOC6 trajectory solution.*

energy of the spacecraft and thereby decreasing the semi-major axis and eccentricity. This is done with both thrusting and flybys. For this thesis the focus is on the mapping phase and therefore these first few flybys and corresponding thrusting arcs are ignored for the analysis.

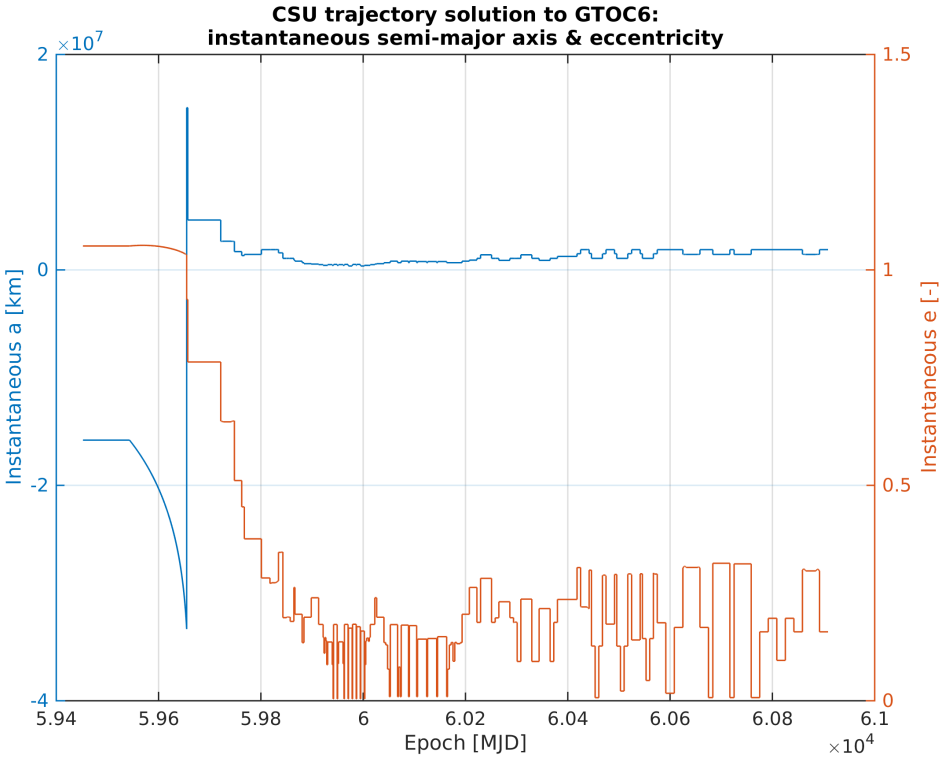

Figure 3.3: *Instantaneous semi-major axis a and eccentricty e during the complete solution of CSU.*

Figure 3.3 shows the in-plane shape changes during the trajectory legs. It clearly shows the capture phase which is characterised by lowering the energy of the spacecraft until it is captured by Jupiter which happens when the semi-major axis switches from sign and the eccentricity becomes below 1. A more detailed view of the mapping phase is given in Figure 3.4.

The results in Figures 3.3 and 3.4 show that the mapping phase contains eccentricities below the value of 0.35. Also the values for the semi-major axis stay below 30 Jupiter radii. Furthermore, the quick rapid changes are caused by a combination of a Keplerian arc followed by a flyby and again followed by another Keplerian arc and flyby and so on. The CSU solution is characterised by almost thrice as many Keplerian arcs than thrusting arcs.

Two more other orbital elements are of interest in this analysis, which are the out-of-plane parameters inclination and right ascension of the ascending node. These are displayed in Figures 3.5 and 3.6.

For both figures, again, the quick jumps can be seen as explained before. Next, it is interesting to see, that to reach almost every face on the moons requires Jovian centric trajectory legs with inclinations ranging from 0 till nearly 12 degrees inclination. Note that the orbits of the moons are nearly planar, i.e. a maximum inclination of 0.5 °. The extreme difference in the inclinations between the moon orbits and the trajectories is clearly visible in Figure 3.2. This is explained by the fact that not all faces are easily accessible. Faces near the equator of the moon and also in the vicinity of where the position vector of the moon would pierce the body, i.e. longitudes of around zero or around 180 degrees, could potentially be mapped with trajectories nearly similar to the moon. Other faces require the trajectory to be more inclined

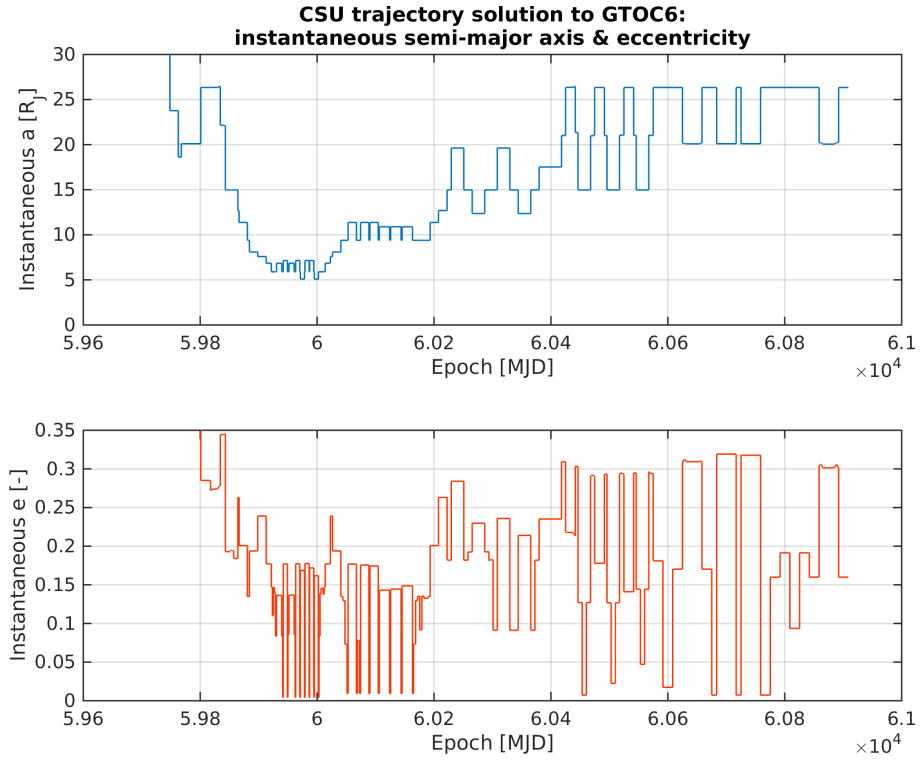

Figure 3.4: *Instantaneous semi-major axis a and eccentricty e during the mapping phase of the CSU solution.*

to map them. As one could expect, the RAAN parameter utilises the full range of 0 till 360 degrees, because the RAAN parameter of each moon is also spread over the full range.

Next, the change of each orbital parameter is analysed. Here, the orbital parameters argument of pericentre and true/mean anomaly are ignored for the time being. Note that the capture phase is ignored. Another aspect that needs to be differentiated in the results shown in Figures 3.7 - 3.14, is the difference in effects caused by the flybys and the low-thrust propulsion.

In Figures 3.7 and 3.8 the change in semi-major axis is displayed, where it becomes clear that the flybys affect the semi-major axis significantly compared to thrusting. For the extreme cases it is a factor of 1000 and for the remaining case it is still 100 times larger compared to thrusting. A similar observation can be made for the eccentricity in Figures 3.9 and 3.10 where the difference between effects caused by flybys and thrust arcs is around two orders of magnitude.

The latter two orbital parameters relate to the size and shape of the orbit. In Figures 3.11 and 3.12 the change in inclination is shown followed by the right ascension of the ascending node in Figures 3.13 and 3.14 which are defining the orientation of the orbit. Again the important notion can be made for the inclination that the flybys take care of the main change required in inclination with values in the order of a magnitude of 2 higher compared to the low-thrust counterpart.

In the results of RAAN parameter there is something else going on. On the global scale there is a magnitude of 2 difference again between the contributions of flybys and thrust arcs. However the large variations for RAAN are grouped around the absolute value of 180 degrees. This can be explained by the fact that low-inclination orbits may experience a decreasing inclination past zero degrees so that the orbit goes through reference plane. This ”negative” inclination does is not exist, because the inclination is defined as a positive value ranging from

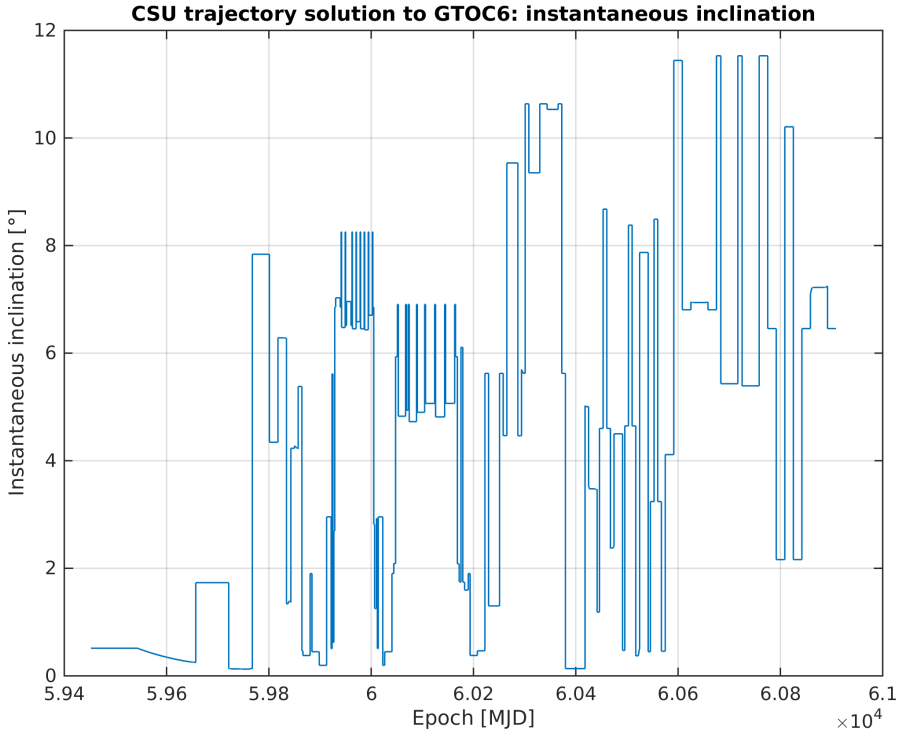

Figure 3.5: *Instantaneous inclination i during the complete solution of CSU.*

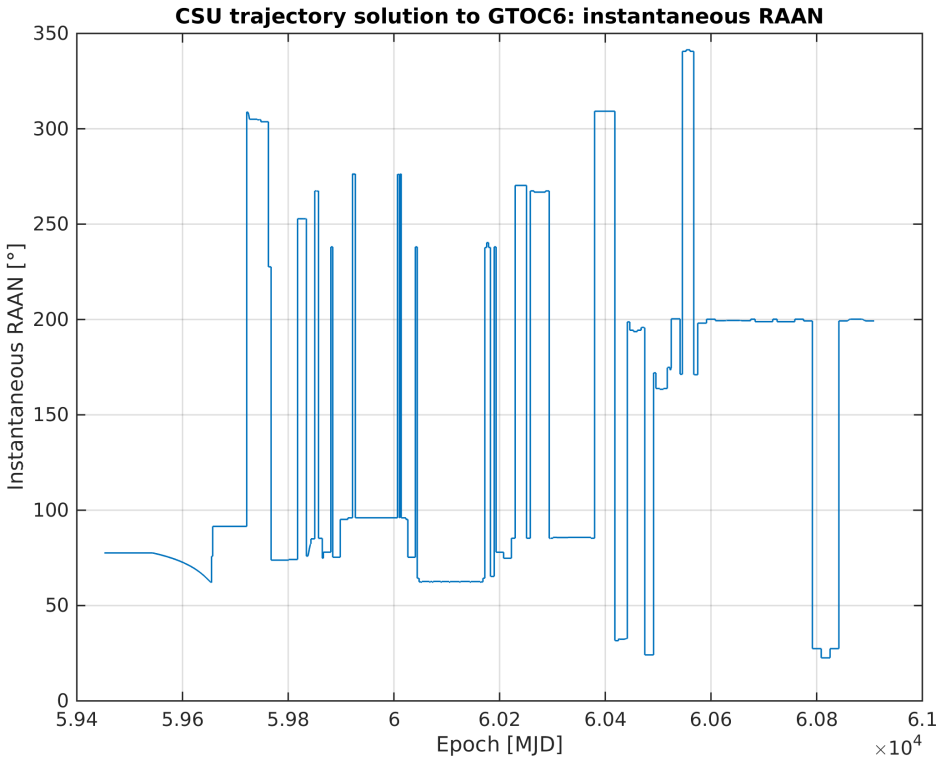

Figure 3.6: *Instantaneous RAAN Ω during the complete solution of CSU.*

0 till 180 degrees. Instead the inclination remains positive, but the orientation of the orbit is rotated a 180 degrees about the Z-axis which equals the rotation of RAAN. This rotation of 180 degrees comes back in the performance measurements of the spherical shaping method in Chapter 4 where the RAAN parameter plays an important role.

To sum up, it becomes immediately clear that the flybys are causing the largest changes which is also obvious due to the possible instantaneous changes and thereby avoiding long duration propellant consuming thrusting with the spacecraft. An important conclusion follows that the low-thrust propulsion is used for small changes and mostly in-plane. This is logic, because out-of-plane changes (inclination) require relatively considerable amounts of ΔV. In Figure 3.15 it is shown how much ΔV is required for the low-thrust arcs followed by Figure 3.16 where the control accelerations of the spacecraft required to follow the trajectory and the corresponding thrust levels are shown. The vertical lines represent the shut down and activation of the low-thrust engine, due to flybys or coasting arcs. Note that the flybys are officially not powered according to the GTOC6 definition.

More elaborate analysis of parts of the CSU solution data is required for the subject optimisation. These analyses will be made in the corresponding chapter, because of the specific nature of the data. In Table 3.1 an overview is given of the range of values for the orbital parameters that are to be expected during the mapping phase of GTOC6.

Table 3.1: *Characterstics for the orbital elements of the mapping phase of the CSU solution.*

Orbital parameter	Unit	Lower bound	Upper bound
a	$[R_J]$	5.0	27.0
e	[-]	0.0	0.35
i	[°]	0.0	12.0
Ω	[°]	0.0	360.0
ω	[°]	0.0	360.0
θ	[°]	0.0	360.0

Note that these values are not binding for the GTOC6 problem. Different methodologies can result in different trajectories to achieve the same result. For example Yam (2012) found trajectories with an inclination of nearly 23 °. Therefore these values are used with caution.

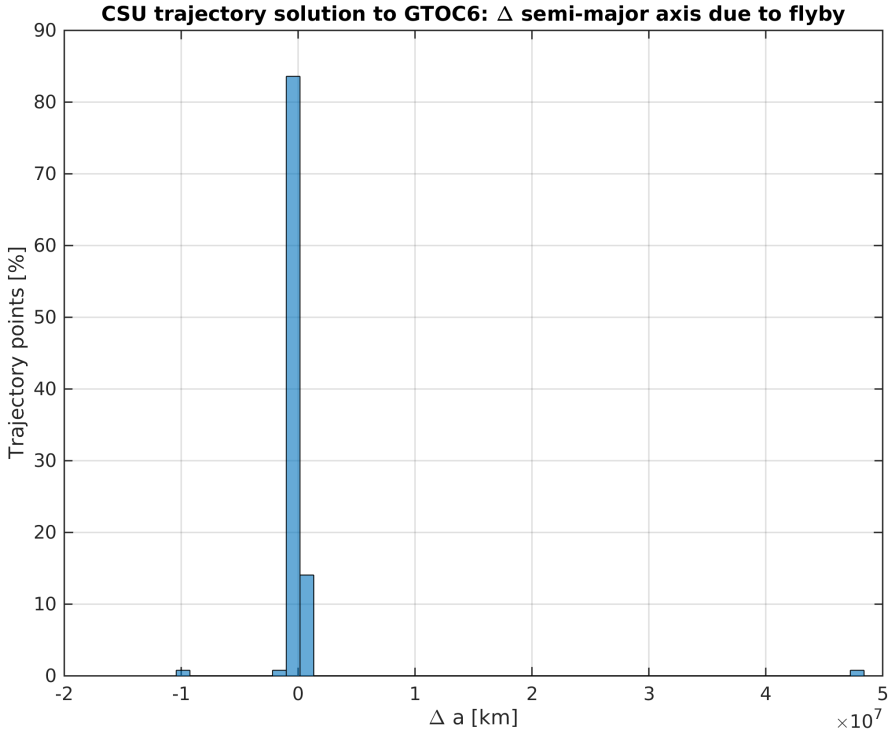

Figure 3.7: *Change in semi-major axis a due to flybys.*

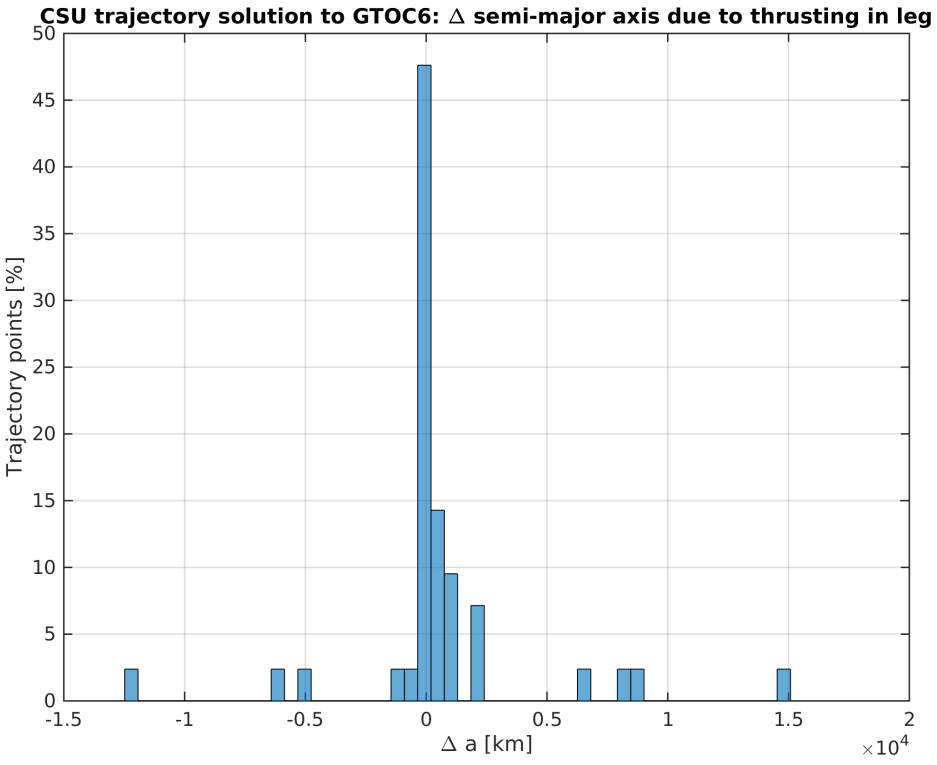

Figure 3.8: *Change in semi-major axis a due to low-thrust arcs.*

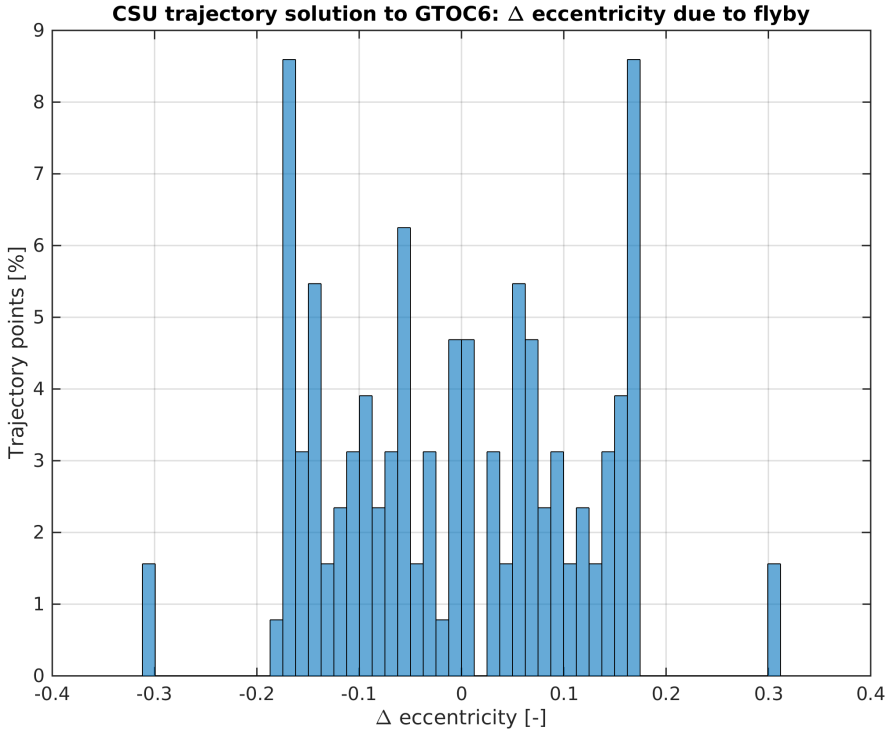

Figure 3.9: *Change in eccentricity e due to flybys.*

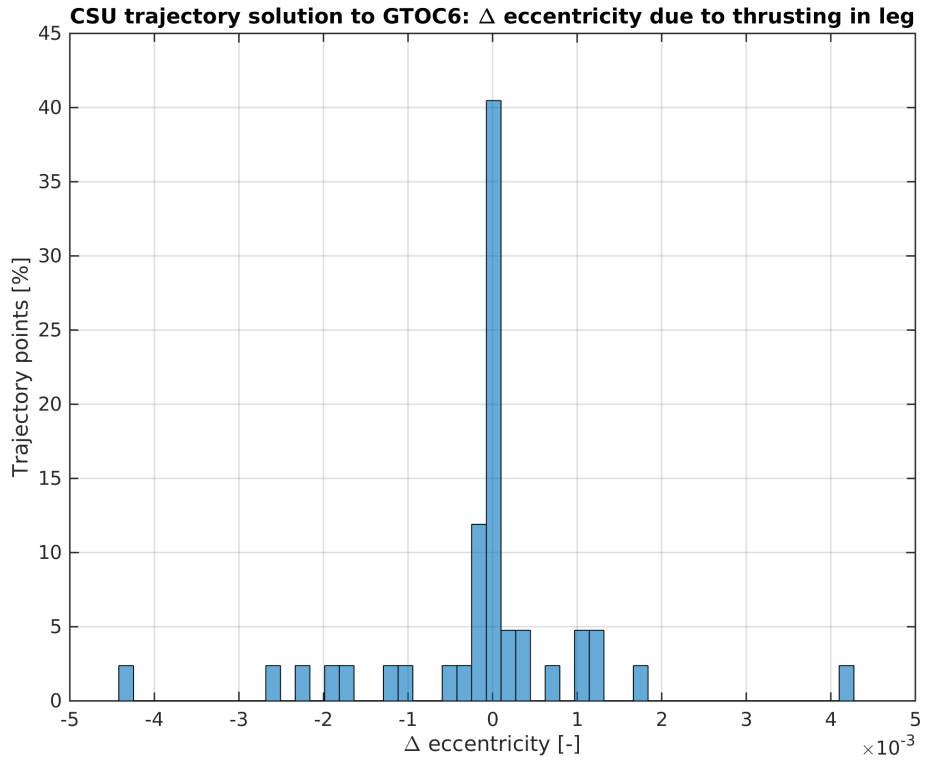

Figure 3.10: *Change in eccentricity e due to low-thrust arcs.*

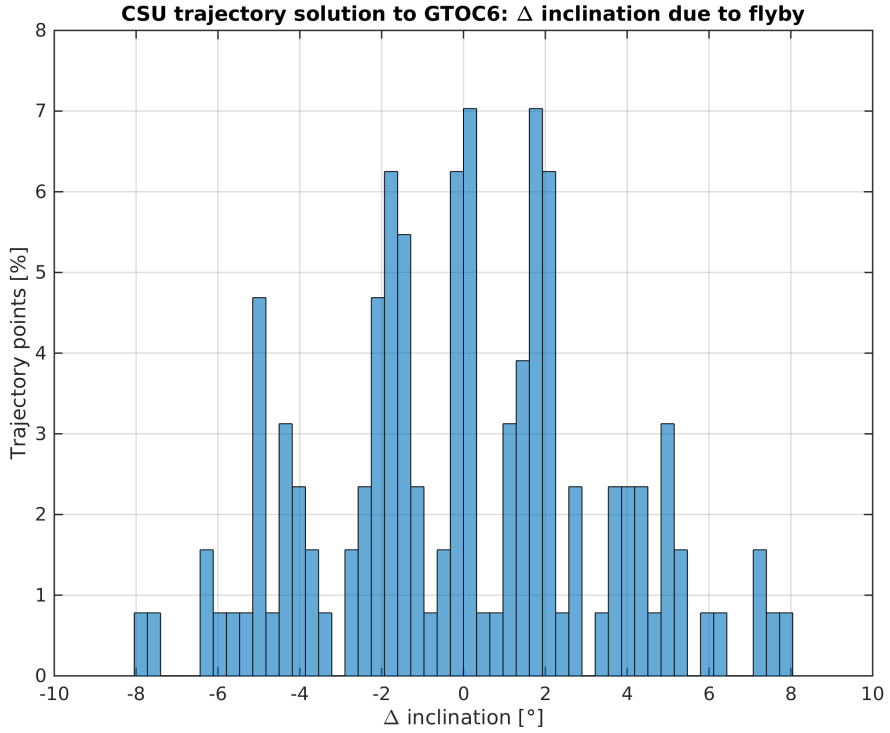

Figure 3.11: *Change in inclination i due to flybys.*

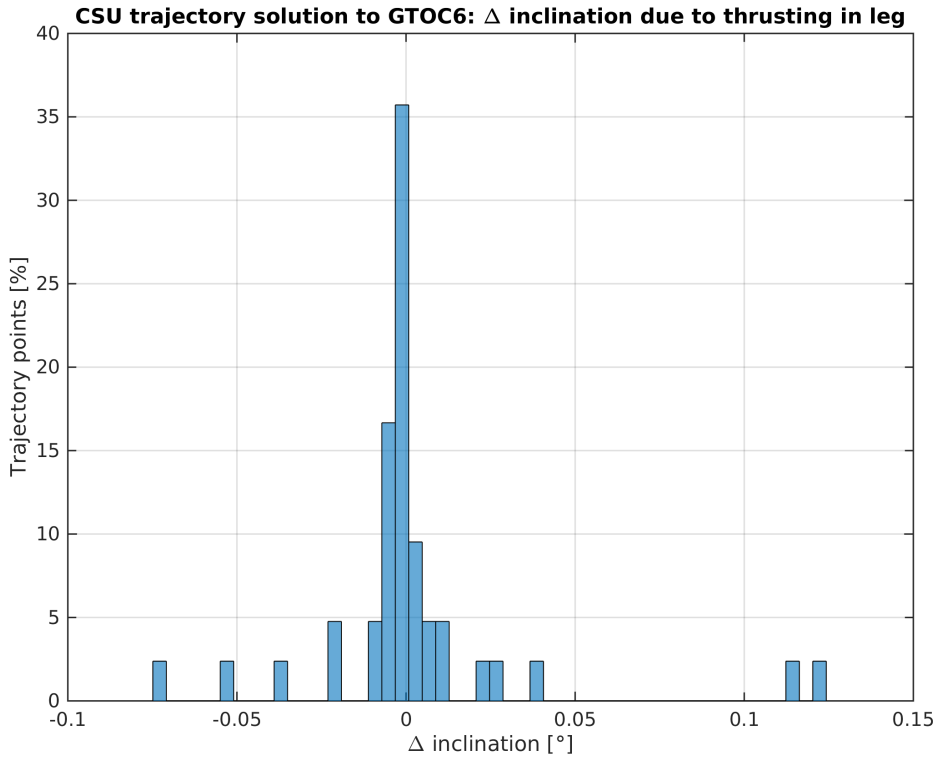

Figure 3.12: *Change in inclination i due to low-thrust arcs.*

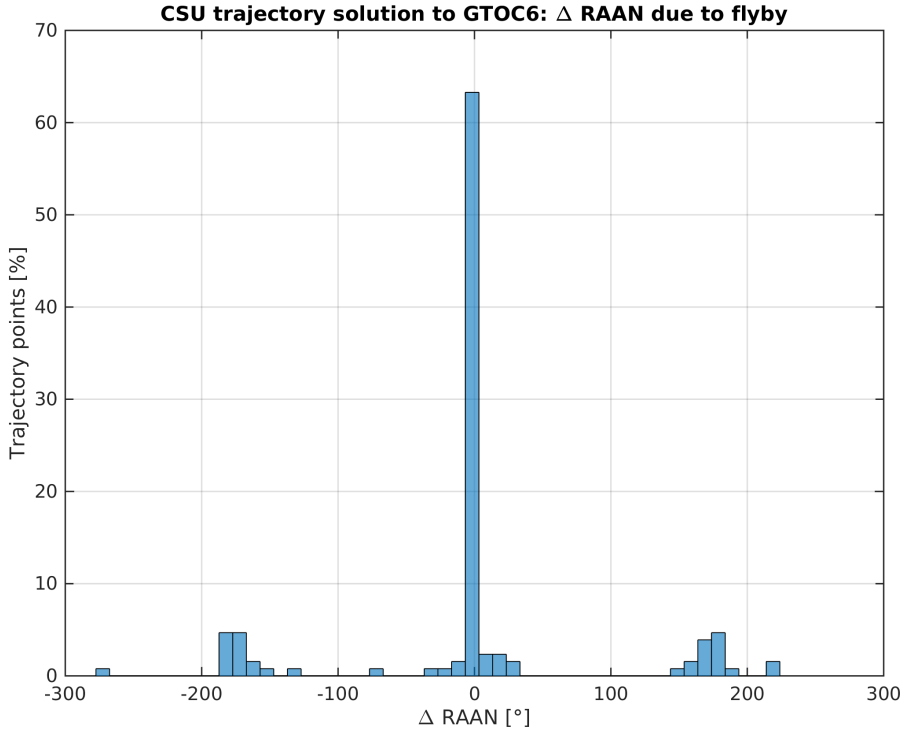

Figure 3.13: *Change in RAAN Ω due to flybys.*

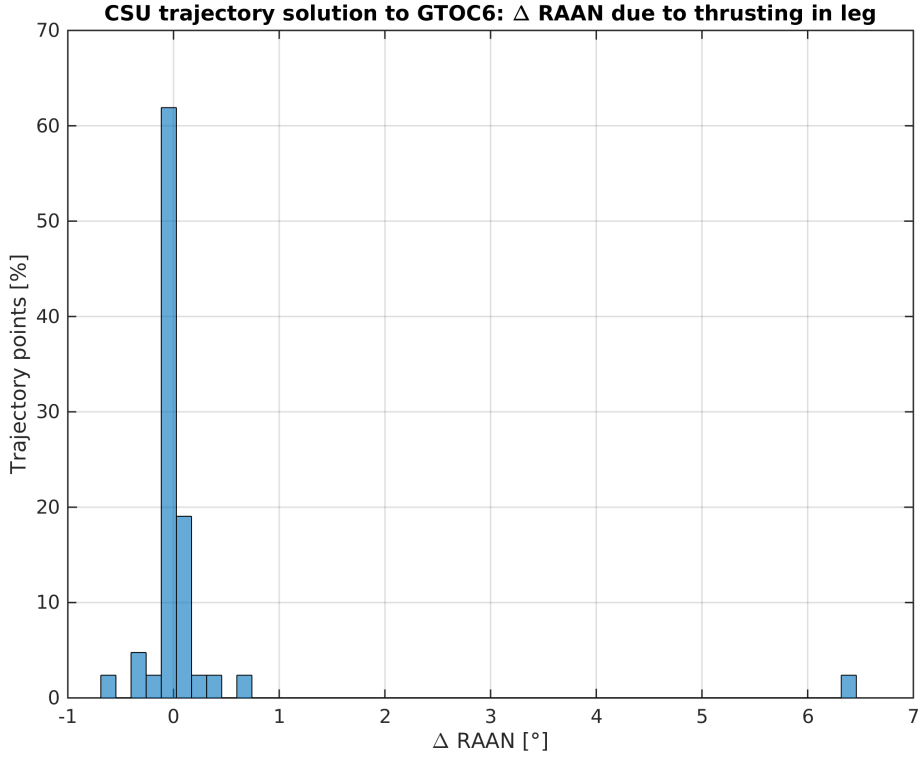

Figure 3.14: *Change in RAAN Ω due to low-thrust arcs.*

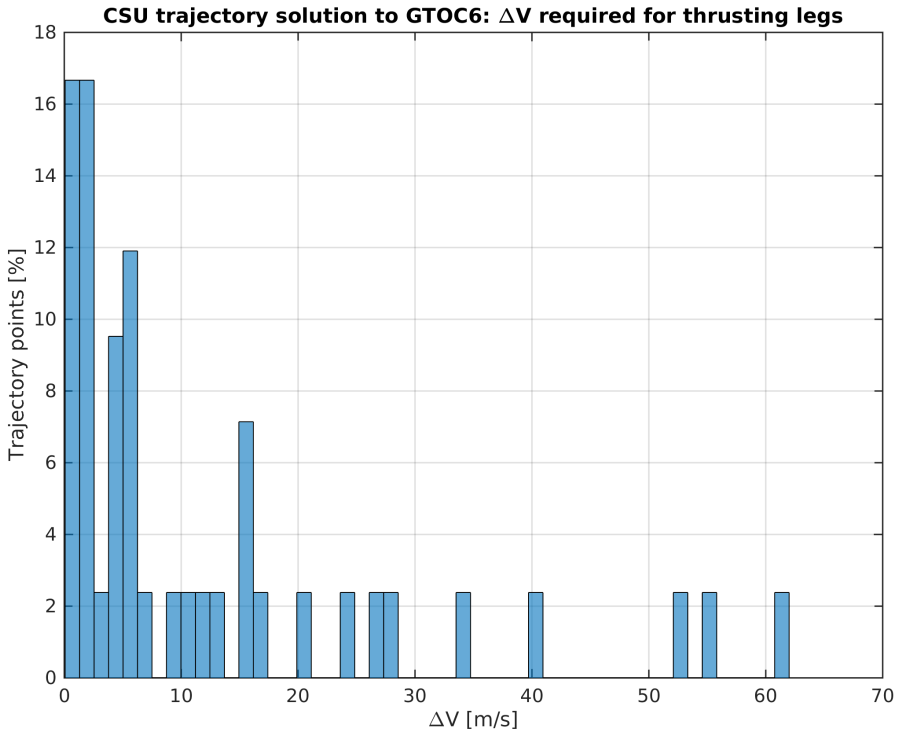

Figure 3.15: *Required ΔV for low-thrust arcs.*

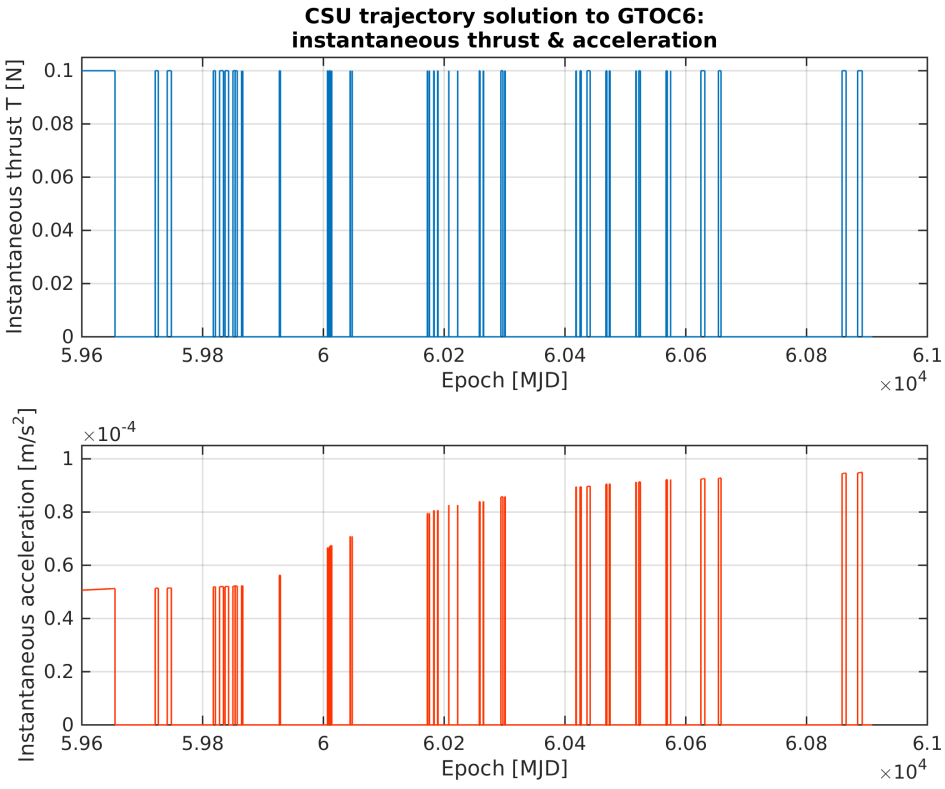

Figure 3.16: *Thrust T and acceleration a for low-thrust arcs.*

Part II

Low-Thrust Multiple Gravity-Assist Trajectory Model

4

Low-Thrust Trajectory

THE low-thrust trajectory black box consists of three major blocks. Each block is essential for creating the trajectories required for the GTOC6 problem. The first block is the low-thrust trajectory model which is discussed in this chapter. Besides low-thrust propulsion, the trajectories are influenced by flyby manoeuvres and the objective of the GTOC6 problem, global mapping of the Galilean moons. The latter two blocks are detailed in the sequential chapters.

4.1 Spherical Shaping

The spherical shaping method by Novak and Vasile (2011) is inspired by the pseudo-equinoctial elements of Vasile and Pascale (2006). Instead of using non-singular elements in the pseudo-spectral method, the spherical shaping method applies parametrisation in spherical coordinates. In this section a brief explanation of the method is given and the reader is referred to Novak and Vasile (2011); Novak (2012) and in specific Roegiers (2014) for the detailed description and analysis of the algorithm supplied in the Tudat toolbox.

The spherical coordinate system is shown in Figure 4.1.

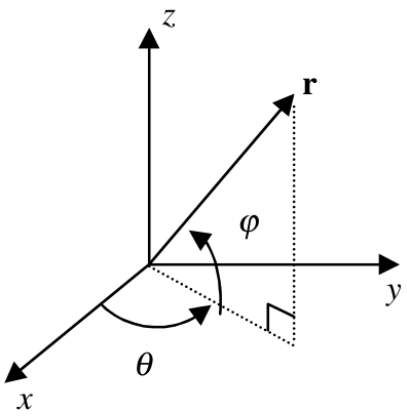

Figure 4.1: *Illustration of the spherical coordinate system. (Novak and Vasile, 2011)*

Instead of having the default spherical state vector $\boldsymbol{x} = \left[r, \theta, \phi, \dot{r}, \dot{\theta}, \dot{\phi}\right]^T$, with spherical shaping the angle θ is chosen to parametrise the trajectory. This results in three shaping functions for the radius $r = R(\theta)$, the angle $\phi = \Phi(\theta)$ and the time $t = T(\theta)$, which replaces θ. The new state vector is $\boldsymbol{x} = [r, t, \phi, r', t', \phi']^T$. Here the prime $'$ stands for the derivative with respect to the parameter θ. The corresponding equation of motion in spherical coordinates is given in Equation (4.1).

$$\dot{\theta}^2 \frac{\mathrm{d}^2 \boldsymbol{r}}{\mathrm{d}\theta^2} + \ddot{\theta} \frac{\mathrm{d}\boldsymbol{r}}{\theta} = -\mu \frac{\boldsymbol{r}}{r^3} + \boldsymbol{u} \tag{4.1}$$

$$\boldsymbol{r} = [r\cos\theta\cos\phi,\ r\sin\theta\cos\phi,\ r\sin\phi]^T \tag{4.2}$$

Here the radius vector $\boldsymbol{r}$ is defined in Equation (4.2) and $\dot{\theta} = \frac{1}{t'}$ and $\ddot{\theta} = -\frac{t''}{t'^3}$. Furthermore, $\boldsymbol{u}$ is the control acceleration vector.

The functions $R(\theta)$ and $\Phi(\theta)$ define the geometry of the shape whereas the function $T(\theta)$ defines the time evolution along the trajectory. Novak and Vasile (2011) derived the following relation for the time function in Equation (4.3) resulting in reasonable thrust profiles. Other arbitrary functions resulted in thrust profiles that could not be considered low-thrust.

$$T' = \sqrt{\frac{DR^2}{\mu}} \tag{4.3}$$

The expression for D is given in Equation (4.4).

$$D = -R'' + 2\frac{R'^2}{R} + R'\phi' \frac{\phi'' - \sin\phi\cos\phi}{\phi'^2 + \cos^2\phi} + R\left(\phi'^2 + \cos^2\phi\right) \tag{4.4}$$

The radius R and elevation angle ϕ functions can be chosen arbitrarily, however, they do need to satisfy the boundary conditions analytically.

The number of boundary conditions is determined by the initial and final state vector $\boldsymbol{x} = [r, t, \phi, r', t', \psi']^T$ giving a total of 12 BCs. Note that the spherical shaping method normalises the boundary conditions before solving the trajectory and afterwards denormalises the values back again. Two position boundaries ($T(\theta)$) can be satisfied by selecting the appropriate values for θ for initial and final position. This leaves one with 10 boundary conditions that need to be satisfied by the shaping function R and Φ. In case the 10 are satisfied, the time-of-flight is uniquely defined.

Novak and Vasile (2011) decided to use oscillating functions for the shaping functions R and Φ which are given in Equations (4.5) and (4.6). Their motivation for these functions is that the minimum-thrust arc is the Keplerian arc.

$$R(\theta) = \frac{1}{a_0 + a_1\theta + a_2\theta^2 + (a_3 + a_4\theta)\cos\theta + (a_5 + a_6\theta)\sin\theta} \tag{4.5}$$

$$\Phi(\theta) = (b_0 + b_1\theta)\cos\theta + (b_2 + b_3\theta)\sin\theta \tag{4.6}$$

Note that 11 parameters are used instead of the 10 required to satisfy the boundary conditions. The additional parameter allows for a degree of freedom for shaping the trajectory which is used to satisfy the time-of-flight constraint. The latter can not be solved analytically and therefore a numerical iteration (root-finding) is performed. In this form the method does not have free parameters and therefore has limited flexibility for optimisation.

The spherical-shaping method is able to cope with the rendezvous problem and can therefore match both position as well as velocity. Also three-dimensional trajectories and multiple

revolutions can be handled. Thrust limit on the other hand, is done by applying constraints afterwards and not during the shaping. This is a significant disadvantage, because the thrust has to be computed after which it can be determined if the trajectory violates the maximum thrust. It is preferable to set the maximum thrust level a priori.

4.2 Limitations to the Tudat Implementation

Besides the general limitations of the spherical shaping method noted in the previous section, Roegiers (2014) found several additional limitations in her practical application of the method worked out for the Tudat software library.

First of all, the inclination of the implementation is limited to several tens of degrees due to the assumed elevation function in Equation (4.6) to shape the third dimension. In this chapter an elaborate analysis is performed of the limitations of the performance of the spherical shaping method, to get a thorough picture of the orbits and trajectories that the algorithm is able to produce without significant errors in orbital states and other parameters like ΔV. This is important for the actual trajectory computations in the GTOC6 problem, where the flybys change the characteristics of the current trajectory instantly.

According to Roegiers (2014), the limitation of the elevation function also has its influence on the allowable RAAN values for the departure and arrival conditions of the rendezvous problem. Besides that, identical in-plane orbit transfers at different inclinations were giving different ΔV values. This was solved with a transformation fix by Roegiers (2014), however, this did not improve the performance with respect to the RAAN parameter. Also this will be treated in this chapter.

On top of the limitations the algorithm was only analytically verified. The spherical shaping method computes the trajectories by solving the two-point boundary value problem and therefore needs to solve several intermediate shaping equations. In cases of extreme and exotic trajectories, one or more of these equations will not result in a solution (not solvable) and therefore the trajectory is invalid. To verify the latter a number of control points along the shaped trajectory are computed. Each point is checked to verify that the solution does not violate the assumptions and that the shaping functions do not result in significant errors.

As an example, for a highly-eccentric inclined orbit the shaping method has trouble to find a suitable trajectory. The orbital states of the control points deviate from the desired trajectory. The errors accumulate resulting in a final orbital state that does not match the given arrival point of the two-point boundary value problem. Analytical verification is in place to assure that the solutions are valid and that the shaped trajectories reach their intended destination. As already mentioned, the analytical verification is available, though numerical verification is not. The latter is discussed in Section 4.4.

The maximum error for the position and velocity values at the destination are set to respectively 1 m and 1×10^{-6} m/s per AU. A position error of 1 m is considered acceptable in case of analytical verification. This means that the shaped trajectory is allowed to deviate a maximum of 1 m of the intended arrival position. A similar analogy applies to the velocity error and the arrival velocity. Significant position and velocity errors in the analytical solution indicate that the trajectory is not suitable for this implementation of the spherical shaping method in Tudat. Therefore the more strict the position and velocity errors are, the more reliable the solutions are that are made by the shaping method. On top of this the numerical integration will introduce another error on top of the analytical inaccuracy. The latter will be discussed in Section 4.4.

4.3 Additional Components to the Tudat Implementation

Before the Tudat implementation could be used to its full extent for the GTOC6 problem, there were a few new features and error handling that were added to the implementation.

It was mentioned in the previous section that the analytical verification could potentially result in errors along the control points. For highly-eccentric inclined orbits the time shaping function D in Equation (4.4) would result in negative values at one or more control points along the trajectory. Negative values indicate that the local curvature of the trajectory is curved away from the central body which is not allowed. Beside negative values, the algorithm available in Tudat produced NaN (not a number, undefined) values as well. However, these were not filtered by the method, allowing for incorrect solutions, marked as valid by the algorithm. Therefore an additional check was added for these undefined values in the time equation.

Although the implementation verified the solutions analytically, this did not mean that the solutions were viable or practical. Therefore verification by numerical integration was added to the implementation as well. In the research before this thesis two integration methods were suggested to be tested (Hoving, 2014). The best option would be used for the complete problem. The testing of the numerical integration is done in the next section.

Finally, the shaping algorithm in Tudat did not contain computations for mass and thrust force during the trajectory for a user-defined electrical propulsion unit. This is needed for determining thrust violations during trajectories, which were done after the trajectory was computed, again at the control points. This was also added to the existing Tudat algorithm.

4.4 Verification of Trajectory

For the verification of the spherical shaped trajectories a proper integrator needed to be selected. From the preceding literature research two suitable integrators were selected, namely

- *RK7(8)* - The embedded Runge-Kutta-Fehlberg method utilising a 7^{th}-order integration method and an 8^{th}-order error estimation method.
- *DOPRI8* - Also known as RK8(7) which is an embedded Runge-Kutta by Prince & Dormand with an 8^{th}-order integration method and a 7^{th}-order error estimation method.

Also the workhorse *RK4*, a 4^{th}-order Runge-Kutta method, was used. Using the *RK4* with its fixed step-size allowed for direct comparison of the control points that were computed analytically with the spherical shaping method. In this way, both the accelerations and the orbital states at all intermediate points between the arrival and departure state could be traced and verified. Also debugging the propagation code was very convenient in this way and allowed for identifying and solving several problems that occurred while computing the control accelerations of the spacecraft required for each integration step. Although in the literature study (Hoving, 2014) it was assumed that the *RK4* would perform insufficiently compared to the variable step-size integrators, it was included to avoid neglecting an integration method that uses a relatively low amount of computational effort per integration step. However comparing with the performance and the quality of the results of the other integrators the *RK4* was no match in terms of number of steps (considerable small step-sizes for the complete trajectory), computation time, and position and velocity errors.

Variation of the orbital parameters (Modified Equinoctial Elements, Gauss' form of planetary equations) was selected in the literature research to do the propagation of the trajectory. However, also Cowell's method (direct integration of the acceleration in Cartesian coordinates)

was used to verify the performance and the quality of the variation of orbital parameters method. Again the fixed step-size *RK4* was used to verify the MEE elements computed with variation of orbital parameters.

The spherical shaping method is parametrised with the azimuthal angle θ and therefore the independent variable for the integration is this azimuthal angle instead of the usual parameter time. For time as independent parameter a conversion needs to be performed, which can only be computed after the shape has been determined, since only after shaping the azimuthal angle its derivative with time is known. This derivative needs to be integrated to come to the time variable needed. This is a tedious and time consuming process if a certain time step is to be expected and therefore time is not selected as a suitable independent variable. Using time as an independent variable is considered an item on the future work list, discussed in Chapter 11.

In Table 4.1 the results are shown for an arbitrary Earth-planet X transfer. Here the number of revolutions is set to five. The table shows the final orbital state when arriving at planet X for the analytical computation with the spherical shaping method and the two integrators. Also both propagation methods have been shown, where the top shows Cowell's method with Cartesian coordinates and below is the variation of orbital elements using MEE. A few things can be noticed clearly from the table.

Table 4.1: *Results of integrators for propagation using direct integration of Cartesian coordinates (top) and integration of Lagrange's planetary equations with MEE (bottom). Absolute and relative tolerances are set to* 1×10^{-14}.

Description	Unit	Analytical	RKF7(8)	DOPRI8(7)
x	[m]	208039903233.6481	208039903233.6726	208039903233.6425
y	[m]	-2090471735.325027	-2090471735.316544	-2090471735.3747
z	[m]	-5174612856.376161	-5174612856.376186	-5174612856.376189
$\dot{x}$	[m/s]	1173.498351890574	1173.498351890982	1173.498351890466
$\dot{y}$	[m/s]	26296.66247574401	26296.66247574425	26296.662475744
$\dot{z}$	[m/s]	520.8174308493636	520.8174308493649	520.8174308493742
N_{steps}	[-]		270	444
t_{cpu}	[s]		0.064026	0.102812
$\lvert\boldsymbol{x}_{error}\rvert$	[m]		0.0259035763240423	0.0499855722120127
$\lvert\boldsymbol{v}_{error}\rvert$	[m/s]		4.696817155095998e-10	1.098095392332208e-10
p	[m]	225957136071.9289	225957136071.9327	225957136071.9297
f	[-]	0.08534789825483075	0.08534789825477763	0.08534789825482669
g	[-]	-0.03785208090442532	-0.0378520809044809	-0.03785208090442944
h	[-]	0.010450298503653	0.01045029850365433	0.01045029850365347
k	[-]	0.01232832999533678	0.0123283299953379	0.01232832999533641
L	[rad]	6.272880482712712	37.68880701861052	37.68880701861048
N_{steps}	[-]		184	255
t_{cpu}	[s]		0.042946	0.059225
$\lvert\boldsymbol{x}_{error}\rvert$	[m]		0.02934584202593212	0.03531411073146219
$\lvert\boldsymbol{v}_{error}\rvert$	[m/s]		4.679904566463769e-09	4.214644110182441e-09

[1] Analytical refers to the control points computed by the spherical shaping method.
[2] Orbital state at the arrival position.
[3] Propagated trajectory is an Earth-planet X trajectory with 5 revolutions at epoch is 51544.5 MJD.

First, the *RKF7(8)* shows to be favourable with the number of steps. This also translates to the computation time (less function calls) which is considerably less for *RKF7(8)* compared to *DOPRI8*. On the absolute position error, again *RKF7(8)* is clearly more suitable for the job. For the velocity on the other hand the roles are switched. Overall, the performance of the *RKF7(8)* is dominant and especially the computation time plays and important role for the given accuracy in the GTOC6 problem. The same holds for propagation using Cartesian coordinates or MEE. MEE shows better performance in terms of speed and accuracy (except

velocity). Note, that only one test case is shown here, but that these results are consistent for many different test cases that were performed. These test cases consisted of different orbit transfers with varying values for all orbital elements.

For numerical verification in this chapter the same position and velocity errors apply as for the analytical verification. In other words, a maximum error for the position and velocity values at the destination are set to respectively 1 m and 1×10^{-6} m/s per AU. Note that these errors can be loosened by the constraints of the given problem. For now the shaping method in general is tested and therefore these strict maximum errors are used.

The variable step-size integrators can be set with predefined absolute and relative tolerances. For tolerances of 1×10^{-13} (both absolute and relative) and lower the required maximum position and velocity error can be achieved for the *DOPRI8* integrator. The *RK7(8)* requires a tolerance of 1×10^{-14} to achieve similar results, though, it requires slightly less computational effort. In the results of Table 4.1 the same tolerance of 1×10^{-14} is used to show the difference in performance. However, the performance is similar for both integrators when using a tolerance of 1×10^{-13} for *DOPRI8*. Therefore the results in Table 4.1 are used to determine the integrator. This concludes the *RKF7(8)* integrator with the variation of orbital elements as propagation method for the remainder of the thesis.

4.5 Testing and Analysis of the Spherical Shaping method

In this section the spherical shaping method is benchmarked. With the benchmarking the performance and the limitations of the method are found that could give indications of potential problems during the next phase of the GTOC6 problem, the optimisation of the trajectories. All transfer trajectories which are considered here are rendezvous trajectories. An important remark for all simulations in this section is that the time-of-flight is left free for the spherical shaping method to solve. A fixed time-of-flight would result in complete different transfers for different lengths of trajectories. Also accuracy of the trajectories would be influenced by the constant time-of-flight. Therefore the time-of-flight parameter is left free.

4.5.1 Curvature Verification

The spherical shaping method makes use of analytical verification of the computed shaping functions. This is done by taking control points along the trajectory and calculate various parameters at these points. Computing these values is a time-consuming process and by reducing the number of control points the computation time can be decreased significantly.

One particular and important parameter value is computed with the shaped time function D in Equation (4.4). Novak (2012) used this simplified function to shape the time to avoid complexity and time-consuming computations. The assumption for the simplified function was that in-plane the spacecraft can only thrust tangentially. Normal thrust is ignored. In order to satisfy this assumption, the function D is not allowed to be negative anywhere along the trajectory. Therefore only calculating the function values of D at the initial and final control point of the trajectory is not sufficient. There must be a number of control points to verify that the assumption is not violated Novak (2012); Roegiers (2014). A number of 20 control points was deemed a proper compromise between being effective, fast and thoroughly verifying the trajectory up till a satisfying level by Roegiers (2014). Note that this number is per revolution and therefore a constant interval in the independent variable θ is used to verify the trajectories, also for increasing numbers of revolutions. A variation of 50 and 100 control

points was performed to test the performance compared to the required computational effort. It turned out that the number of control points is linearly proportional to the computational effort. Only a few edge cases, which were valid with 20 control points, were considered invalid after increasing the number of control points. However, the computational effort weighed considerably in the decision to keep the number of 20 control points.

4.5.2 Limitations of various transfer trajectories

According to Roegiers (2014) the shaping method has trouble with trajectories that go past inclinations of 50 °. Though several test cases were performed and with the comet Tempel 1 transfer being the extreme case with an inclination of 10.527 ° and eccentricity of 0.517, a thorough test for the full range of values for each orbital element was not performed. To get a picture of the limitations, a full test case was performed.

The test case concerns a transfer between Earth and a fictional planet X transfer, with Earth and planet X at respectively 1 AU and 3 AU. The initial orbit of Earth is circular, has no inclination and its departure position in the orbit is fixed during the test case. Also the other orbital parameters are set to zero degrees. The destination orbit of planet X is changed to reflect different geometries of the problem throughout the whole test case; only the semi-major axis is kept constant. The parameters for planet X are not continuous and are given within a range of values with defined step-sizes which are found in Table 4.2. Note that there are two different step-sizes for the last three orbital parameters; the smaller step-size was used to refine the test case after having obtained the initial results.

Invalid trajectories for this chapter are defined as analytically and numerically not verified. To pass analytical verification the position and velocity values at the destination had a maximum error of respectively 1 m and 1×10^{-6} m/s per AU. To clarify, the shaping method solves the set of coefficients to shape the most optimal trajectory for the given boundary conditions (departure and arrival states, time-of-flight and number of revolutions). However, for increasing eccentricity and inclination (limited elevation function) the algorithm has more trouble with shaping the trajectory. The trajectory starts deviating from the intended destination orbital state. Due to accumulating errors along the trajectory the final position and velocity differ from the intended arrival orbital state. For this reason, a maximum error is set on both the position and velocity at the final boundary condition. Note that at this stage determining the validity of the trajectories is only based on the position and velocity errors and whether the algorithm can solve the coefficients to shape the trajectory (e.g. positive D function values). The ΔV is considered in the next section. Note, the error value per AU is to scale the maximum allowable error with distance (relative error).

Numerical verification is used to check if the trajectory is valid for a spacecraft to follow. The same error is applied to the position whereas the velocity was set to 1×10^{-3} m/s per AU. The latter is used to have the same ratio between position and velocity errors as is with the propagation constraints of GTOC6. Note that the numerical position and velocity errors are also defined with respect to the intended destination and not with the actual destination of the shaped trajectory. This is because one wants to determine if the trajectory is able to put the spacecraft at the intended destination point. Therefore the analytical boundary error is kept significantly small.

In Figure 4.2 the results of the full test case are shown. A grand total of N_G =193,536 transfer trajectories have been computed from which 59,898 (30.95 %) are found to be invalid. The invalid (both analytical and numerical) trajectories consider the full range of values for all orbital parameters and the figure shows the inclination for which invalid trajectories occur. It is clear that when all varieties of transfers are considered, that the spherical shaping method

Table 4.2: *Initial and final orbit parameters for determining the limitations of the shaping method.*

Orbital parameter	Unit	Orbit Earth	Planet X orbit		
			Lower bound	Upper bound	Step-size
a	[AU]	1.0	3.0	-	-
e	[-]	0.0	0.0	0.95	0.05
i	[°]	0.0	0.0	85.0	5.0
Ω	[°]	0.0	0.0	360.0	22.5 / 45.0
ω	[°]	0.0	0.0	360.0	22.5 / 45.0
θ	[°]	0.0	0.0	360.0	22.5 / 45.0

produces invalid trajectories for all inclinations. Note that the invalid solutions with low inclination are due to the high eccentricity transfers of 0.7 and more.

Filtering out the high eccentricity values creates a more clean result which is displayed in Figure 4.3. Here eccentricities above 0.45 are removed from the search, leaving a selection of the total of $N_S = 92{,}160$ trajectories, and this shows that the spherical shaping method is able to make transfers up till 45 degrees inclination (23.04 %, or 10.97 % invalid w.r.t. N_G). After that, the number of invalid trajectories increases relatively fast. Both figures confirm the results from Roegiers (2014) that trajectories above 50 degrees inclination are invalid even for reasonable orbital element parameters. To be precise, this translates to the fact that spherical shaping (Tudat's implementation) is not suitable for trajectories with an inclination of 45.0 ĉirc and higher.

The final test is to find the values for the remaining orbital parameters for which the trajectories start to fail ($e = 0.45$ and $i = 45\,^\circ$). To get a more clear picture for the parameters Ω, ω and θ, the smaller step-sizes in Table 4.2 are used for said parameters. Also inclinations above 50 degrees are neglected. This resulted in a grand total of N_{GS} =495,616 transfer trajectories from which 3,142 (0.63 %) are found to be invalid. For the edge case with $e = 0.45$ and $i = 45\,^\circ$, a total of 4,096 selected trajectories, only 12 (0.29 %, or 0.0024 % invalid w.r.t. N_{GS}) are invalid solutions. In Figure 4.4 the relaxations with the parameters Ω, ω and θ are given.

Two observations can be made from these results. First, problem occur when the true anomaly parameter is around zero degrees, meaning that the shaping method starts having trouble when arriving at the pericentre of the arrival orbit. The destination pericentre comes closer to the initial orbit for increasing eccentricity (in this case where the initial orbit is smaller). On top of that, large inclinations require a considerable change in the orientation of the velocity vectors compared to the departure velocity orientation in the initial orbit. Secondly, both RAAN and the argument of the pericentre may lead to problems when their values are around either zero or 180 degrees. Note that the set of the three parameters that belong to the same invalid trajectory can not be read from the figure. For θ this is not of influence, however, the combinations of Ω and ω are appear to happen around the same values. To clarify, if Ω is around 180 degrees than so is ω, the same holds for around zero. The latter leads to the conclusion that the shaping method may lead to invalid trajectories when the arrival position is relatively close to the departure position for considerable large eccentricities and inclinations in the arrival orbit.

The final conclusion from these test results is that the shaping method has no problem with describing transfer trajectories, as long as the eccentricity and the inclination stay below respectively 0.45 and 45 °. An important remark that comes with this conclusion is, that the initial orbit is circular and not inclined. Also the remaining orbital parameters did not vary during the test, which could be looked into for further research into the spherical shaping method. And as final note, the results of this section are performed without the reference

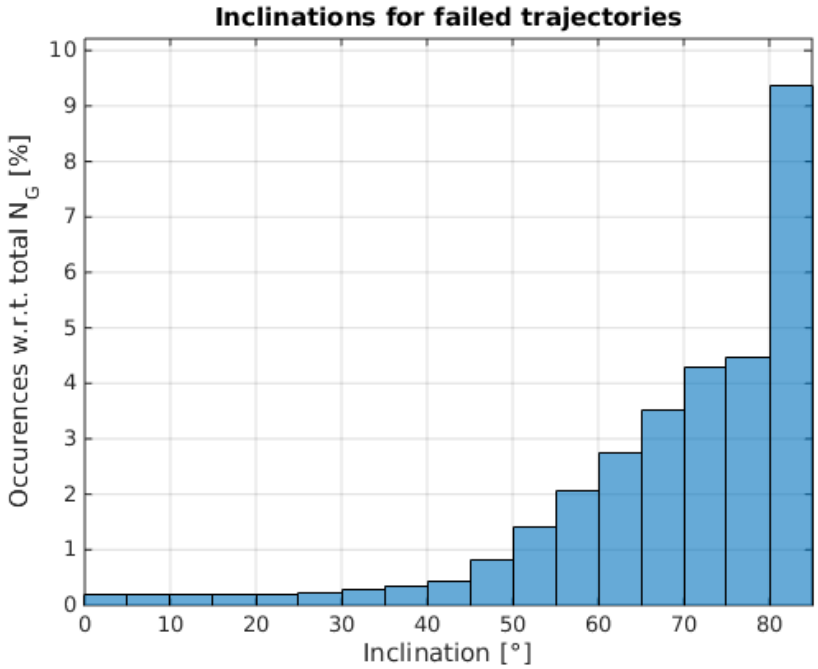

Figure 4.2: *Inclination for invalid trajectories for the whole range of orbital parameters.*

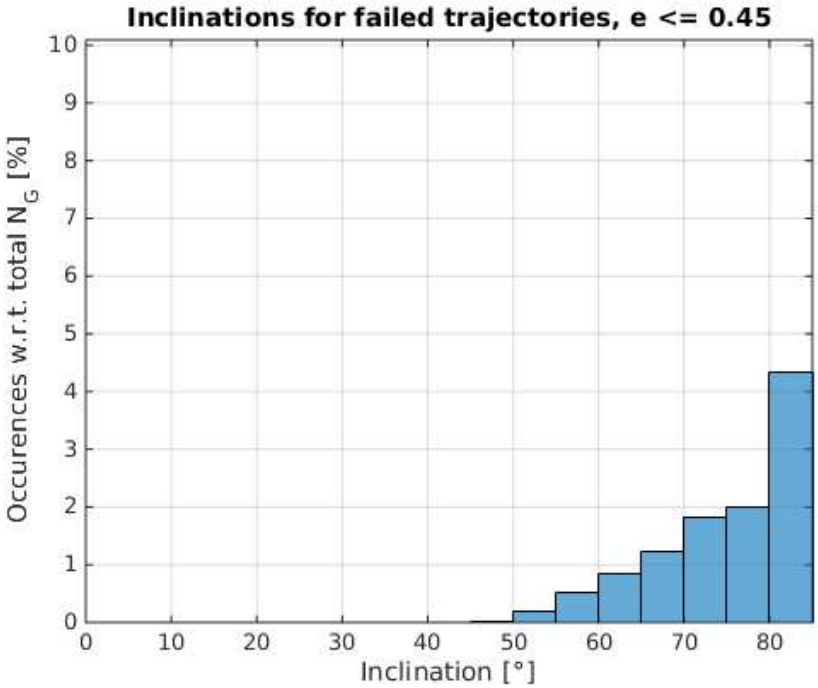

Figure 4.3: *Inclination for invalid trajectories where the eccentricity is less or equal to 0.45.*

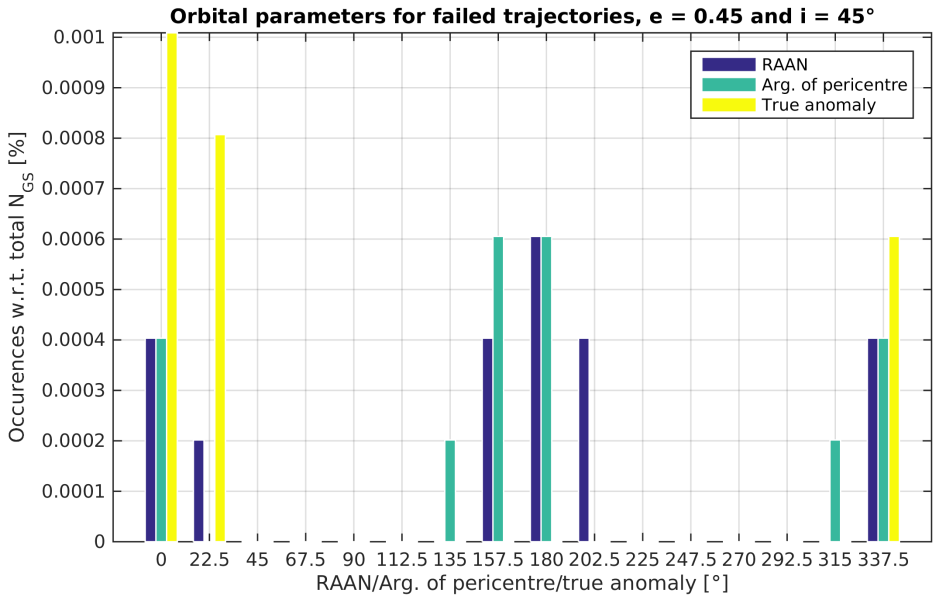

Figure 4.4: *Orbital parameters for first invalid trajectories where the eccentricity equals 0.45 and inclination* $45\,°$.

transformation fix suggested by Roegiers (2014) to solve the ΔV issue with in-plane transfers at different inclinations. Though the fix was not used for this test, it was found that the fix did not alter the results significantly.

4.5.3 Transfer Accuracy of Shaping Method

One problem during the development of the spherical shaping method for Tudat that Roegiers (2014) encountered, was the deviating ΔV values for the same in-plane transfer at different inclinations. These should result in the same values and for that reason she came up with a reference frame transformation (RFT) fix that solved the problem caused by the elevation shape function with the assumption of low inclinations. Also the Keplerian arc was made possible with the fix. The fix involved rotating the initial and final orbit over the inclination angle i, followed by a rotation of the RAAN angle Ω. After solving the coefficients for the shaped trajectory, the reverse of the rotations was applied to get back to the original situation. Though the in-plane transfers were corrected, it did introduce a new deviation in ΔV calculations in case of different RAAN parameters for the departure and arrival orbit. According to Roegiers (2014) for an increasing difference in initial and final RAAN the error in ΔV would increase considerably. In this section a thorough test is performed for both versions of the algorithm to determine the efficiency and most suitable algorithm for the GTOC6 problem.

The tests performed are again computing various transfer orbits, similar to the test in the previous subsection. However, now both departure and arrival orbit are set to in-plane transfers, and thus have the same inclination. In case of the test without the RFT fix, increasing values for ΔV are expected while increasing the inclination of the same in-plane transfer. Here the solution to the zero degrees inclination is used as a reference and to determine the correctness of the transfer in an inclined plane. To be able to determine the correctness a small change to the transfer trajectory is made. The destination orbit will experience an offset in the RAAN parameter compared to the departure orbit. Note that, except for RAAN, the remaining orbital parameters are kept the same for initial and final

orbit to be able to compare the transfer and to measure the influence of the difference in the RAAN parameters. For the case with the reference frame fix, the previous tests are expected to result in the same ΔV values for a relative difference of zero degrees in RAAN, hence the fix. However, as soon as the RAAN of both orbits are different it is expected that the increasing inclination will decrease the correctness of the shaping method for the same transfer in that plane. This is, because both orbits have to be rotated over the angles i and Ω.

The accuracy of the transfer trajectories is determined as the deviation (error) in the computed ΔV required for the transfer. The velocity change is an accumulation of the accelerations required for the trajectory which is determined with an *RK4* integrator using the same number of steps as used for the computing the control points (20). Note that the accuracy of the ΔV computation is limited to the number of control points. Due to the fact that the number of control points are directly related to the computation time, a sacrifice has to be made on the accuracy on ΔV. Roegiers (2014) found an accuracy for ΔV of 1×10^{-5} AU/day ~ 17 m/s. Furthermore, Roegiers (2014) stated that for her test cases the exact same ΔV was computed with an increased number (100) of control points. However, the computational effort was increased severely. Therefore the ΔV computation is kept in the implemented form. The ΔV is a proper comparison value for accuracy, because it takes into account the errors during the whole trajectory and not just peak values of acceleration. For the test cases, two different levels of accuracy are defined, which have a maximum error with the reference case of respectively 1 % and 5 %. The reference case is the transfer trajectory in the plane with an inclination of zero degrees.

Before showing the full test case of a large range of applied relative RAAN changes, the focus is set upon Figures 4.5 and 4.6. Here the difference between the RFT fix can be seen very well. The figures that follow in this subsection show the limiting case of inclination for a given case of initial and final eccentric orbit and a set error tolerance on the required ΔV. From the first two figures it is unambiguous that without the RFT fix the inclination of transfer trajectories is very limited up till 15 ° for $\Delta\Omega = 0.0\,°$. Though the chosen elevation shape function resulted before in invalid trajectories for larger inclination values ($> 45.0\,°$), with the accuracy taken into account it is even more limited. For the GTOC6 problem these values would be even problematic, referring to the inclinations needed to map the moon faces in the order of 12.0 °. On the other hand, the asymptote shows very clearly the result of the RFT fix, allowing for higher inclinations without too much sacrifice on ΔV.

An important notion is that the absolute value of the inclination determines the level of accuracy. For larger inclinations the shaping method becomes less accurate. In case of RAAN, the accuracy is related to the relative difference in RAAN parameters of the initial and final orbit. A departure orbit with RAAN of 0 ° and an arrival orbit with RAAN of 5 ° results in a similar ΔV value as for orbits with RAAN's of respectively 165 ° and 170 °. Here the change in RAAN between initial and final orbit is the same resulting in a negligible difference in ΔV. Note, that the latter is verified, but not shown in the figures. The figures show the results for an initial orbit with RAAN of 0 °.

Another observation from Figures 4.5 and 4.6 is that the results are not symmetrical around $\Delta RAAN = 0.0\,°$. Take for example $\Delta VRAAN = \pm 10.0\,°$. There are slight differences in ΔV and therefore the maximum inclination within the tolerance level, albeit it very small and barely visible. This is due to small differences in trajectory length to reach the destination. Figure 4.7 clarifies the latter. Here three different transfer trajectories are shown from Earth to planet X. The trajectory with $RAAN = 0.0\,°$ is displayed for reference purposes to compare with the remaining two trajectories and makes one full revolution ($N_{rev} = 1$). It is clear that the trajectories with destination $RAAN = -10.0\,°$ and $RAAN = 10.0\,°$ are not equal, the former is shorter. Note that $RAAN = -10.0\,°$ is the same as $RAAN = 350.0\,°$ with zero

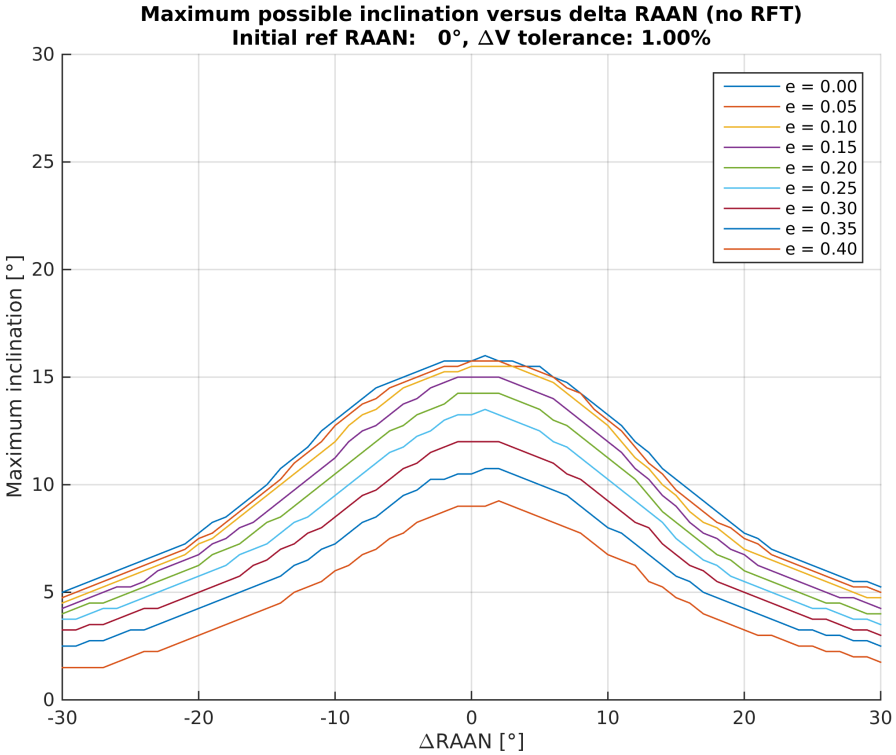

Figure 4.5: *Applicability for transfer trajectories with various eccentric orbits at different inclinations and small relative RAAN changes. Maximum ΔV error is set to $1.0\,\%$ with respect to the reference trajectory in the zero degrees inclined plane and the RFT is not used.*

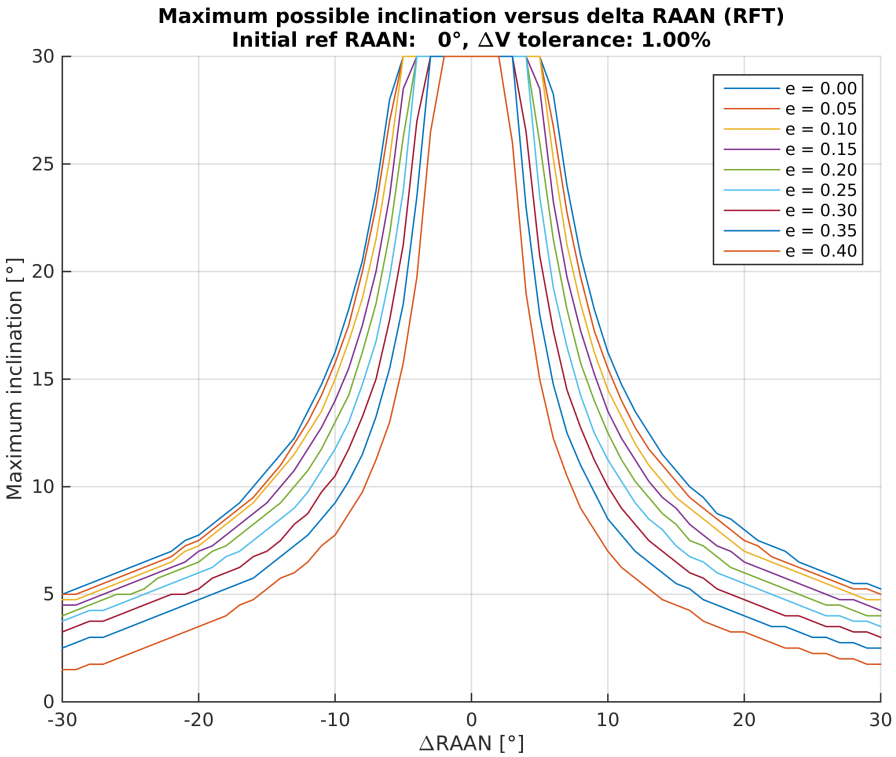

Figure 4.6: *Applicability for transfer trajectories with various eccentric orbits at different inclinations and small relative RAAN changes. Maximum ΔV error is set to $1.0\,\%$ with respect to the reference trajectory in the zero degrees inclined plane and the RFT is used.*

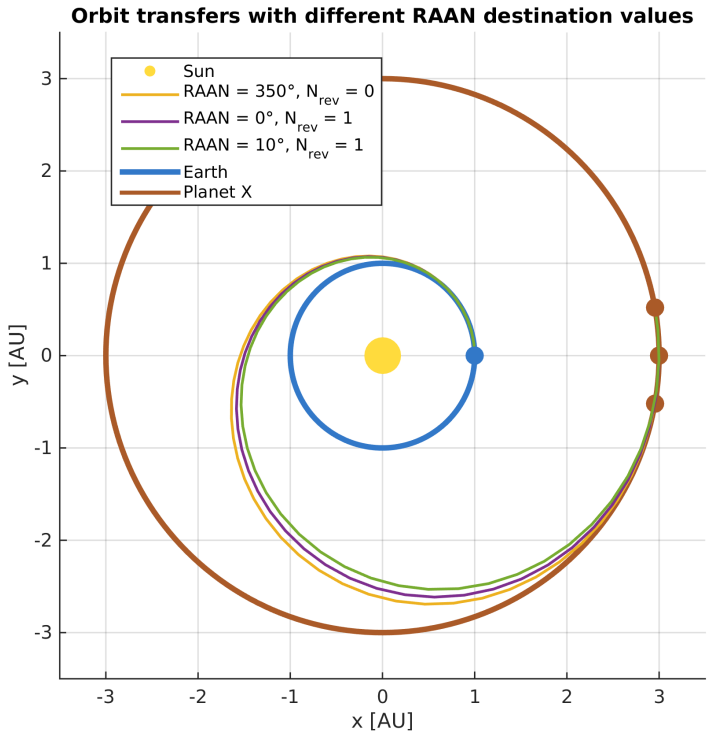

Figure 4.7: *Difference between orbits transfers where the arrival orbit has a RAAN offset with respect to the departure orbit of* $\pm 10.0\,^{\circ}$.

revolutions ($N_{rev} = 0$) whereas the other two trajectories make at least one full revolution ($N_{rev} = 1$).

In Figures 4.8 and 4.9 the full test is shown for a continuous changing RAAN parameter compared to the initial orbit. Here the RFT is not applied and it is immediately clear that the increasing inclination has a negative effect on the applicability of the algorithm. Even worse, the continuous increasing RAAN causes the maximum achievable inclination to decrease even further. Though a RAAN value of $360\,^{\circ}$ should equal the transfer with $0\,^{\circ}$, the $360\,^{\circ}$ has an extra revolution added to the transfer. A change of RAAN of $361\,^{\circ}$ is relatively speaking a change of $1\,^{\circ}$ with a complete revolution added. This is strictly taken not correct, because the RAAN parameter is used for the orientation of the orbit and not for the position within the orbit like the true anomaly. The RAAN parameter is constant for a given body (in the simplified problem descriptions in this report). However, as shown in Figure 4.7 the length of the trajectory influences the results and that is also related to the RAAN parameter of the departure and arrival orbit if all the other orbital parameters are kept constant. Therefore in the full test the number of revolutions is taken into account.

The results in Figures 4.8 and 4.9 show a decreasing trend of the maximum inclination possible with the given error tolerance for increasing revolutions. From these results it can already be concluded that the performance of the spherical shaping method without the RFT fix is debatable for three-dimensional trajectories. Also they would be insufficient for the GTOC6 problem. Therefore at this point it can already be said that the RFT fix is needed to use the spherical shaping method for the GTOC6 problem.

The same full RAAN test can be done for transfers with the RFT fix. These results are shown in Figures 4.10 and 4.11. Here the effect of the RFT fix can be seen clearly. Even

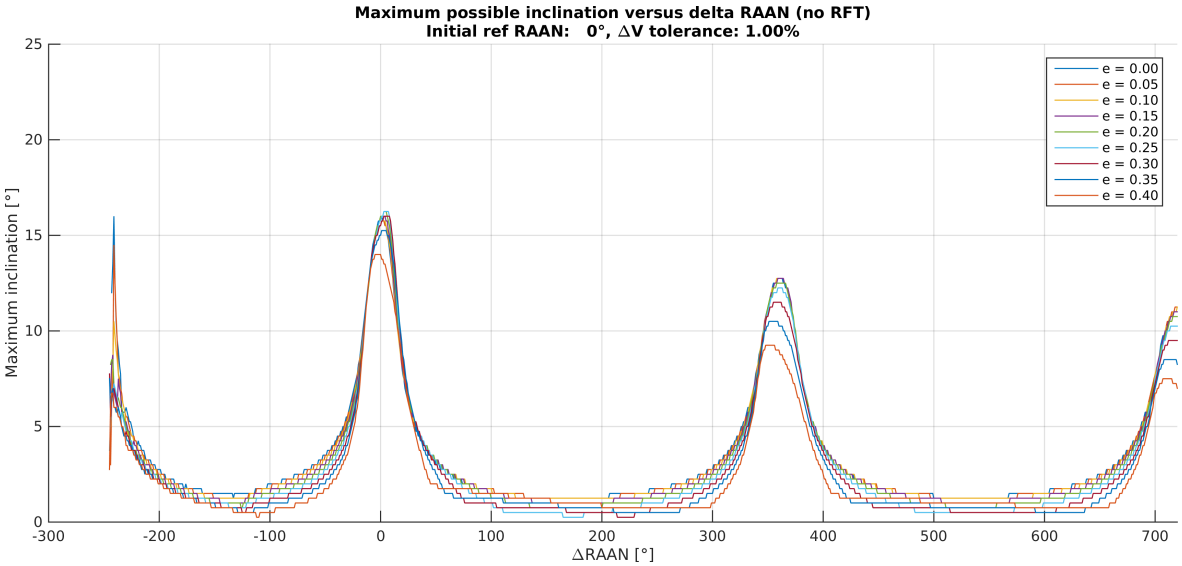

Figure 4.8: *Applicability for transfer trajectories with various eccentric orbits at different inclinations. Maximum ΔV error is set to $1.0\,\%$ with respect to the reference trajectory in the zero degrees inclined plane and the RFT is not used.*

for increasing number of revolutions, the efficiency for relative small RAAN changes is not affected.

Finally, two specific cases for a GTOC6 problem are shown. Note that the spherical shaping method normalises the semi-major axis of the given problem and therefore only the ratio of the initial and final orbit have influence on the results, and not the absolute values. Therefore most of the Galilean moon trajectories are not expected to show considerable differences compared to their Solar system counter part of Earth and planet X. Although the same moon transfer is an exception with its semi-major axis ratio of one, which will also be treated here. The two cases concern the largest and smallest semi-major axis ratio between the moons. The latter is also referred to as the same moon transfer (Europa to Europa in this case) and results are shown in Figure 4.12.

Here it is clear that the errors are severe. Note that the usual tolerance of $1.0\,\%$ and $5.0\,\%$ is not applied here. The reference trajectory for comparing here has the same departure and arrival conditions, which results in $\Delta V = 0.0\,\mathrm{m/s}$. Therefore a deviation of $\Delta V = 50.0\,\mathrm{m/s}$ is considered acceptable as maximum error. The main difference with the other transfers is that in this case there is no low thrust used to increase the semi-major axis. Therefore the ΔV value is actually needed just to overcome the accuracy flaw in the RAAN parameter of the transfers. It is quite clear that these errors become considerably large for small changes in RAAN already. With the previous results the ΔV required for increasing the semi-major axis was relatively flattened out and therefore still acceptable inclinations could be achieved. However, to conclude whether this is a huge disadvantage of the RFT fix for same moon (or planet) transfers, the same case is repeated without the RFT fix.

The results for the RFT fix and without are shown in respectively 4.13 and 4.14. Surprisingly, the original shaping method without the RFT fix performs even worse around $\Delta RAAN = 0.0\,°$, which was not confirmed by Roegiers (2014). For an eccentricity of zero the trend of both graphs is nearly the same except for around $\Delta RAAN = 0.0\,°$, where the RFT fix shows its advantage. The remaining eccentric trajectories without the RFT fix display worse accuracies compared to their RFT fix counter parts. From this it can be concluded that the RFT fix is necessary for the spherical shaping method.

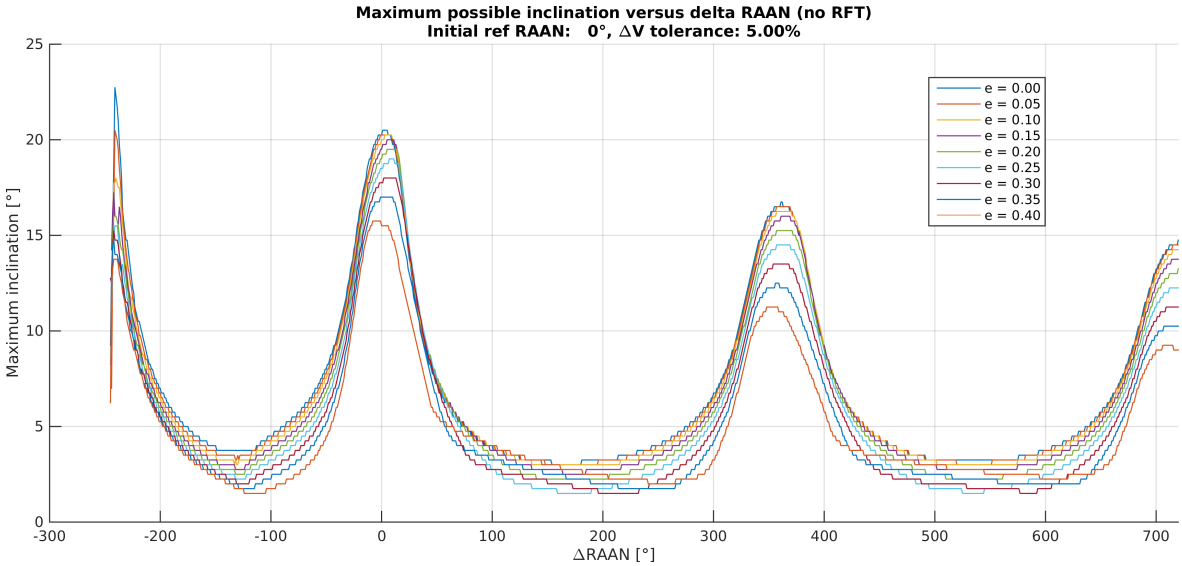

Figure 4.9: *Applicability for transfer trajectories with various eccentric orbits at different inclinations. Maximum ΔV error is set to $5.0\,\%$ with respect to the reference trajectory in the zero degrees inclined planeand the RFT is not used.*

Furthermore, Figures 4.15 and 4.16 display various different moon transfers for the usual range of eccentricities till the value 0.4. Though the colors are used to indicate the moon transfer and not the eccentricity, it is clear from the previous results that the largest eccentricities cause the most stringent inclination limitations.

Also the reverse transfer, from large initial to small final orbit, has negligible influence on the results, even though the transfer trajectory is not exactly the same. This can also be said for different reference values for the RAAN of the initial orbit. The change in the final orbit is relative to the initial orbit. However, using a reference RAAN of $0.0\,^\circ$ or for example $90.0\,^\circ$ has shown negligible difference.

Besides the fact that the RFT fix is preferable in terms of accuracy there is another important reason to go with RFT. Looking at the CSU solution of the trajectories in Chapter 3, it is clear that low thrust is only used for small relative changes in RAAN whereas the inclination does reach reasonable larger values which are absolute. The latter has more impact on the accuracy than the small relative RAAN changes. Therefore, also for the GTOC6 problem it is preferred to go with the RFT fix in the spherical shaping method.

Although it is clear from these benchmarks that the RFT fix is to be used to model the trajectories, the RFT fix is not perfect either. An additional measure is needed to make sure that valid and reliable trajectories are produced during the optimisation. This comes in the form of a black box for the spherical shaping method that limits the transfer trajectories with respect to various orbital elements and therefore avoiding too much error. The black box is discussed in the next section.

4.6 The Black Box and its Limitations

After having benchmarked the spherical shaping thoroughly, there are several characteristic trajectory cases that can be distinguished using the spherical shaping method. They are as follows

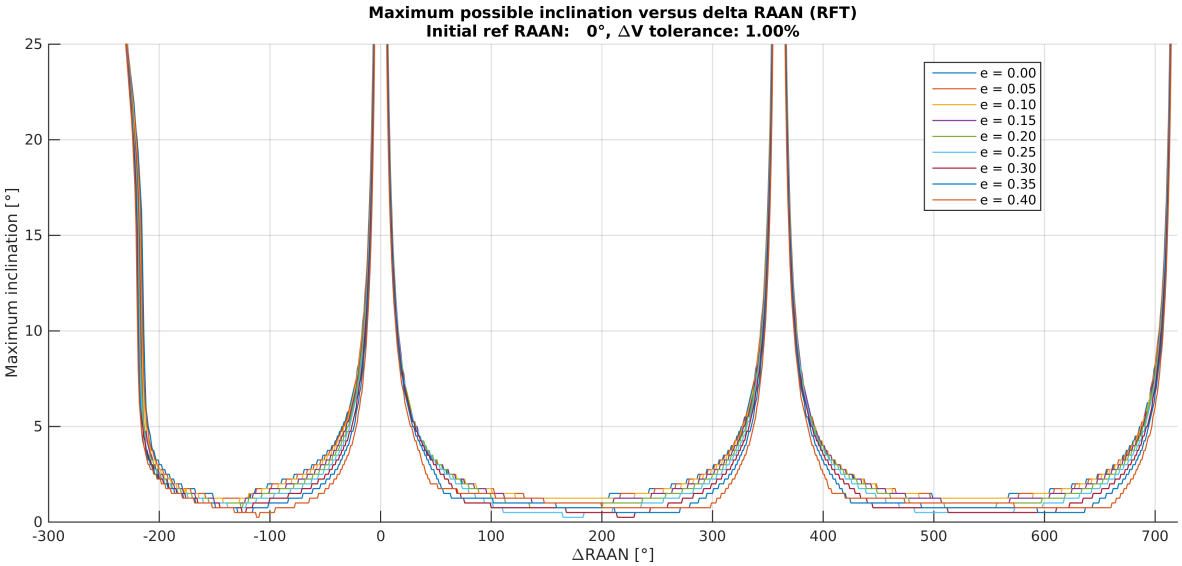

Figure 4.10: *Applicability for transfer trajectories with various eccentric orbits at different inclinations. Maximum ΔV error is set to $1.0\,\%$ with respect to the reference trajectory in the zero degrees inclined plane and the RFT is used.*

1 The shaping method is not able to shape a viable trajectory with the given boundary conditions. This is most likely due to extreme trajectories with highly eccentric and large inclination values. Also unsuitable combinations of time of flight and number of revolutions can cause failed trajectories (negative and/or NaN values of D).
2 The shaping method is able to find plausible/viable trajectories. However, the final boundary conditions (position and velocity) are not analytically verified, i.e. the final control point has too much error with the desired arrival conditions.
3 The shaping method is able to find plausible/viable trajectories. However, the final boundary conditions (position and velocity) are not verified with numerical integration. This can be due to mainly accumulating errors during the numerical integration, or mainly due to errors already existing in the analytical trajectory, though still within the tolerances, or a combination of both. In case the analytical errors are relatively large, near the tolerance values, it is likely that the error due to numerical integration will surpass the set tolerances.
4 The shaping method is able to find plausible/viable trajectories and has verified them with numerical integration. Although the trajectory is viable, it is still possible that the computed trajectory is far from optimal due to the nature and assumptions of the shaping method.

Only the last option results in valid trajectories. Although the trajectories can still be far from optimal there is a solution to avoid these non-optimal solutions in the optimisation phase. The proposed solution is a black box model for the low-thrust trajectory where ΔV penalties are applied to non-reliable trajectories. In the black box there are three different regions defined which are shown in Figure 4.17.

The first one is trajectories that are thought to have a maximum ΔV uncertainty of one percent, which are considered valid, and do not receive a penalty. Trajectories with a maximum ΔV uncertainty of five percent are also considered valid, but not accurate, and therefore have a penalty related to their uncertainty added to their ΔV to encourage the optimiser to find better solutions. Instead of computing for every transfer the reference trajectory, the uncertainty is based on the relative distance between the inner and outer boundaries of the

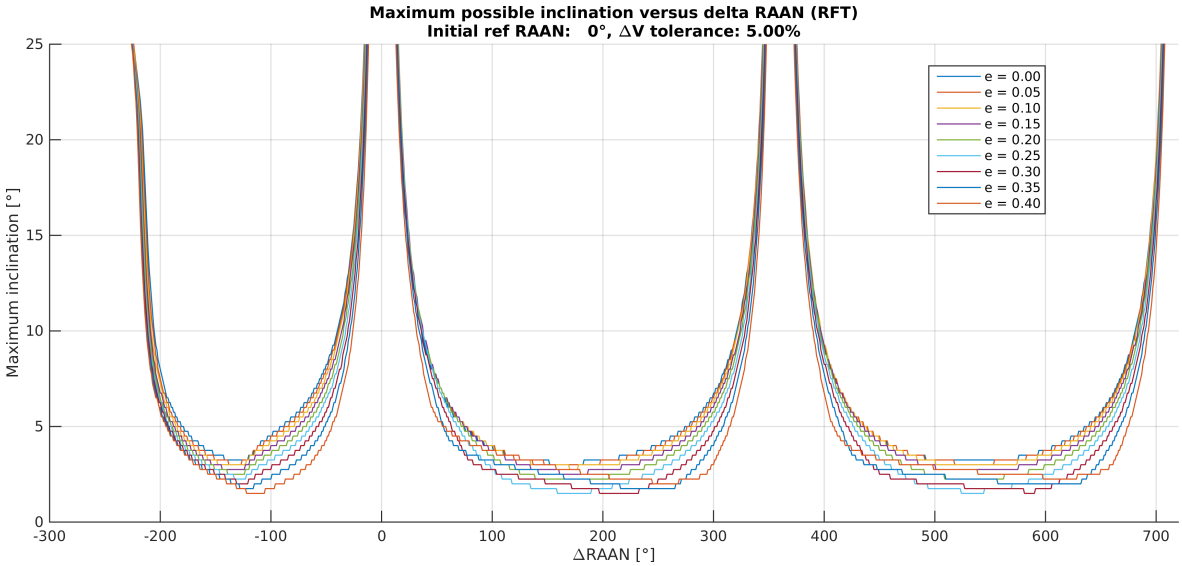

Figure 4.11: *Applicability for transfer trajectories with various eccentric orbits at different inclinations. Maximum ΔV error is set to $5.0\,\%$ with respect to the reference trajectory in the zero degrees inclined plane and the RFT is used.*

triangle. For this the inclination and the change in RAAN is used of the boundary conditions for the spherical shaping method.

The final region is outside the five percent uncertainty and these solutions are considered invalid and results in a considerable large penalty, also related to the uncertainty. The scaling of the penalties is needed to avoid the solution space of the problem becoming flat and therefore impossible for the optimiser to get to the actual solutions. Each region is defined by creating a triangle shape that fits the results shown in the figures in this subsection. As mentioned before, the ratio of semi-major axis and the eccentricity are of influence on the accuracy and therefore they need to scale the triangle that defines the valid region.

4.7 Final Remarks

Although the capture phase is not treated in this thesis, it is noteworthy to mention that the spherical shaping method in its current implementation will have problems computing near-parabolic trajectories. Adding potentially high inclinations that lower the performance even more, leads to the conclusion that the spherical shaping method is not sufficient for assessing the approach or capture phase of the GTOC6. However, this can be overcome by the use of an improved elevation function for shaping the third dimension. This is the thesis project of a fellow Master of Science student at the moment of writing (Vroom, 2015).

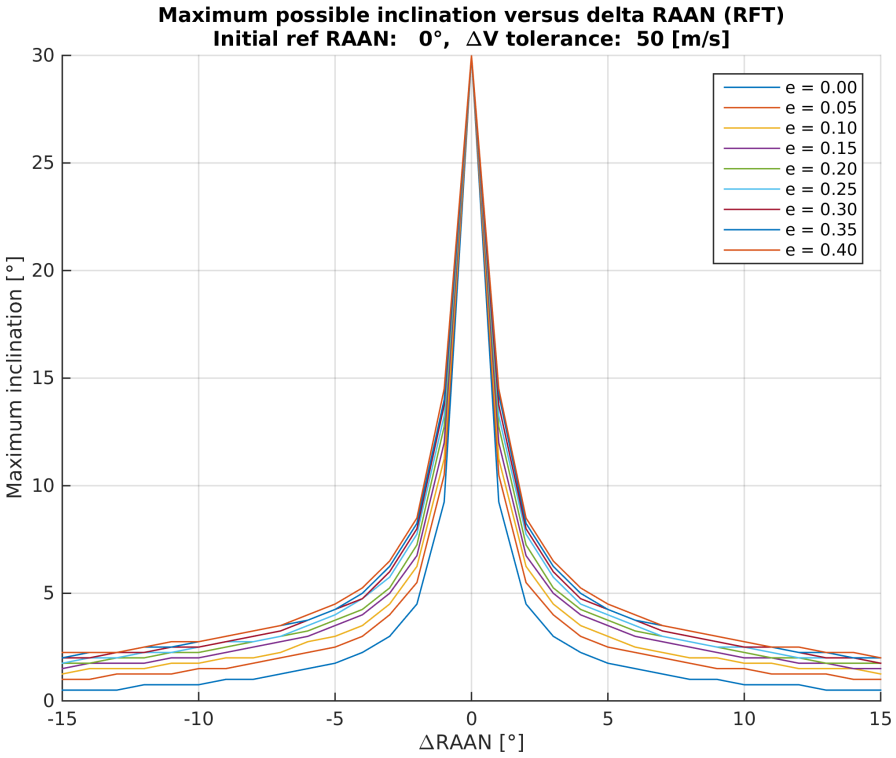

Figure 4.12: *Applicability for an Eu-Eu transfer trajectory with various eccentric orbits at different inclinations. Maximum ΔV error is set to $50.0\,\mathrm{m/s}$ with respect to the reference trajectory in the zero degrees inclined plane and the RFT is used.*

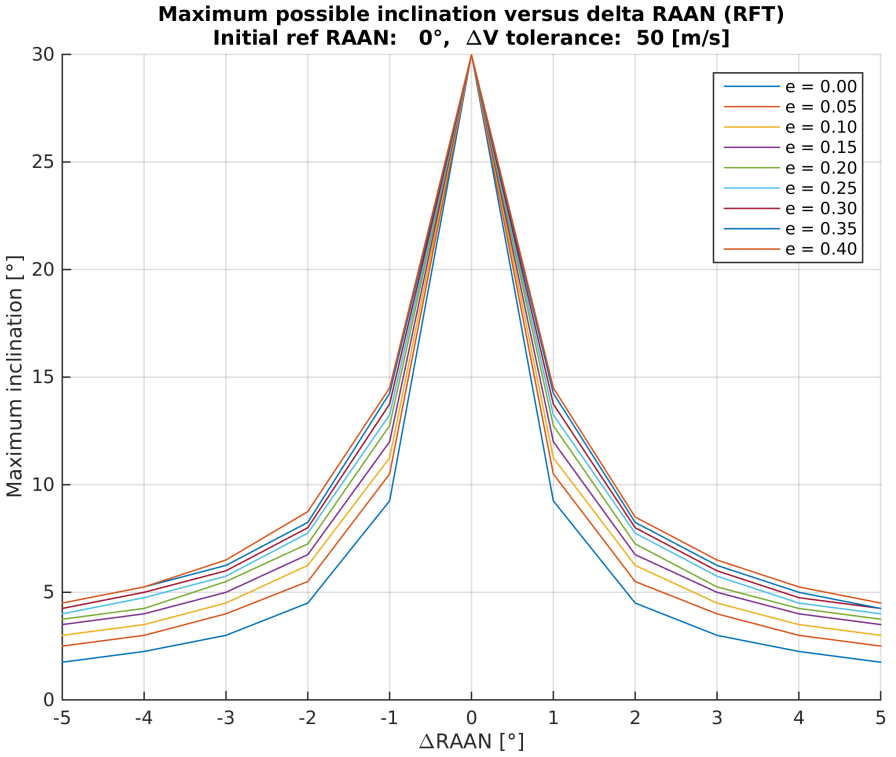

Figure 4.13: *Applicability for an Eu-Eu transfer trajectory with various eccentric orbits at different inclinations. Maximum ΔV error is set to $50.0\,\mathrm{m/s}$ with respect to the reference trajectory in the zero degrees inclined plane and the RFT is used.*

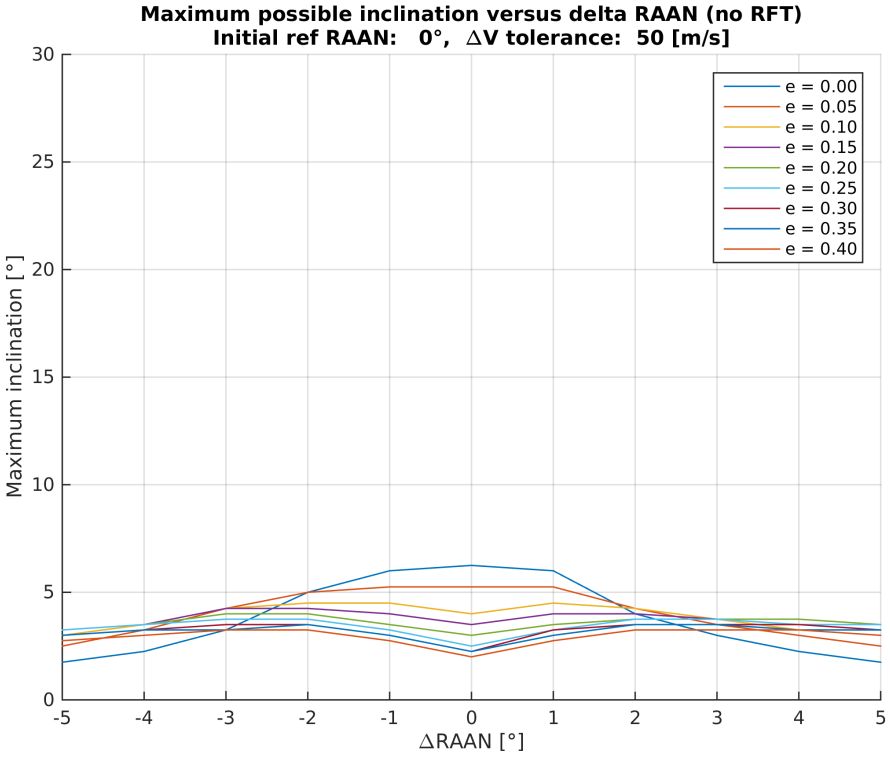

Figure 4.14: *Applicability for an Eu-Eu transfer trajectory with various eccentric orbits at different inclinations. Maximum ΔV error is set to $50.0\,\mathrm{m/s}$ with respect to the reference trajectory in the zero degrees inclined plane and the RFT is not used.*

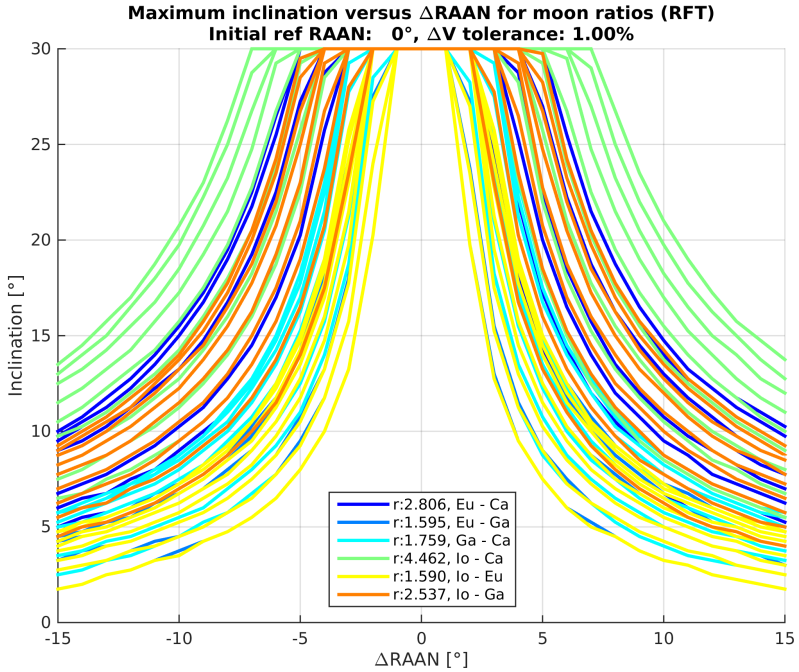

Figure 4.15: *Applicability for different moon transfer trajectories with various eccentric orbits at different inclinations. Maximum ΔV error is set to $1.0\,\%$ with respect to the reference trajectory in the zero degrees inclined plane and the RFT is used.*

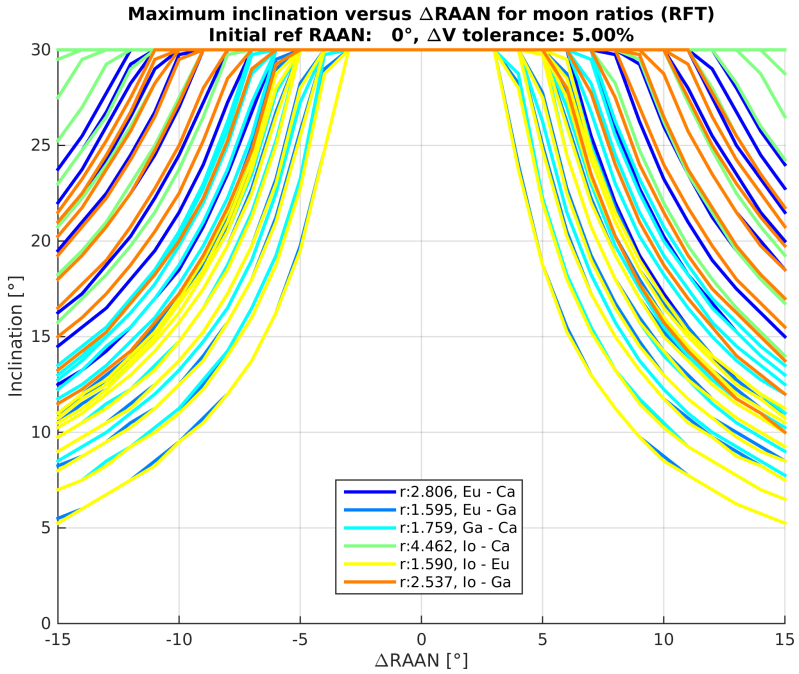

Figure 4.16: *Applicability for different moon transfer trajectories with various eccentric orbits at different inclinations. Maximum ΔV error is set to $5.0\,\%$ with respect to the reference trajectory in the zero degrees inclined plane and the RFT is used.*

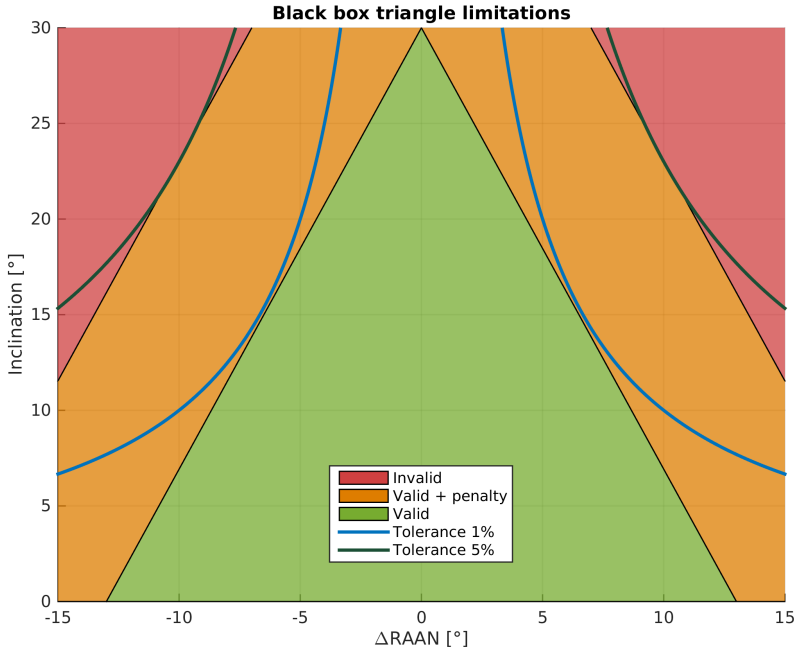

Figure 4.17: *The regions defining the limitations of the blackbox for the spherical shaping method.*

5
Multiple Gravity Assist

THE gravity-assist manoeuvre is an important asset in the GTOC6 problem. Besides the fact that the flybys are synonymous for mapping the moons from a certain distance, more importantly, they allow for significant changes required in the trajectory to visit different faces and moons. In this chapter the single non-powered flyby is discussed and extended with the multiple gravity assists.

5.1 Single Unpowered Gravity-Assist

The gravity-assist manoeuvre is used to alter the orbital characteristics of the spacecraft. Basically, the fly-by is a hyperbolic trajectory near the attracting body that occurs inside the sphere of influence of that body. For the swing-by, the law of preservation of the total momentum applies, see Equation (5.1).

$$m_{sc,i}V_{sc,i} + m_{M,i}V_{M,i} = m_{sc,f}V_{sc,f} + m_{M,f}V_{M,f} \tag{5.1}$$

Here the subscripts $_i$ and $_f$ denote the initial and final state of the gravity-assist respectively. For clarity, Jupiter is the dominating gravitational body whereas the moon is used for the gravity-assist (subscript $_M$). Furthermore, $_{sc}$ stands for the spacecraft.

During the fly-by, the spacecraft and the moon exchange momentum resulting in an increase of velocity or change in inclination (rotation of the velocity vector) for the spacecraft and a decrease for the body assuming constant mass for both. Note that this is the case for a gravity-assist where the spacecraft moves 'behind' the moving body. The reverse is true when the spacecraft performs the fly-by in 'front' or 'ahead' of the body. Since the moon is relatively large in mass compared to the spacecraft, the effective velocity of the moon is not altered by the gravity-assist.

For the GTOC6 problem the classical gravity assist is required for the trajectories. This means that moons are assumed to have no atmospheres and that there is no impulsive, powered flight, during the flyby.

A detailed view of the swing-by is given in Figure 5.1. Here the in-plane geometry is shown of the swing-by which is happening inside the sphere of influence of the moon. The hyperbolic

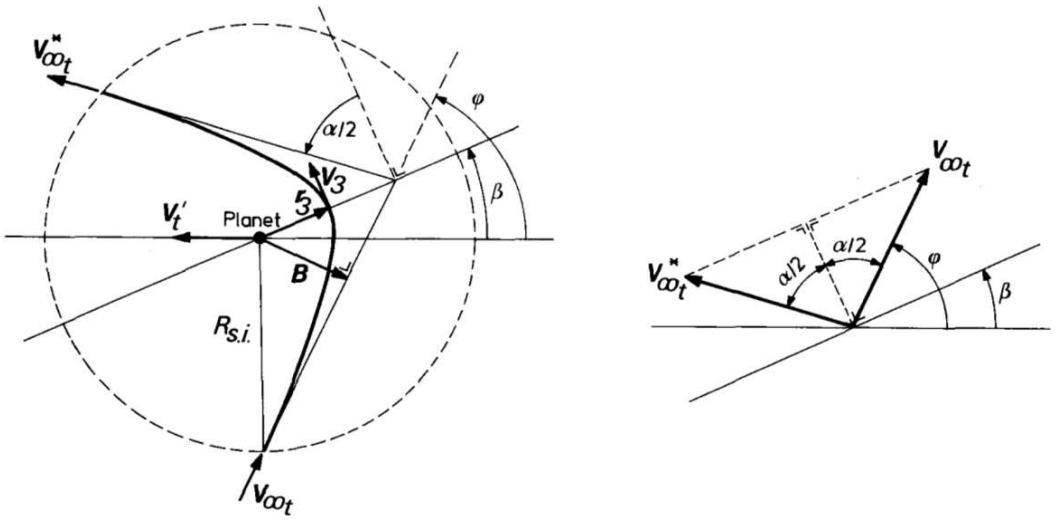

Figure 5.1: *The in-plane geometery of hyperbolic encounter trajectories. (Cornelisse et al., 1979)*

excess velocities V_{∞_t} and $V^*_{\infty_t}$ are with respect to the moon and represent the velocities with which the spacecraft enters and leaves the sphere of influence. Due to the conservation of angular momentum the magnitudes of the excess velocities are the same. However, the outgoing excess velocity is bent in a new direction after the gravity assist. In the reference frame of Jupiter, in which the velocity of the moon V'_t (in-plane) is defined, this means a different velocity. Note that V'_t is the projected velocity of the moon in the Jovicentric reference frame on the trajectory plane. The asymptote of the incoming hyperbolic trajectory approximates the aiming direction of the spacecraft for the gravity assist. Here the distance between the centre of mass of the moon and the asymptote is defined as the impact parameter B, where the vector $\boldsymbol{B}$ points to the aiming point on the asymptote. Furthermore, V_3 and r_3 denote the velocity and radius at the pericentre of the fly-by trajectory. The final important parameters are the two angles α and β. The angle α represents the asymptotic deflection angle and β defines the angle between the moon's velocity and the symmetry line of the hyperbolic trajectory.

The total change in velocity of the spacecraft due to the gravity assist can be derived from the geometry on the right in Figure 5.1. Subtracting the two excess-velocity vectors, initial and final, results into a simplified relation which is given in Equation (5.2).

$$\Delta V = 2V_{\infty_t} \sin \frac{1}{2}\alpha \tag{5.2}$$

Note that the initial and final excess-velocity vectors are equal in magnitude but not in direction.

The classical gravity-assist algorithm is part of the Tudat software library and allows for a full three-dimensional swing-by. Due to the fact that the flyby is performed instantaneously when the position of the flyby body and spacecraft coincide, the spacecraft does not enter a sphere of influence in which the hyperbolic trajectory around the moon is performed. Therefore the orientation of the incoming trajectory can not be determined with respect to the swing-by body, which means that it is not known whether the spacecraft passes ahead or behind the body. Another parameter is needed to represent the out-of-plane component for the gravity assist. This is introduced as a rotation angle which defines the rotation with respect to the flyby plane that is defined by the velocity vector of the spacecraft and that of the swing-by body.

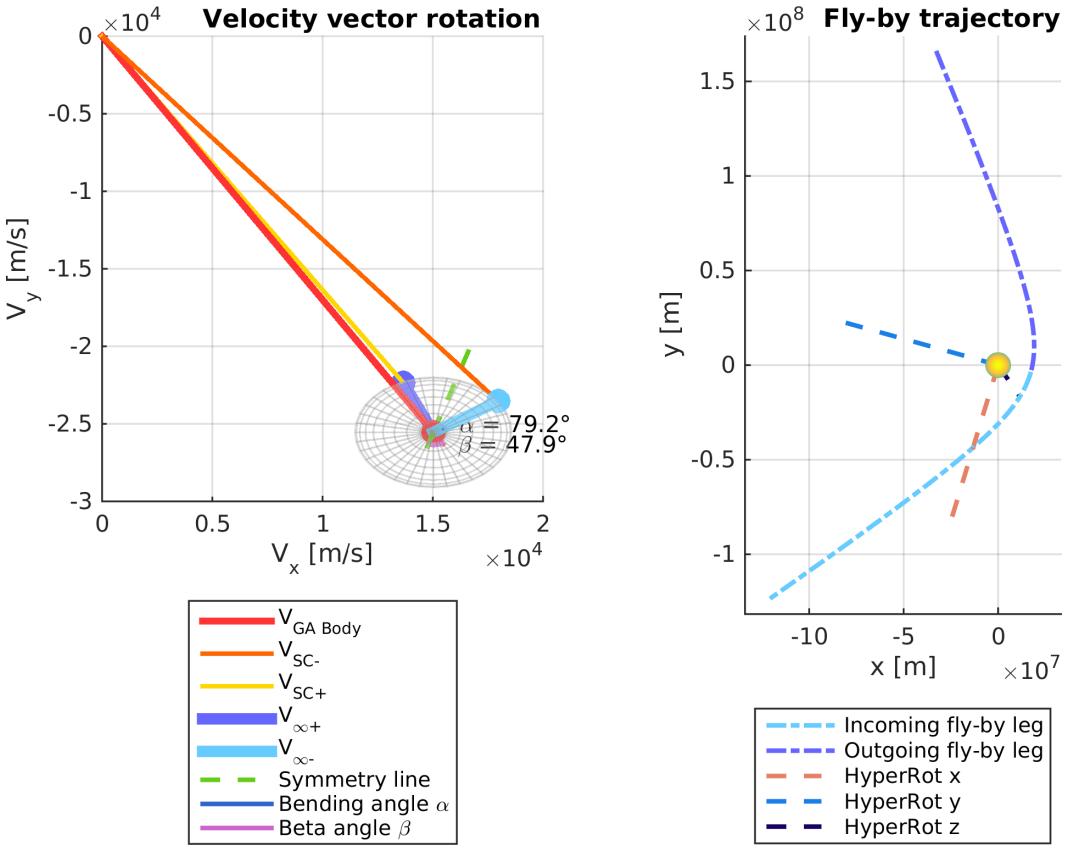

Figure 5.2: *Visualisation of the Gravity Assist Manoeuvre using the Tudat method.*

In Figure 5.2 the flyby as computed with the Tudat algorithm is visualised. On the left the velocities of the spacecraft and the gravity-assist body are shown with respect to Jupiter. Also the excess velocities with respect to the flyby body are shown together with the sphere of rotation of these velocities due to the swing-by. On the right the actual flyby trajectory is depicted inside the sphere of influence.

To use the algorithm of Tudat, a total of five input parameters need to be supplied. First the flyby-body velocity, the spacecraft velocity and the gravitational parameter of the swing-by body are given at the moment of the instantaneous flyby. As said before, because there is no sphere of influence and therefore no flyby trajectory in said sphere, the orientation of the trajectory around the gravity-assist body and also the pericentre radius of the flyby can not be determined. Therefore these are supplied as well to the function, where the former is the rotation angle mentioned before.

5.2 MGA

Multiple gravity-assist (MGA) manoeuvres are used for the GTOC6 problem to map the moons and to decrease the amount of thrust needed by the propulsion unit. In essence the MGA is a sequence of single gravity assists patched together, and in case of GTOC6 with one or more low-thrust legs in-between. No other tools or algorithms are needed to perform MGA. The sequence with the moon destinations does need to be known or can be made part of the optimisation process. More information on the trajectory model for optimisation is discussed in Chapter 8.

A problem that is introduced when patching multiple GA's is that the next GA depends on the outcome of the previous GA's. It is possible that a sequence of 3 flybys yields a promising result whereas adding the fourth GA will make the solution far from desirable. According to Novak (2012) this dependency can be removed by decoupling the legs from the GA's. This is done by introducing powered swing-bys. However, GTOC6 is only focussed on the non-powered gravity-assists. Although, the continuous low-thrust legs between the GA's allow for a certain degree of decoupling. More on this will be treated in the results of the optimisation problems in Chapter 10.

6

Moon Mapping

An important element of the GTOC6 problem is the global mapping of the moons which is used as the main objective to be maximised given a number of constraints. For this thesis study the mapping objective is taken into account as well, although not all simulations performed in Chapter 10 are focussed on maximising the global mapping objective, they do compute the mapped faces alongside the objective of optimising the ΔV. In this chapter the mapping of the moons is further elaborated upon and it also discusses how the points associated with it are computed.

6.1 Computational Geometry

In Chapter 2 the principle of moon mapping was explained and also how points can be scored. A face on the moon was said to be mapped when the pericentre vector of the flyby with respect to the moon pierces through that face on the football surface. To compute if the face is pierced by the vector, use is made of computational geometry, specifically inclusion methods. There are several if not many ways to assess the latter, however for the GTOC6 problem a few considerations are important. First of all, the complexity and the size of the problem result in a considerable large number of flyby computations and therefore the calculations of the coverage of the moon faces as well. For this reason the algorithm to determine the moon mapping score should be relatively quick and should also be lightweight in respect to memory management. Another consideration is converting the three-dimensional problem into a two-dimensional problem. This simplifies the algorithms considerably and keeps the algorithms within a few computations.

The buckly ball consists of 32 polygons that represent the faces, from which 12 are pentagons and 20 hexagons. These polygons are regular and convex, meaning that they do not have intersections and that the sides and angles are equal. This simplifies the algorithm for finding if the point (projection of pericentre vector onto hexagon or pentagon plane) is inside the area of the hexagon or pentagon.

From Haines (1994) and Ward and Heckbert (1994) follows a comparison of several convex inclusion methods and according to the latter the best option for convex polygons with a few sides is either the half-plane method or the spackman test. Both methods do require a little

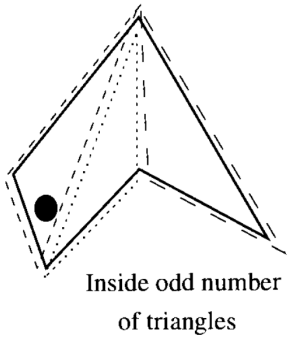

Figure 6.1: *Triangle-fan method. Adapted from Ward and Heckbert (1994).*

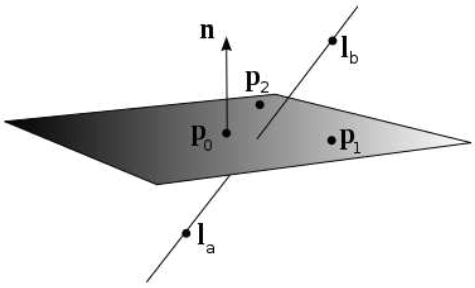

Figure 6.2: *Line-plane intersection. (Morris, 2007).*

preprocessing and additional memory for storing data on the polygons. However this is not an issue, due to the fact that the computations will not happen on memory-limited systems like on a spacecraft.

Also, Ward and Heckbert (1994) states that the half-plane method is slightly faster for polygons having 3 or 4 sides compared to the spackman test. On the other hand, for a polygon with 10 sides the performance is similar. Note that the moon polygons have 5 or 6 sides and therefore it can be reasonably assumed that the half-plane method is slightly faster there as well. Walker and Snoeyink (1999) confirmed that the half-plane performs better than the spackman test. Therefore the half-plane method is selected to assess the inclusion of the pericentre vector in the faces of the moon. The method is explained in the next section.

6.2 Pericentre Vector Pointing Algorithm using the Half-Plane Test

The half-plane method is based on a simple dot product computation. One takes as example a triangle that has three sides and vertices. The first side, made of two vertices is used as baseline and the third vertex is used to define the 'inside' of the triangle. By computing the dot product it can be found that the third vertex is on the left or the right side of the baseline. The same can be done for the point of interest to check for inclusion. If the dot products have the same sign, than the point is on the same half-plane defined by the baseline as the third vertex. Doing so for all the sides of the triangle in an ordered fashion it can be determined if the point is inside a given geometry, in this case the triangle. It is important that the vertices are sorted in order of drawing, either clockwise or counter-clockwise, to make sure that all sides of the polygon have the same 'inside' direction. This process can be sped up by reversing the computation. Instead of determining for every side that the point is inside the triangle, one can also determine if the point is outside the triangle. As soon as a point is on the 'outside' of one of the baselines, the polygon can be excluded from the search.

With the half-plane method all kinds of polygons can be tested and the usual way is to split up the polygon into multiple triangles that are formed with the vertices of the main polygon. A visualisation is given in Figure 6.1.

In case of a convex polygon, the triangle fan is not necessary, and one can traverse the sides of the complete polygon with the half-plane method, instead of making several smaller triangles, and therefore saving computation time (Haines, 1994). This is also referred to the exterior edge strategy which according to Haines (1994) resulted in a factor of 1.7 - 11 times faster computations. Note that all computations are done in two-dimensional space. Therefore the

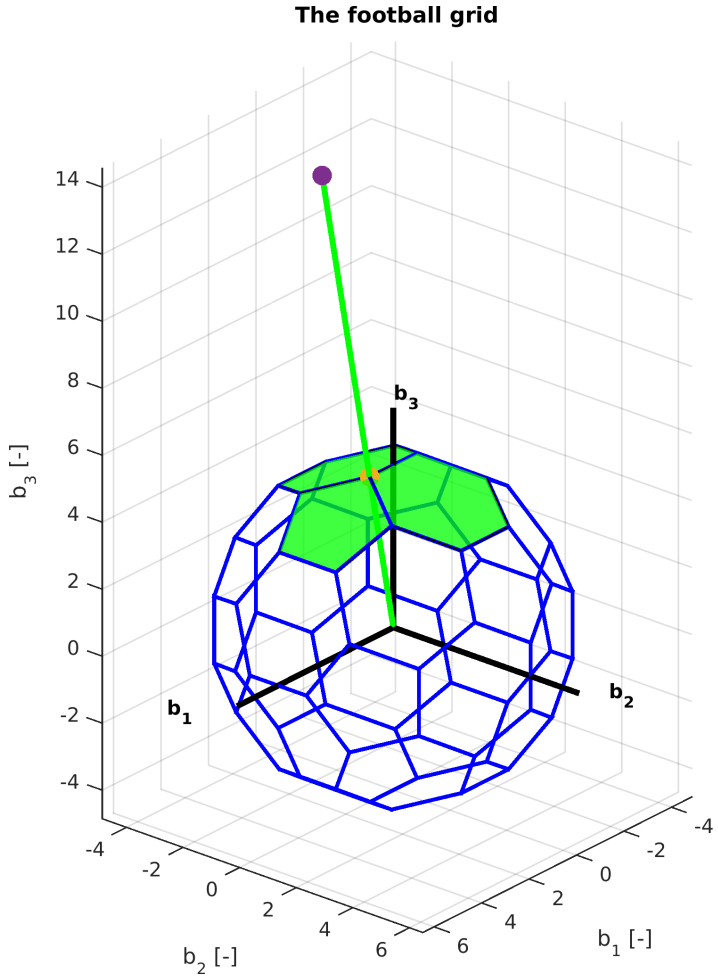

Figure 6.3: *Moon mapping football and the pericentre vector of the flyby piercing through the corner of three faces.*

three-dimensional problem needs to be converted to a two-dimensional case. To do this, first the pericentre vector is projected onto the plane of where the polygon of interest is residing. This is in three-dimensional space as shown in Figure 6.2 where a line-plane intersection computation is performed.

With the pericentre vector intersection found in the plane of the polygon the problem can be converted to two-dimensional to allow for the half-plane check. The conversion is simplified by ignoring one of the three dimensions for both the point as well as the polygon, preferably the one with the least amount of information that will change the shape of the geometry the least. By computing the normal of the polygon plane the largest component and the corresponding dimension can be set to zero. Now the problem is two-dimensional. Note that due to the conversion, the polygon is still convex, however, most likely not regular any more.

Since all polygons of the moon have to be evaluated, another additional computation is added to speed up the algorithm. By defining a bounding box in the form of a cone in the three-dimensional problem, where the tip of the cone is at the origin, it can be quickly determined (with inner product) if the pericentre vector is inside the cone and thus most likely on the inside of the polygon as well. This feature gave on average a speed boost of a factor 5 compared to checking all the polygons in detail.

Now the algorithm is capable of detecting the polygons that have been pierced by the flyby vector. Note that if the vector goes exactly through a side or vertex that is shared by multiple

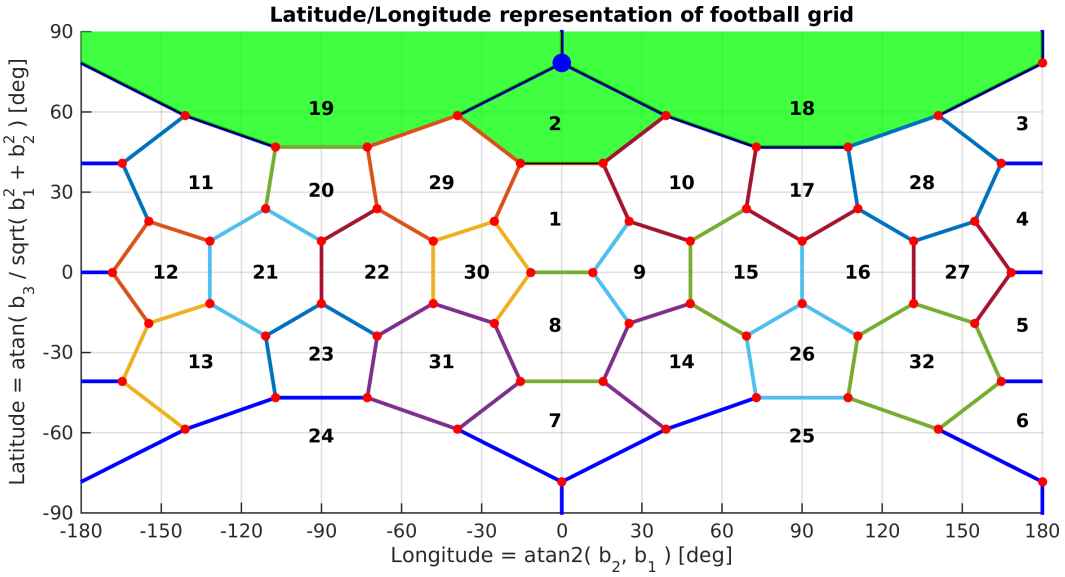

Figure 6.4: *The longitude and latitude presentation of the mapping with the pericentre vector of the flyby piercing through the corner of three faces.*

polygons, all polygons are possible for being mapped. However, only one face can be mapped at the time, and the face with the highest number of points will be returned as the surface that is mapped with the given pericentre vector of the flyby as specified by the original GTOC6 assignment.

Part III

Optimisation & Validation

7 Optimisation Algorithm

THE optimisation of the actual trajectory including multiple gravity assists is performed with a well-known evolutionary algorithm. After a thorough selection procedure in the preceding literature study (Hoving, 2014) it followed that DE is the most suitable candidate for optimising a wide range of trajectory problems. However, DE comes in many varieties, each having their pro and cons. This chapter will explore the available varieties and put them to the test to see what the optimal settings are for the DE algorithm to tackle low-thrust gravity-assist problems.

7.1 DE Basic Working Principle

One of the most powerful stochastic real-parameter optimisation methods nowadays, is the DE algorithm. The algorithm is from the mind of Storn and Price (1997) and was designed to fulfil the following requirements.

- Optimise non-differentiable, non-linear and multi-modal objective functions.
- Allow for parallel computing to handle complex problems that require considerable computational effort.
- Easy to use algorithm by needing only a few robust control variables.
- Good and consistent convergence to the global minimum in independent trials.

The DE method is similar to the generally known genetic algorithm (GA), also based on evolutionary computing. However, DE performs better than GA in nearly most cases concerning trajectory design (Biesbroek, 2006). Because of its popularity there are many versions and papers to be found on the subject. This section details the original method based on Storn and Price (1997) and applies additional information from Das and Suganthan (2011).

The basic DE method consists of four main stages which are initialisation, mutation, crossover and selection. Apart from the first stage, the other stages will be repeated until convergence is found.

Initialisation

The algorithm starts with an initial population of size NP. Each individual in the population is defined as a D-dimensional parameter vector, see Equation (7.1).

$$\boldsymbol{x}_{i,\,G}, \qquad i = 1,\, 2,\, ...,\, NP \tag{7.1}$$

The initial values for the parameter vectors are chosen randomly and in most cases are based on a uniform probability distribution. It is possible to use a preliminary solution as initial population $\boldsymbol{x}_{nom,\,0}$. To ensure good optimisation a normally distributed random deviation should be added on top of the preliminary solution $\boldsymbol{x}_{nom,\,0}$.

Mutation

Mutation refers to creating new parameter vectors for the next generation. DE takes three random parameter vectors defined by the mutually exclusive indices r_1, r_2 and $r_3 \in 1,\, 2,\, ...,\, NP$. The mutant vector follows from taking the weighted difference from the vectors r_2 and r_3 and add this to the first vector r_1. In Equation (7.2) the mathematical relation for the mutation vector $\boldsymbol{v}_{i,\,G+1}$ is given.

$$\boldsymbol{v}_{i,\,G+1} = \boldsymbol{x}_{r_1,\,G} + F \cdot (\boldsymbol{x}_{r_2,\,G} - \boldsymbol{x}_{r_3,\,G}) \tag{7.2}$$

Here the parameter F is the weight of the differential term and is defined as $F > 0$ and is one of the control parameters of the DE algorithm. The mutation is applied to each target vector $\boldsymbol{x}_{i,\,G}$ in the current generation of the population. In Figure 7.1 the mutation has been visualised for a two-dimensional objective function.

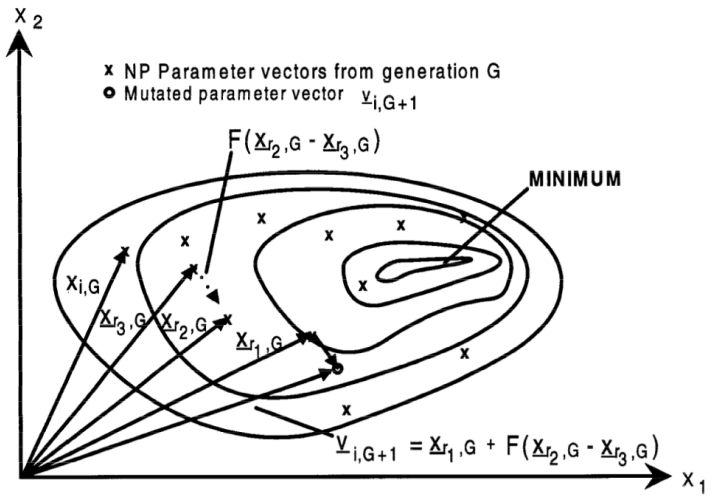

Figure 7.1: *An example of a two-dimensional cost function showing its contour lines and the process for generating $\boldsymbol{v}_{i,\,G+1}$. (Storn and Price, 1997)*

Crossover

Crossover is added to the process to increase the diversity of the newly formed parameter vectors. The vector acquired after crossover is called the trial vector $\boldsymbol{u}_{ji,\,G+1}$ and the mathematical relation is given in Equation (7.3).

$$\boldsymbol{u}_{ji,\,G+1} = \begin{cases} \boldsymbol{v}_{ji,\,G+1} & \text{if } (\text{randb(j)} \leq CR) \text{ or } j = \text{rnbr(i)} \\ \boldsymbol{x}_{ji,\,G} & \text{if } (\text{randb(j)} > CR) \text{ and } j \neq \text{rnbr(i)} \end{cases} \qquad j = 1,\, 2,\, ...,\, D \tag{7.3}$$

In Figure 7.2 the crossover operation is visualised. From Equation (7.3) follows that parameters of the trial vector can be either taken from the mutant vector or the original target vector depending on the conditions defined by random numbers and the parameter CR. The parameter CR is the crossover constant $\in [0,\, 1]$ and is a control input for the DE algorithm. The term randb(j) $\in [0,\, 1]$ is determined with a uniform random number generator for the j^{th} evaluation whereas the last term rnbr(i) $\in [1,\, 2,\, ...,\, D]$ resembles a random chosen index defining the parameter of the vector. From the conditions it follows that at least one parameter of the mutated vector is passed on to the trial vector.

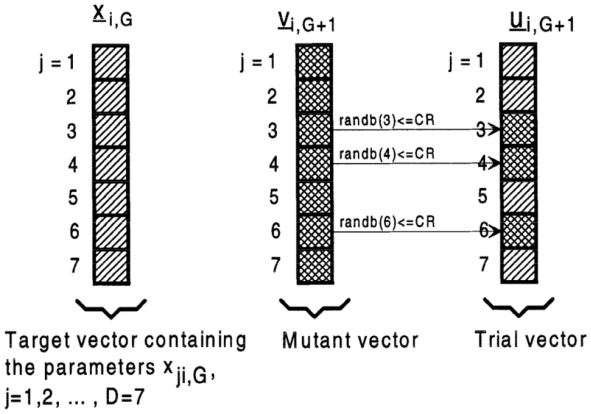

Figure 7.2: *Illustration of the crossover process for $D = 7$ parameters. (Storn and Price, 1997)*

Selection

The last step determines whether a parameter vector that has undergone mutation and crossover continues to the next generation. By determining the objective function of the trial vector $\boldsymbol{u}_{ji,\,G+1}$ and that of the target vector $\boldsymbol{x}_{i,\,G}$ the optimal of the two can be found and this vector proceeds to the next generation. In case the performance of the two vectors is equal, the trial vector is chosen to ensure that the individual target vector does not get stuck on a plane of the objective function.

7.2 DE Schemes

The DE algorithm knows many varieties. A common way to distinguish the different schemes is by using the notation $DE/x/y/z$. Here x represents the method for choosing the target vector for mutation. The DE described here uses randomly chosen vectors which is denoted by 'rand'. However, instead of random also the optimal target vector of the current generation can be chosen. This is denoted as 'best'. The parameter y defines the number of difference vectors to be used. Finally, z determines the crossover scheme. The one used in the description in

the previous section is independent binomial experiments denoted as 'bin', but exponential crossover 'exp' is also possible (Das and Suganthan, 2011).

PaGMO, the optimisation toolbox by ACT of ESA, exploits a variety of schemes for DE and the main schemes are taken into consideration for testing the algorithm. The selected schemes use random or the best vector for mutation and one or two weighted difference vectors. This gives five different schemes with different mutation vectors which come in the flavour of binomial or exponential crossovers, making a total of ten schemes. Below the schemes are listed with their mutation vector calculation.

1. *best/1/exp* and *best/1/bin*

$$\boldsymbol{v}_{i,G+1} = \boldsymbol{x}_{r^i_{best},G} + F \cdot \left(\boldsymbol{x}_{r^i_1,G} - \boldsymbol{x}_{r^i_2,G}\right) \tag{7.4}$$

2. *rand/1/exp* and *rand/1/bin*

$$\boldsymbol{v}_{i,G+1} = \boldsymbol{x}_{r^i_1,G} + F \cdot \left(\boldsymbol{x}_{r^i_2,G} - \boldsymbol{x}_{r^i_3,G}\right) \tag{7.5}$$

3. *rand-to-best/1/exp* and *rand-to-best/1/bin*

$$\boldsymbol{v}_{i,G+1} = \boldsymbol{x}_{r^i_1,G} + F \cdot \left(\boldsymbol{x}_{r^i_{best},G} - \boldsymbol{x}_{r^i_1,G}\right) + F \cdot \left(\boldsymbol{x}_{r^i_2,G} - \boldsymbol{x}_{r^i_3,G}\right) \tag{7.6}$$

4. *best/2/exp* and *best/2/bin*

$$\boldsymbol{v}_{i,G+1} = \boldsymbol{x}_{r^i_{best},G} + F \cdot \left(\boldsymbol{x}_{r^i_1,G} - \boldsymbol{x}_{r^i_2,G}\right) + F \cdot \left(\boldsymbol{x}_{r^i_3,G} - \boldsymbol{x}_{r^i_4,G}\right) \tag{7.7}$$

5. *rand/2/exp* and *rand/2/bin*

$$\boldsymbol{v}_{i,G+1} = \boldsymbol{x}_{r^i_1,G} + F \cdot \left(\boldsymbol{x}_{r^i_2,G} - \boldsymbol{x}_{r^i_3,G}\right) + F \cdot \left(\boldsymbol{x}_{r^i_4,G} - \boldsymbol{x}_{r^i_5,G}\right) \tag{7.8}$$

Das and Suganthan (2011), Storn and Price (1997) and Price (1996) suggested to use the ten different DE schemes above. Also Musegaas (2012) tried the five exponential crossover schemes for extensive tuning of his problems. None of the authors could find a scheme that worked best for all their problems. The general notion is that the scheme is too much problem-dependent.

Whereas having one difference vector is more considered the classic DE variant, the schemes using two difference vectors are noteworthy according to Price (1996). Also Storn and Price (1997) referred to cases of large population sizes NP where the diversity of the population is improved by using two difference vectors.

Since the performance of each scheme is problem-dependent it is difficult to state which schemes are preferred. Therefore a full scale testing is performed in the next chapter, to test the performance of the DE schemes applied to the low-thrust problems using the spherical shaping method by Roegiers (2014) and Novak and Vasile (2011) combined with gravity-assist manoeuvres.

Although PaGMO supplies more schemes, only these ten are selected, because the adaptive variant of DE, where F and CR are set during evolution, comes with the same schemes as the ones listed above. This allows for comparison on two aspects. First, the performance of the schemes is compared. Here the performance is defined as a combination of quality, computational effort (i.e. function evaluations, CPU time, number of generations for convergence) and robustness. Secondly, the performance of the adaptive control parameters can be compared

with the manually selected parameters. The adaptive DE method will be discussed in one of the next sections.

7.3 Control Parameters and Population Size

DE is a simple method that only requires three control variables to tune the algorithm (F, CR and NP). Consequently of its simplicity the algorithm requires a limited amount of coding resulting in fast execution of the single steps. Furthermore, DE is designed for use in parallel computing which is a necessity for very complex problems requiring many (objective function) computations. Also, only a limited number of NP objective function evaluations are performed per generation of DE.

The ranges for the control parameters were mentioned earlier in this chapter and are respectively $F > 0$ and $CR \in [0, 1]$. According to Das and Suganthan (2011) the range for F lies typically in $\in [0.4, 1]$. For CR there are different options, like $CR \in [0, 0.2]$ for objective functions that are separable and $CR \in [0.9, 1]$ for objective functions for which the parameters are dependent. Also the range $CR \in [0.3, 1]$ is a plausible option according to Das and Suganthan (2011).

The selection of the number of individuals (population) and the number of generations is still to be selected for the algorithms. According to Storn and Price (1997) a usually good range for population size is between 5 D to 10 D. Here D is the dimension of the problem (number of decision variables). On the other hand recent authors (according to Das and Suganthan (2011)) state sizes between 3 D to 8 D. Again, the problem has influence on what population size is most suitable and therefore a population size test is added to the benchmark in the next chapter as well. Also the number of generations is problem dependent and on how well the problem converges. However, a maximum number of generations can be set for which the optimisation should evolve the population in case convergence is not met. These maximums are defined per problem and are discussed in the following chapters.

7.4 Adaptive Control

One of the main disadvantages of DE and at the same time for the whole class of stochastic optimisation methods is that obtaining the global optimum is not guaranteed. Also the algorithm could get stuck in a local optimum. Furthermore, stochastic methods can be applied to a wide variety of problems, however, the performance is often problem-dependent. In case of DE, the parameters are very sensitive (Gämperle et al., 2002). Therefore tuning of the control parameters to the problem at hand is required which is a time-consuming procedure.

An alternative to tuning the control parameters of the DE algorithm manually, is to let these parameters be defined and altered by the evolution itself. In this way the time-consuming fine-tuning of the control parameters is avoided. On top of that, the control parameters are not constant during the full evolution of the population and can therefore utilise the strength of certain combinations of parameters depending on where in the evolution the population is evolving. If convergence is needed, the control parameters will be adapted to allow for swift convergence and therefore not allowing many new random individuals that will slow down the convergence. However, self-regulating control parameters might not be feasible for very time-consuming problems (Brest et al., 2006).

Both tuning the control parameters as well as choosing a suitable scheme for the problem at hand is very important according to Price (1996) and Musegaas (2012). Finding the optimal

settings requires multiple runs and can be time consuming and therefore adaptive is considered to overcome this (Brest et al., 2006). Note that adaptive requires more computations for determining new F and CR values, which in turn should improve the optimisation and therefore also save computational time. In the next chapter the tuning of the DE algorithm is further elaborated.

PaGMO comes with an adapted version of the DE algorithm called jDE. Two flavours are present at the time, meaning two different schemes to apply as adaptive control. The first scheme is based on Brest et al. (2007) and Brest et al. (2009) and uses the following equations to adapt the control parameters F and CR.

$$\boldsymbol{F}_{i,G+1} = \begin{cases} F_l + \text{rand}_1 \cdot F_u & \text{if } \text{rand}_2 < \tau_1, \\ \boldsymbol{F}_i & \text{otherwise} \end{cases} \tag{7.9}$$

$$\boldsymbol{CR}_{i,G+1} = \begin{cases} \text{rand}_3 & \text{if } \text{rand}_4 < \tau_2, \\ \boldsymbol{CR}_{i,G} & \text{otherwise} \end{cases} \tag{7.10}$$

For each individual a set of F and CR values is computed for each generation. Here $rand_j$, $j \in [1, 2, 3, 4]$ are uniform random values $\in [0, 1]$ and τ_1 and τ_2 represent the probabilities of adjusting the control parameters F and CR or keeping the control values of the previous generation. The probability values are given the value 0.1. Furthermore, the new F values are determined inside a range defined by F_l and F_u and by default are between 0.1 and 0.9. The CR values are completely defined in the range $\in [0, 1]$. The values are randomly initialised at the beginning of the evolution.

The second adaptive version is from the mind of Elsayed et al. (2011). He employs a similar computation for the control parameters as with the mutation vectors in the DE schemes. For each scheme there is a specific mutation of both control parameters. Below the adaptive scheme for each DE scheme is given.

1. *best/1/exp* and *best/1/bin*

$$\boldsymbol{F}_{i,G+1} = \boldsymbol{F}_{r^i_{best},G} + \text{rand}_1 \cdot 0.5 \cdot \left(\boldsymbol{F}_{r^i_1,G} - \boldsymbol{F}_{r^i_2,G}\right) \tag{7.11}$$

$$\boldsymbol{CR}_{i,G+1} = \boldsymbol{CR}_{r^i_{best},G} + \text{rand}_2 \cdot 0.5 \cdot \left(\boldsymbol{CR}_{r^i_1,G} - \boldsymbol{CR}_{r^i_2,G}\right) \tag{7.12}$$

2. *rand/1/exp* and *rand/1/bin*

$$\boldsymbol{F}_{i,G+1} = \boldsymbol{F}_{r^i_1,G} + \text{rand}_1 \cdot 0.5 \cdot \left(\boldsymbol{F}_{r^i_2,G} - \boldsymbol{F}_{r^i_3,G}\right) \tag{7.13}$$

$$\boldsymbol{CR}_{i,G+1} = \boldsymbol{CR}_{r^i_1,G} + \text{rand}_2 \cdot 0.5 \cdot \left(\boldsymbol{CR}_{r^i_2,G} - \boldsymbol{CR}_{r^i_3,G}\right) \tag{7.14}$$

3. *rand-to-best/1/exp* and *rand-to-best/1/bin*

$$\begin{aligned} \boldsymbol{F}_{i,G+1} = {} & \boldsymbol{F}_{r^i_1,G} + \text{rand}_1 \cdot 0.5 \cdot \left(\boldsymbol{F}_{r^i_{best},G} - \boldsymbol{F}_{r^i_1,G}\right) \\ & + \text{rand}_2 \cdot 0.5 \cdot \left(\boldsymbol{F}_{r^i_2,G} - \boldsymbol{F}_{r^i_3,G}\right) \end{aligned} \tag{7.15}$$

$$\begin{aligned} \boldsymbol{CR}_{i,G+1} = {} & \boldsymbol{CR}_{r^i_1,G} + \text{rand}_3 \cdot 0.5 \cdot \left(\boldsymbol{CR}_{r^i_{best},G} - \boldsymbol{CR}_{r^i_1,G}\right) \\ & + \text{rand}_4 \cdot 0.5 \cdot \left(\boldsymbol{CR}_{r^i_2,G} - \boldsymbol{CR}_{r^i_3,G}\right) \end{aligned} \tag{7.16}$$

4. *best/2/exp* and *best/2/bin*

$$\begin{aligned}\boldsymbol{F}_{i,\,G+1} &= \boldsymbol{F}_{r^i_{best},\,G} + \text{rand}_1 \cdot 0.5 \cdot \left(\boldsymbol{F}_{r^i_1,\,G} - \boldsymbol{F}_{r^i_2,\,G}\right) \\ &\quad + \text{rand}_2 \cdot 0.5 \cdot \left(\boldsymbol{F}_{r^i_3,\,G} - \boldsymbol{F}_{r^i_4,\,G}\right) \qquad (7.17)\\ \boldsymbol{CR}_{i,\,G+1} &= \boldsymbol{CR}_{r^i_{best},\,G} + \text{rand}_3 \cdot 0.5 \cdot \left(\boldsymbol{CR}_{r^i_1,\,G} - \boldsymbol{CR}_{r^i_2,\,G}\right) \\ &\quad + \text{rand}_4 \cdot 0.5 \cdot \left(\boldsymbol{CR}_{r^i_3,\,G} - \boldsymbol{CR}_{r^i_4,\,G}\right) \qquad (7.18)\end{aligned}$$

5. *rand/2/exp* and *rand/2/bin*

$$\begin{aligned}\boldsymbol{F}_{i,\,G+1} &= \boldsymbol{F}_{r^i_1,\,G} + \text{rand}_1 \cdot 0.5 \cdot \left(\boldsymbol{F}_{r^i_2,\,G} - \boldsymbol{F}_{r^i_3,\,G}\right) \\ &\quad + \text{rand}_2 \cdot 0.5 \cdot \left(\boldsymbol{F}_{r^i_4,\,G} - \boldsymbol{F}_{r^i_5,\,G}\right) \qquad (7.19)\\ \boldsymbol{CR}_{i,\,G+1} &= \boldsymbol{CR}_{r^i_1,\,G} + \text{rand}_3 \cdot 0.5 \cdot \left(\boldsymbol{CR}_{r^i_2,\,G} - \boldsymbol{CR}_{r^i_3,\,G}\right) \\ &\quad + \text{rand}_4 \cdot 0.5 \cdot \left(\boldsymbol{CR}_{r^i_4,\,G} - \boldsymbol{CR}_{r^i_5,\,G}\right) \qquad (7.20)\end{aligned}$$

Also for this adaptive version each individual has a set of F and CR computed for each generation. Here $rand_j$, $j \in [1, 2, 3, 4, 5]$ are uniform random values $\in [0, 1]$. The calculation for the control parameters is exactly the same as for the mutation vector, except that control parameters F and CR are computed instead of the velocity vector. Furthermore, the difference vector is multiplied with a uniform random value in the range [0, 0.5].

According to Brest et al. (2006) the adaptive versions make use of good F and CR combinations during the evolution that were also found by extensive tuning of the normal DE for the same problem. This advantage makes the adaptive DE less problem dependent, because the better combinations of F and CR are found during the optimisation. For the research in optimisation techniques, where the problem dimension (number of legs and GA's) are different, this can be a considerable advantage, because time is saved on time-expensive tuning. Note that tuning the population size remains a necessary process as it is for the normal DE algorithm. The next chapter will deal with the tuning and thoroughly testing the control parameters on two different problems.

8

Trajectory Leg Optimisation

Optimisation knows many settings and options and can become a very time-consuming process, specifically for complex problems. Also the model that describes the problem needs to be validated in order to trust the results that are coming from the optimiser. Therefore this chapter deals with the benchmarking of two different trajectory problems defined by Novak (2012) to compare the results and to fine tune the settings required to have smooth optimisation of the actual GTOC6 problem in the next chapters. First the two different trajectory problems are given. This will be followed by the test case of DE and adaptive DE. Due to problem dependency and sensitivity to the combination of control parameters and schemes the best algorithm and scheme are chosen after the full test case. This is followed by determining the optimal population size. The algorithm and settings that follow from this chapter will be used for the GTOC6 optimisation problem.

8.1 Trajectory Problems

The test suite for the control parameters for DE and the adaptive variants consists of two different problems. One problem is a single low-thrust trajectory from Earth to the comet Tempel 1. The second one is a low-thrust trajectory from Earth to the asteroid Apollo with a gravity-assist manoeuvre at Earth. Both problems are taken from Novak (2012) and one of the main reasons for that is that they are also solved with the spherical shaping method. The latter allows for better comparison of the results, even though implementation and problem definition differences can still influence the results. All problems are defined in the ecliptic and mean equinox of J2000.0 reference frame.

8.1.1 Earth - Tempel 1

The first test problem is a single low-thrust transfer from Earth to the comet Tempel 1. In this case there is no gravity-assist manoeuvre involved and the target is an inclined and eccentric orbit, which provides a challenge for the spherical shaping code. Note that the spacecraft is rendezvousing with the comet and that at departure and arrival no velocity impulse is given (i.e. $V_\infty = 0.0\,\mathrm{m/s}$).

Furthermore, Roegiers (2014) performed the optimisation of several low-thrust trajectories using grid search including the comet problem. In theory, similar results are to be expected with the problems from Novak (2012). However, Roegiers (2014) noticed slight deviations in the results which she contributed to differences in the implementation of the shaping method and to settings used for various computations during the spherical shaping of the trajectory. Among the settings are constants (e.g. gravitational parameters, AU) and ephemerides of the celestial bodies. Also for this problem her result ($\Delta V = 11.51\,\text{km/s}$) deviated considerably from the solution ($\Delta V = 11.13\,\text{km/s}$) of Novak (2012). For the validation of the optimisation model a value for the true anomaly has been assumed, see Table 8.2. However, the true anomaly could be introduced as an extra decision variable in the optimisation model to find the true anomaly angle that solves the discrepancy with the solution of Novak (2012).

The low-thrust problem is defined with three decision variables, namely the start epoch, the time-of-flight and the number of revolutions. As input for the optimisation, the same values used by Novak (2012) and Roegiers (2014) are applied, see Table 8.1. The population size used for the optimisation by Novak (2012) is $10D$, where D is the decision vector size. Accordingly, a total of 30 individuals make up the population.

The orbital elements of the comet Tempel 1 are given in Table 8.2. Note that the sixth Kepler element is not given in both papers and therefore it is assumed that the actual position of the comet in the orbit is given by NASA's small body database. This element could be one reason for the difference in ΔV in the solutions of Roegiers (2014) and Novak (2012).

Table 8.1: *Decision vector for the Earth - Tempel 1 problem.*

Variable	Unit	Lower bound	Upper bound
Epoch	[MJD]	51544.0	57390.0
Time-of-flight	[days]	400.0	1500.0
$N_{revolutions}$	[-]	0	2

Table 8.2: *Orbital elements of comet Tempel 1 (Novak, 2012).*

Element	Unit	Apollo
a	[AU]	3.124
e	[-]	0.517
i	[°]	10.527
Ω	[°]	68.933
ω	[°]	178.926
θ[1]	[°]	0.0

[1] Assumed value for true anomaly at epoch 57 602.018 MJD (Park and Chamberlin, 2015)

The Tempel 1 problem has only three decision variables and therefore allows to create a pork chop plot of the solution space. It gives insight into the problem and whether the optimisation algorithm is searching in the correct areas. The result is displayed in Figure 8.1. In principle three different pork chops could be created, one for each number of revolutions. However, the pork chop in Figure 8.1 is made in the same way as done by Novak (2012) and Roegiers (2014) to be able to compare the results. Both authors stacked the individual pork chops on top of each other, allowing for the optimal result of the three layers to be shown.

The main goal of this optimisation problem is to achieve the value of $\Delta V = 11.51\,\text{km/s}$ (with epoch $= 55\,669.0\,\text{MJD}$, $TOF = 1460\,\text{days}$ and $N_{revolutions} = 0$) acquired by Roegiers (2014) or even better. Due to the nature of the grid search that she applied it is possible that by zooming in on the grid a better optimum can be found near one of the grid points that she

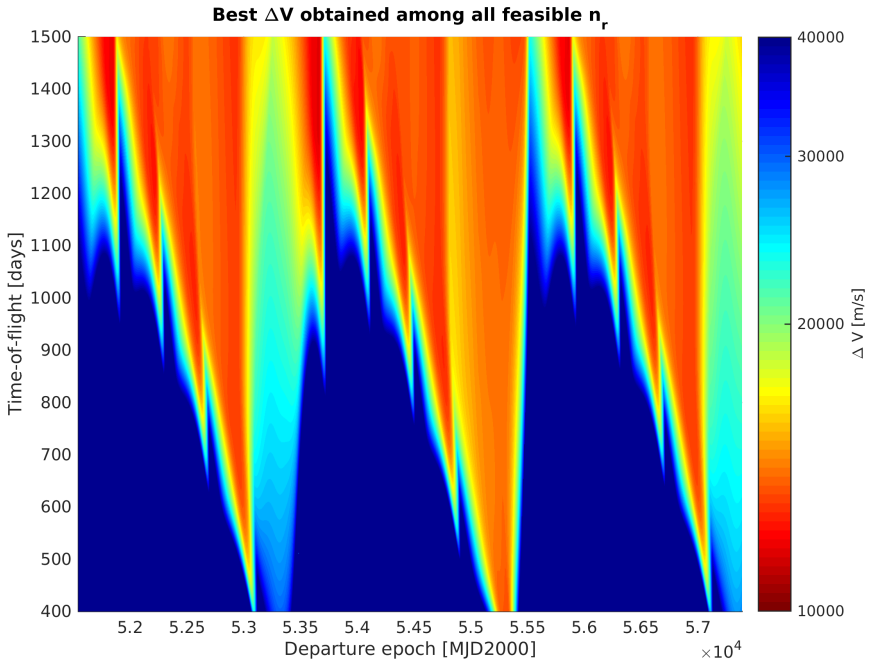

Figure 8.1: *Pork chop plot of the low-thrust trajectory from Earth to Tempel 1. The graph is the optimal result of stacking three individual pork chops on top of each other for* $n_r \in [0, 2]$.

computed as optimal. Unfortunately Novak (2012) did not mention the TOF or the number of revolutions for his optimal solution.

8.1.2 Earth - Earth - Apollo

The second problem is a low-thrust gravity-assist transfer from Earth to the asteroid Apollo. Here the spacecraft starts at Earth, performs a low-thrust arc to come back to Earth to perform a flyby and then continues with the last thrust arc to get to the asteroid. Due to the large eccentricity of the Apollo orbit the gravity-assist manoeuvre becomes necessary to keep the ΔV of the thrust arcs low. This is also preferred for the GTOC6 problem and therefore this makes a great test problem. Note that at Earth the spacecraft gets a velocity impulse which is not included in the optimisation ΔV. This trajectory problem with flybys by Novak (2012) is not performed by Roegiers (2014).

The low-thrust gravity-assist problem is defined with 13 decision variables which are given in Table 8.3. Here the angles α and β define the orientation respectively in-plane and out-of-plane of the hyperbolic excess velocity $V_{-\infty}$. $h_{flyby,1}$ and $\theta_{rot,1}$ are respectively the altitude and the rotation angle of the flyby manoeuvre. The $V_{-\infty,1}$ and corresponding orientation angles define the hyperbolic excess velocity before the gravity assist. This is needed because the spherical shaping method handles rendezvous conditions at both the departure and arrival node. Therefore the arrival velocity at the flyby needs to be known in order for the shaping method to create a trajectory. Note the middle block of seven decision parameters that defines one low-thrust arc with a gravity-assist manoeuvre attached to it. This block can be used iteratively as will be shown for the GTOC6 problem.

The actual decision vector and model used by Novak (2012) for optimisation are not known. However, the boundary conditions for time-of-flight and maximum $V_{-\infty}$ were partially given

and can also be found in Table 8.3. Furthermore, the population size used for the optimisation by Novak (2012) is again $10D$, making a total of 130 individuals as population.

Table 8.3: *Decision vector for the Earth - Earth - Apollo problem.*

Variable	Unit	Lower bound	Upper bound
Epoch	[MJD]	55197.0	57023.0
$V_{-\infty,0}$	[m/s]	0.0	5000.0
α_0	[°]	0.0	360.0
β_0	[°]	-90.0	90.0
TOF_1	[days]	200.0	800.0
$N_{rev,1}$	[-]	0	1
$V_{-\infty,1}$	[m/s]	0.0	10000.0
α_1	[°]	0.0	360.0
β_1	[°]	-90.0	90.0
$h_{flyby,1}$	[km]	200.0	10000.0
$\theta_{rot,1}$	[°]	-180.0	180.0
TOF_2	[days]	200.0	1000.0
$N_{rev,2}$	[-]	0	1

The orbital elements of the asteroid Apollo are given in Table 8.4. Again Novak (2012) did not supply the true anomaly element and therefore a value was assumed, see Table 8.4.

Table 8.4: *Orbital elements of asteroid Apollo (Novak, 2012).*

Element	Unit	Apollo
a	[AU]	1.471
e	[-]	0.56
i	[°]	6.4
Ω	[°]	25.9
ω	[°]	285.7
θ^1	[°]	0.0

[1] Assumed value for true anomaly at epoch 54 933.135 MJD (Park and Chamberlin, 2015)

An important remark on this solution is that the model by Novak (2012) used a coasting arc to go from Earth to the gravity-assist manoeuvre at Earth. The model used here has two low-thrust arcs, however it should theoretically be possible to aim for a coasting arc, for the first thrust leg, with the spherical shaping method due to the RFT fix. Furthermore, Novak (2012) applies extensive space pruning before optimisation. The resulting hyper-spaces with promising search areas are to be optimised with DE. Another important remark, is the used pruning method GASP. The method creates Lambert arcs and uses powered swing-bys to allow for independent legs in the problem. Taking these considerations into account, it is clear that comparing the solutions will be more difficult.

Also the details on the final result are minimal and only the total ΔV solution is given to the problem which has a value of 10.25 km/s. Here 4.93 km/s is for the launch velocity at Earth and 5.32 km/s is for the low-thrust arc. The total solution is a combination of low and high thrust and because no further details are given on the optimisation model, it is not known whether the trajectory is optimised for only the low-thrust arc or a combination of low and high thrust. Therefore, it is assumed that both low and high thrust are optimised and that a solution in the same order of magnitude of Novak (2012) will be considered adequate to validate the model. More on the this is discussed in the next section with the results.

8.1.3 Initial Results & Validation

Both problems were tested before starting on the complete benchmark. This was done using the adaptive DE (jDE) algorithm using the default settings (DE/rand/1/exp and adaptive scheme by Brest et al. (2009)). The population size was chosen similar to that of Novak (2012), namely $10D$, to allow for more compatibility with his settings for the optimisation.

For the Tempel 1 problem a maximum of 500 generations was set whereas for the Apollo problem 2500 generations was the maximum. Convergence is defined by the built-in algorithm of PaGMO. It determines every 40 generations if the summation of difference in either the fitness variables or the decision variables decreased up till a certain predefined limit. This limit was left to be the default value of PaGMO which is set to 1.0×10^{-6}.

The initial results for the Tempel 1 problem provided the same solution for every repeated optimisation run (10x) and are given in Table 8.5. From the results it is clear that the solution is in the neighbourhood of and slightly improved compared to the solution of Roegiers (2014). Note that the latter performed a grid search, however, the solution found is very close to her solution. Therefore this problem is considered verified.

Table 8.5: *Initial results Earth - Tempel 1 problem.*

Variable	Unit	Hoving	Novak	Roegiers
ΔV	[km/s]	11.495[1]	11.13	11.51
Epoch	[MJD]	53664.3	-	55669.0
Time-of-flight	[days]	1497.39	-	1460
$N_{revolutions}$	[-]	0	-	0

[1] 10 out of 10 times.

The initial results for the Apollo problem provided not the same solution for every repeated optimisation run (10x) and are given in Table 8.5. Because the model of Novak (2012) is different, the same result is less likely to happen. However, in this case the result is consistently and considerably better. This can be related to the model difference and the space pruning that could have skipped uninteresting areas which did include promising values. Note that this is the case for optimisation of both low and high thrust. Furthermore, the results from the optimisation were verified with numerical propagation and are therefore considered valid solutions.

Table 8.6: *Initial results Earth - Earth - Apollo problem.*

Variable	Unit	Hoving	Novak
Total ΔV	[m/s]	8764.29[1]	10250.00
Total ΔV	[m/s]	8777.34[2]	10250.00
Total ΔV	[m/s]	8797.16[3]	10250.00

[1] 2 out of 10 times.
[2] 7 out of 10 times.
[3] 1 out of 10 times.

The difference with the solution of Novak (2012) is considerable. Taking a closer look, the high-thrust impulse at departure is $V_{+\infty} = 601.21\,\text{m/s}$ and the two low-thrust arcs have respectively $\Delta V = 3271.15\,\text{m/s}$ and $\Delta V = 4891.93\,\text{m/s}$. It is clear that the shaping method did not find the coasting arc. Furthermore, the hyperbolic excess velocity at departure is relatively low compared to the value that Novak (2012) achieved. The low value indicates that the thrust legs and flyby have to compensate to find a suitable trajectory to rendezvous with Apollo. Unfortunately Novak (2012) did not provide information on the flyby or thrust leg which could have been helpful to identify the culprit in the optimisation problem here. However, it is clear that the optimisation models are not the same. An attempt was made

to only optimise for ΔV of the two low-thrust legs, though, this resulted in even worse ΔV for the low-thrust legs. Therefore the result shown above with ΔV containing both high and low thrust is used. Another point that has to be kept in mind is the unknown value for the true anomaly used by Novak (2012). This can make a considerable difference as well besides the model differences. Although the low-thrust gravity-assist model has no matching solution with Novak (2012), and therefore can not be said to be validated, it does result in proper trajectories that are numerically verified. The combined objective (high and low thrust) does prove to have considerable better results and the thrust leg of Novak (2012) does show similar values with the second thrust leg mentioned above. Still the optimisation problems are different due to the coasting arc that causes a higher hyperbolic excess velocity at departure.

A clear result from this problem is that the increasing dimension of the decision vector and adding the gravity-assist causes the optimiser to have more difficulty with convergence. Although several trials have been performed, so far the most optimal solution is 8764.29 m/s. However, DE is known to get stuck in local optima, as is shown with the results here. Also it is never guaranteed with DE that the optimum has been found. Therefore, for the upcoming test suite, the current best value of 8764.29 m/s is used as aim value for the optimisation process.

8.2 Test Suite - Adaptive versus normal DE

With the optimisation goals set for the two problems and the models verified and validated the tests for DE schemes and the adaptive variants can be performed. The variables for the test suite are the schemes and the F and CR control parameters in case of the normal DE algorithm.

The values for F and CR are in the range of $\in [0.2, 1.0]$ with steps of 0.2. A grid of 25 cases is the result. Each case is repeated 10 times to verify the robustness of the algorithms. Musegaas (2012) performed a more thorough analysis by increasing the number of repeats in the order of hundreds to allow for statistical analysis. He used it to analyse his optimisation toolbox, which was to be used for various problems. However, this is considered too extensive and unnecessary for the goal of this thesis. A repeat in the order of hundreds would increase the computation time enormously to out of proportions for this thesis subject. This will become clear with the results. For the adaptive DE only the scheme and the adaptive variant are variables.

The results of the optimisation can be found in Appendix C. Before discussing the results it was noticed that the adaptive algorithm was implemented incorrectly for the test suite. Due to recording of the data of the optimisation the evolution method was repeatedly called. This caused the control parameters to be initialised with random values each time a new generation was being made, instead of taking the values of the previous generation into account. However, it turned out that this new approach, which in definition is not adaptive, worked out better than the actual adaptive methods. The results of the incorrect implementation of jDE with adaptive variant 1 are shown in Figure 8.2. An explanation for this improved behaviour due to the incorrect implementation is given later on in this section.

In Appendix C the correct versions of the adaptive method have been given. For each figure the settings for F and CR are found on the x- and y-axis in case of normal DE. In case of adaptive DE the 10 schemes applied are shown on the axis. The crossover scheme is set on the x-axis whereas the selection and mutation schemes can be found on the y-axis. The colour shows the number of occurrences (i.e. percentage) that the threshold has been passed for that

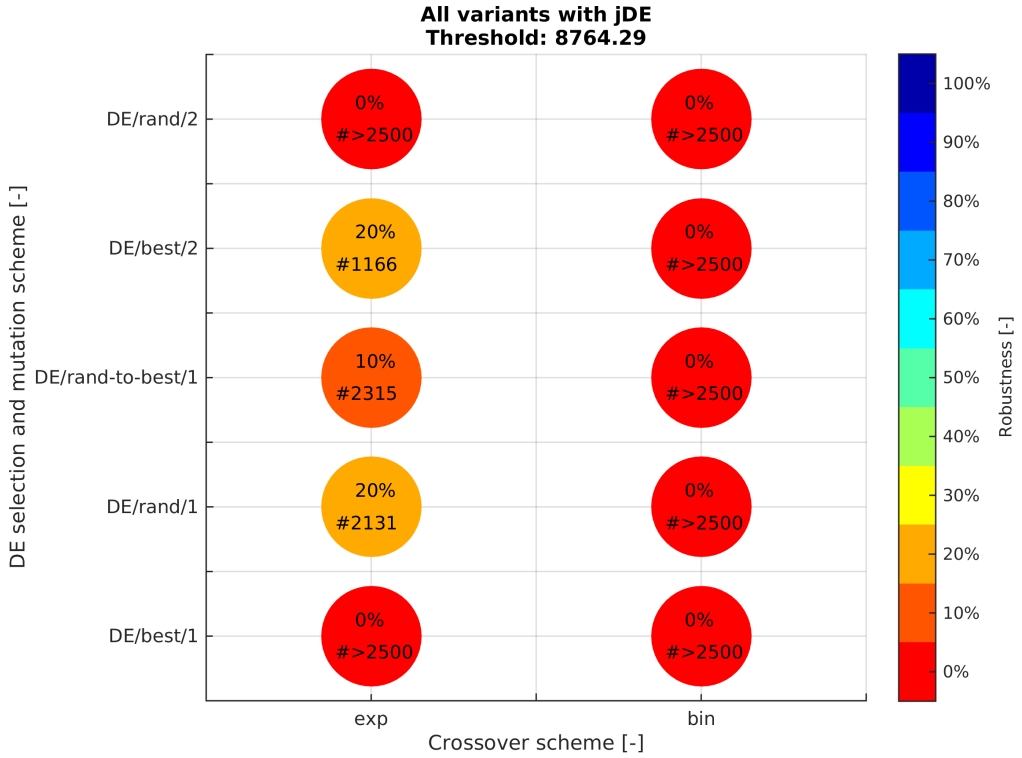

Figure 8.2: *Optimisation tuning results for the incorrect jDE (adaptive, scheme 1). Threshold for* $\Delta V = 8764.29\,\mathrm{m/s}$.

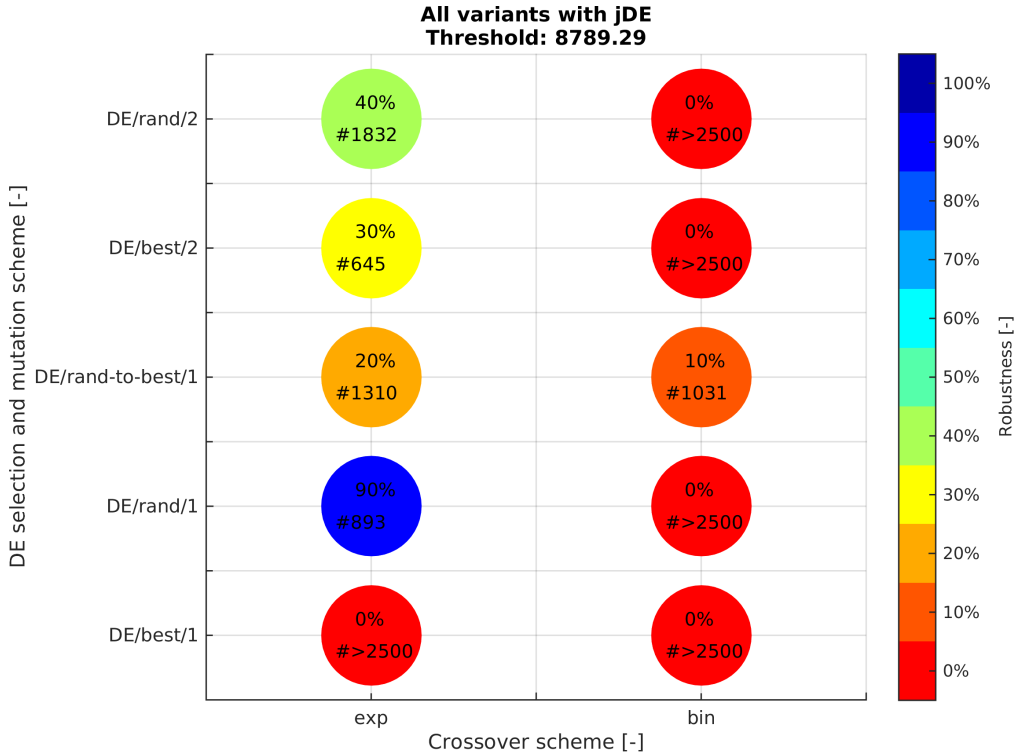

Figure 8.3: *Optimisation tuning results for the incorrect jDE (adaptive, scheme 1). Threshold for* $\Delta V = 8764.29 + 25.0\,\mathrm{m/s}$.

setting and is an indication of the robustness of the tested algorithm. Inside the coloured circles there are two values. The top one equals the previous percentage value in case the colour is not clear. The bottom value indicates the number of generations until the threshold value has been reached. This threshold value is either set to the expected optimum or the that same with a certain deviation added.

From the colourful results two major findings can be made. First the expected optimum value of 8764.29 m/s is barely found at all. Only the schemes *rand/1/bin* and *rand/1/exp* show a few occurrences for normal DE. In case of adaptive also *rand/1/exp* and *best/2/exp* show potential. However, this is the case for the incorrect adaptive variant, the actual adaptive versions show no success there. Secondly the normal DE schemes with manual and constant control parameters only have a single few results. This makes it really hard to select a proper scheme with suitable control parameters. Although also the incorrect adaptive variant does not display conclusive results.

The expected optimum appears to be very difficult to find. Therefore the threshold has been increased by a relatively small step of 25 m/s. The result for the incorrect adaptive DE is displayed in Figure 8.3 whereas the other results can be found again in Appendix C. With the new threshold the optimisation schemes look more promising. Especially the *rand/1/exp* scheme is performing well for this problem, for both the incorrect adaptive and the fixed DE algorithm. It appears that the gravity-assist problem favours the scheme where random selection is performed on the individuals over selecting the best individual for mutation. This can be explained by the fact that the continuous low-thrust arcs allow for many possibilities with slight alterations to the variables. Also the gravity-assist manoeuvre allows for a considerable amount of flexibility. The combination of the two trajectory elements allow for many if not infinitely many possibilities and therefore it is very likely that the solution space consists of many local optima. On top of that small changes in the decision vector allow for considerable varying solutions that change quickly from being optimal solutions to worse individuals that will slow down the optimisation process.

To get a picture of the optimisation process the champion values of the complete evolution are shown in Figures 8.4 and 8.5. For this the same schedule is selected for both algorithms to be able to assess the difference in random and fixed control parameters. For fixed DE the optimal setting is showing which is $F = 0.8$ and $CR = 0.8$. An interesting observation can be made from these evolutions of champions. The figures show for all ten repeats the evolution of the best individual in the current generation. What is clearly visible is that the incorrect adaptive method is considerably and consistently faster in convergence to the area close around the expected optimum value. Here it can be said that the use of non-constant control parameters during the evolution works to the advantage of this complex trajectory problem. Note, that a few schemes offered slightly improved results for DE with respect to convergence. However, non was able to get close to the incorrect adaptive version. Furthermore, that these findings are valid for this current low-thrust gravity-assist problem and that they are no general conclusion for all low-thrust gravity-assist problems. As mentioned by Novak (2012) and Musegaas (2012) numerous times the optimal DE parameters are very sensitive to the problem at hand.

Another benefit of the incorrect jDE algorithm is that there is no need to fine tune the control parameters. For the standard DE, however, the control parameters that gave a reasonable good performance followed after extensive testing of F and CR values. Therefore the incorrect adaptive method is chosen as the optimal optimisation algorithm for the GTOC6 problem with the *rand/1/exp* scheme in spite of its puzzling aspects. Next follows the selection of the most effective population size to reduce the number of function evaluations and therefore computation time, to improve the solutions even further. Note that all simulations in this

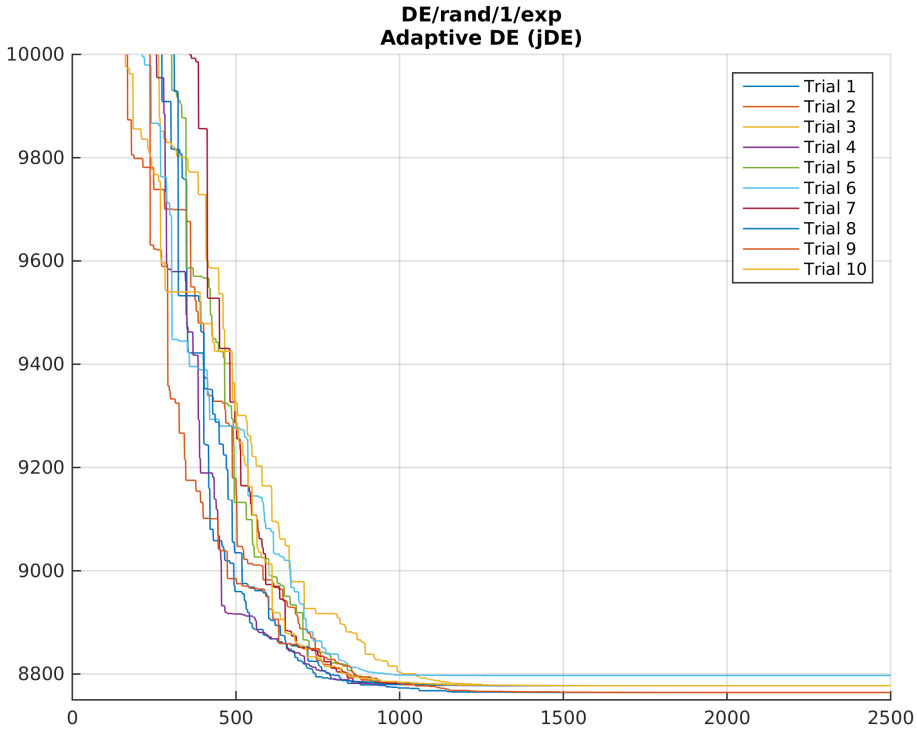

Figure 8.4: *Complete evolution of the champions of the population for the incorrect jDE (adaptive, scheme 1).*

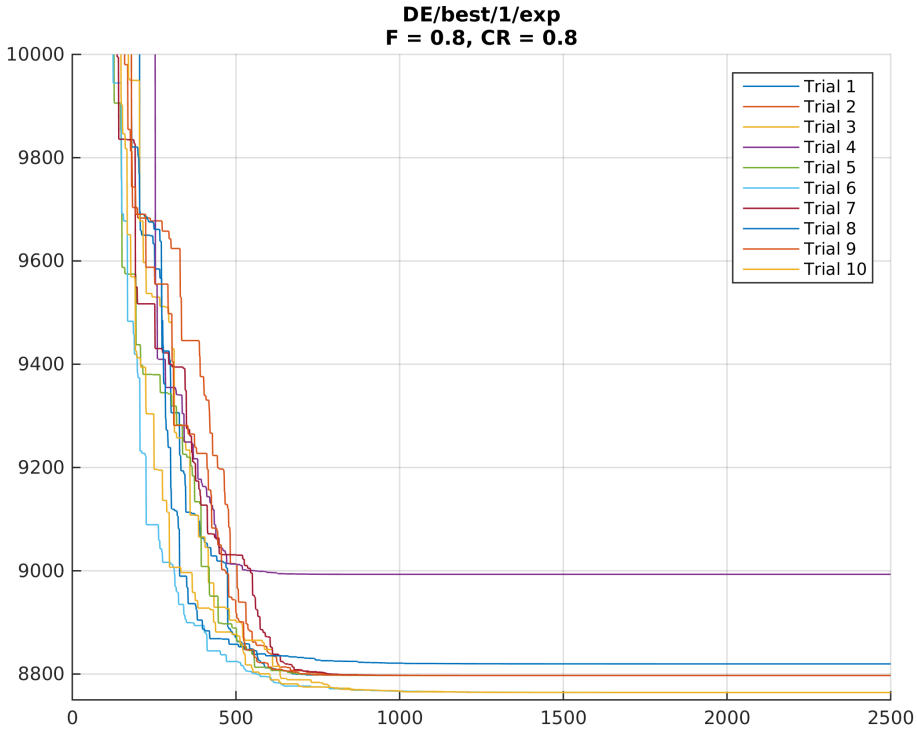

Figure 8.5: *Complete evolution of the champions of the population for the DE with rand/1/exp.*

section are performed with the same population size which might not work in favour of all tested schemes.

8.3 DE with Random Control Parameters

As mentioned in the previous discussion the adaptive method selected for the optimisation is actually not adaptive anymore. The official adaptive variant 1 turned out to be less effective compared to the incorrect implemented version. In principle the new algorithm is not wrong, it just does not take into account the control parameters of the previous generation. Therefore the equations to adapt the control parameters are written as follows.

$$\boldsymbol{F}_{i,G+1} = F_l + \text{rand}_1 \cdot F_u \tag{8.1}$$

$$\boldsymbol{CR}_{i,G+1} = \text{rand}_2 \tag{8.2}$$

Here $rand_j$, $j \in [1, 2]$ are uniform random values $\in [0, 1]$. The new F values are determined inside a range defined by F_l and F_u and by default are between 0.1 and 0.9. CR values are again completely defined in the range $[0, 1]$.

The new algorithm is a normal DE with randomly selected control parameters for each generation instead of constant parameters for the whole evolution. To avoid confusion about the naming of the algorithms, the incorrect adaptive method is from here on referred to as *randDE*.

8.4 Test Suite - Population Sizes

The final test suite before continuing with the GTOC6 problem is finding the optimal population size. Previous optimisations were performed with $10D$ individuals which seemed to work well and allowed for a better comparison with Novak (2012) who also used that number. Although various recommendations exist for population sizes as mentioned in Chapter 7, also problem dependency plays a role here. Therefore a complete range of population sizes is performed from $1D$ till $14D$. Furthermore, the number of trials is increased to 50 to give better insight into the consistency of the results.

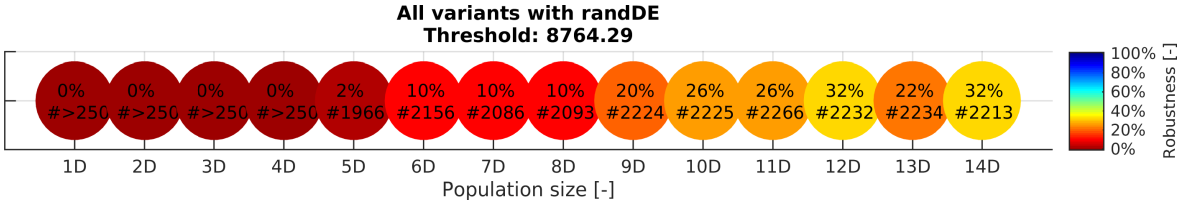

Figure 8.6: *Results for randDE (rand/1/exp) with population sizes* $\in [1D, 14D]$. *Threshold for* $\Delta V = 8764.29\,\text{m/s}$.

In Figure 8.6 the results are shown for when the threshold is set at the expected optimum. Comparing the total of trials that have reached the threshold, one can conclude that population sizes ranging $\in [1D, 8D]$ are not sufficient. From $9D$ onwards at least 20 % reached the optimum. Allowing for a deviation of 25 m/s on the previous threshold gives Figure 8.7. Though populations with $\in [6D, 8D]$ did not perform well before, they now do keep up very well with local optima close to the expected minimum.

A nearly constant result can be seen in the range $\in [9D, 14D]$. Apparently the increased population size does not add significant more better solutions and even for $13D$ the performance

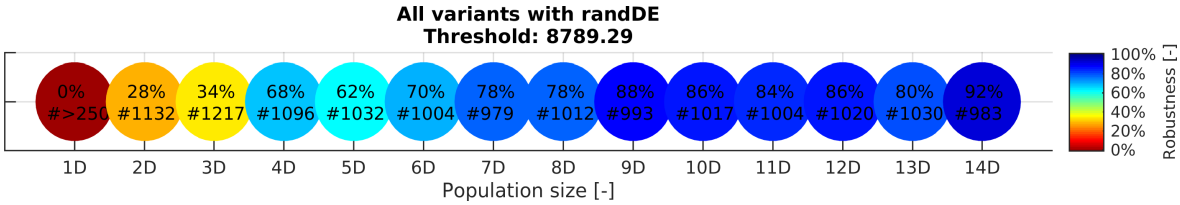

Figure 8.7: *Results for randDE (rand/1/exp) with population sizes* $\in [1D, 14D]$. *Threshold for* $\Delta V = 8764.29 + 25.0\,\mathrm{m/s}$.

drops slightly. Note that even for 50 trials it is hard to draw a significant statistical conclusion from these results. However, taken from these results it is in the best interest for the optimisation process to take a population size as small as possible (reduce computation time) while maintaining a considerable number of optimal solutions that are found consistently (robustness) by the optimiser. Combining the results of both figures gives the conclusion that a population size with $9D$ is the best compromise in terms of quality and quantity and computational effort. A final remark is that 44 (i.e. 88 %) trials are within in the $25\,\mathrm{m/s}$ deviation for $9D$.

9

Sequence Optimisation

THE low-thrust multiple gravity-assist trajectories considered in this chapter only deal with a relative small sub-selection of the whole sequence of the GTOC6 problem. However, the focus is not on solving the GTOC6 problem. Rather it is focussed on finding an optimal strategy to find optimal solutions to sub-sequences of the larger problem. In this chapter the model, the optimisation objective and its constraints are explained. Also the two sub-sequences that are selected from the CSU solution for optimisation are discussed.

9.1 The GTOC6 Low-Thrust Multiple Gravity-Assist Optimisation Model

The optimisation model used for tackling the GTOC6 sub-problems is based on the model used in the previous chapter. A few alterations were made to allow for the flexibility of adding as many gravity-assist manoeuvres as required. The model starts with a flyby at the first moon of the sequence and is followed by a user-defined number of low-thrust legs and gravity assists. Also the rendezvous element was removed. Despite the changes, no new decision variables were introduced. The template of the decision vector is given in Table 9.1.

The first six decision variables define the initial flyby (denoted by $_0$) of the sequence. Here epoch defines the start of the flyby sequence and also the time of the first flyby. The remaining five parameters describe the actual gravity assist. With the in-plane and out-of-plane angles, respectively α and β, the orientation of the hyperbolic excess velocity $V_{-\infty}$ is set. This is followed by the altitude h_{flyby} and the rotation angle θ_{rot}. Note that the entry conditions for the flyby ($V_{-\infty}$) are used instead of the exit conditions ($V_{+\infty}$). The reason for that is simply the flexibility and modularity of the code. Also no sphere of influence is used for the instantaneous flybys as defined by the GTOC6 constraints. Therefore the initial position of the sequence is set to the exact position of the first moon at the given epoch.

After that one can patch a low-thrust leg with a gravity-assist manoeuvre to the initial flyby. Here a total of seven decision variables are needed. Two describe the low-thrust arc characteristics time-of-flight TOF and number of revolutions N_{rev}. The remaining five describe again the next flyby. By adding another set of the seven parameters the sequence is

Table 9.1: *Template decision vector for low-thrust multiple gravity-assist trajectories with* $i \in [1, 2, ..., N_{flybys} - 1]$.

Variable	Unit	Lower bound	Upper bound
Epoch	[MJD]	strategy defined	strategy defined
$V_{-\infty,\,0}$	[m/s]	300.0[1]	3000.0[2]
α_0	[°]	0.0	360.0
β_0	[°]	-90.0	90.0
$h_{flyby,\,0}$	[km]	50.0[1]	2000.0[1]
$\theta_{rot,\,0}$	[°]	-180.0	180.0
For each flyby			
TOF_i	[days]	strategy defined	strategy defined
$N_{rev,\,i}$	[-]	0	1
$V_{-\infty,\,i}$	[m/s]	300.0[1]	3000.0[2]
α_i	[°]	0.0	360.0
β_i	[°]	-90.0	90.0
$h_{flyby,\,i}$	[km]	50.0[1]	2000.0[1]
$\theta_{rot,\,i}$	[°]	-180.0	180.0

[1] GTOC6 constraint.
[2] Boundary defined after global CSU solution analysis.

again increased by yet another segment consisting of a low-thrust arc and a flyby. The total number of decision variables can be computed with Equation (9.1).

$$N_{decision\,vars} = 7 \cdot N - 1 \qquad N \in [2, \infty] \tag{9.1}$$

$$N_{pop\,size} = 9 \cdot N_{decision\,vars} \tag{9.2}$$

The corresponding population size follows from Equation (9.2). A final remark for the decision vector template is that it requires a minimum of two flybys. This makes sense, because a single gravity-assist problem without a thrust leg is not useful.

9.2 Sequence selection

For the sequence optimisation test suite two different sub-sequences of five flybys are selected. The reason is not to focus on one specific subset, because that would lead to conclusions only relevant to that part of the main sequence. Therefore sub-sequences are taken from near the beginning and near the end of the CSU solution, allowing for both short-period and long period interaction in the moon sequences. The solution of CSU did not have sub-sequences that had a length of five gravity assists that have more than two different moon destinations in there. For example, Io is mapped nearly completely in one large sequence. This resulted in only two different moons being visited in the two sub-sequences.

An important note is that the CSU team used different techniques and therefore the solutions can never be fully matched or compared. But they are used as indication of what is possible and to stay in the neighbourhood and to give initial conditions with respect to time-of-flight and epochs. Also the remaining parameters of the sequence are used as initial condition, like the mass of the spacecraft at the start of the sequence.

9.2.1 Sequence 1: Europa, Europa, Europa, Io, Io

The first sub-sequence starts at flyby number 15 of the whole sequence and ends at the 19th gravity-assist manoeuvre. With an epoch of 59 880.709 MJD the sequence starts not much after the capture of the spacecraft into the Jovian system with the moon Europa, to be

continued with Europa, Europa, Io and finally Io again. Also the sequence is situated in the region where radiation is strong and thus more mass penalty is applied. However, also the high-score faces are available on Europa. Note that for pure ΔV optimisation this has no influence. Also the orbital periods and synodic periods are small compared to those of the next sequence.

Table 9.2: *Characteristics of the gravity-assist manoeuvres for the first sub-sequence Eu-Eu-Eu-Io-Io.*

# Flyby[1]	Moon	Epoch [MJD]	$V_{-\infty}$ [km/s]	h_{flyby} [km]	# Face	Face value	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59880.709	1.7625	65.5	23	6	1889.67
2 (16)	Eu	59884.262	1.7625	180.4	11	4	1884.32
3 (17)	Eu	59898.475	1.7625	592.0	12	4	1856.10
4 (18)	Io	59912.873	2.4985	1262.1	31	2	1821.93
5 (19)	Io	59921.730	2.4985	60.7	29	2	1797.62

[1] Number between parenthesis is the actual flyby number in the CSU solution.

The characteristics of the flybys and low-thrust legs of the CSU sequence are given respectively in Tables 9.2 and 9.4. Looking at the gravity assists one can see clearly the resonance pattern in the sequence where the spacecraft arrives with a similar hyperbolic excess velocity at the same moon. Note that just two consecutive flybys with the same excess velocity does not necessarily mean a resonance. The time-of-flight also needs to be an integer multitude of the orbital period in case of a transfer between the same moons. Also the moon can orbit for multiple integers of revolutions allowing for a variety of resonance patters (e.g. 1:1, 2:1, 3:2). Here the two integers are the number of orbits of respectively the moon and the spacecraft. Resonance is the case for the two Eu-Eu transfers which have time-of-flights that are an integer multitude of the orbital period of Europa (3.55 days). Therefore leg one and leg two are respectively 1:1 and 4:1 resonance transfers. Also the Io-Io transfer is a 5:1 resonance. In case of different moon destinations, the synodic period is used as base value for the resonance of the two moons and the spacecraft. In Table 9.3 an overview is given of the all orbital and synodical periods of the Galilean moons.

Table 9.3: *Orbital periods T and synodic periods T_{syn} for the Galilean moons.*

Moon	T [days]	Moons	T_{syn} [days]	Moons	T_{syn} [days]
Io	1.771	Io - Eu	3.553	Eu - Ga	7.056
Eu	3.553	Io - Ga	2.354	Eu - Ca	4.514
Ga	7.157	Io - Ca	1.982	Ga - Ca	12.527
Ca	16.696				

Combining the data with the low-thrust legs, it becomes obvious that the strategy of CSU was to find resonance patterns that made use of the flybys to do all of the work on changing the orbital parameters to arrive at the next moon.

Table 9.4: *Characteristics of the low-thrust legs for the first sub-sequence Eu-Eu-Eu-Io-Io.*

# Leg[1]	Moons	Epoch [MJD]	Arc type	TOF [days]	ΔV [m/s]	T_{max} [N]
1 (15-16)	Eu-Eu	59880.709	coasting	3.55	0.0	0.0
2 (16-17)	Eu-Eu	59884.262	coasting	14.21	0.0	0.0
3 (17-18)	Eu-Io	59898.475	coasting	14.40	0.0	0.0
4 (18-19)	Io-Io	59912.873	coasting	8.86	0.0	0.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.

One final remark is that there is no thrust used in this sequence. Although this means that no propellant has been used, there is still loss of mass due to mass penalties due to close proximity with Jupiter (radiation). For comparison of the spherical shaping accuracy in the test suite the orbital elements of the departure and arrival terminals are given in Table 9.5.

Table 9.5: *Orbital elements of the low-thrust legs of sub-sequence one.*

Element	Unit	Leg 1_d	Leg 1_a	Leg 2_d	Leg 2_a
a	[km]	671224.2	671224.2	578443.5	578443.5
e	[-]	0.135	0.135	0.194	0.194
i	[°]	1.9	1.9	0.4	0.4
Ω	[°]	238.1	238.1	75.3	75.3
ω	[°]	265.3	265.3	14.1	14.1
θ^1	[°]	97.9	97.9	151.9	151.9
Element	**Unit**	**Leg 3_d**	**Leg 3_a**	**Leg 4_d**	**Leg 4_a**
a	[km]	542738.8	542738.8	489722.1	489722.1
e	[-]	0.239	0.239	0.177	0.177
i	[°]	0.2	0.2	3.0	3.0
Ω	[°]	95.2	95.2	96.0	96.0
ω	[°]	331.7	331.7	312.9	312.9
θ^1	[°]	174.5	29.2	47.2	47.2

[1] True anomaly at epoch of departure of arrival

9.2.2 Sequence 2: Ganymede, Ganymede, Callisto, Callisto, Ganymede

The second sub-sequence starts at flyby number 103 of the whole sequence and ends at the 107th gravity-assist manoeuvre. With a start epoch of 60 510.155 MJD the sequence starts not much before the end of the sequence (total of 128 flybys) with the moon Ganymede, to be continued with Ganymede, Callisto, Callisto and Ganymede again. Contrary to sequence one, this sequence is situated in the region where radiation is less strong and thus less mass penalty is applied. However, also the lesser score faces are available on these moons. Also the orbital periods and synodic periods are large compared to those in the first sequence.

Table 9.6: *Characteristics of the gravity-assist manoeuvres for the second sub-sequence Ga-Ga-Ca-Ca-Ga.*

# Flyby[1]	Moon	Epoch [MJD]	$V_{-\infty}$ [km/s]	h_{flyby} [km]	# Face	Face value	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60510.155	1.6314	491.3	4	3	1098.51
2 (104)	Ga	60517.312	1.6314	186.6	10	2	1095.76
3 (105)	Ca	60524.864	1.6650	852.4	17	1	1092.87
4 (106)	Ca	60541.561	1.6650	790.5	23	1	1092.87
5 (107)	Ga	60545.921	1.6615	169.9	11	2	1092.87

[1] Number between parenthesis is the actual flyby number in the CSU solution.

The characteristics of the flybys and low-thrust legs of the sequence are given respectively in Tables 9.6 and 9.7. Looking at the gravity assists one can see clearly again the resonance pattern in the sequence for leg one and three. Different from the first sequence, this one does perform a low-thrust arc in the second leg. The actual leg has two coasting arcs and one thrust arc combined. Notice that due to the larger distance with Jupiter the mass penalty for close proximity is less visible if not negligible in the results.

Table 9.7: *Characteristics of the low-thrust legs for the second sub-sequence Ga-Ga-Ca-Ca-Ga.*

# Leg[1]	Moons	Epoch [MJD]	Arc type	TOF [days]	ΔV [m/s]	T_{max} [N]
1 (103-104)	Ga-Ga	60510.155	coasting	7.16	0.0	0.0
2 (104-105)	Ga-Ca	60517.312	coasting + thrusting	7.55	26.26	0.1
3 (105-106)	Ca-Ca	60524.864	coasting	16.70	0.0	0.0
4 (106-107)	Ca-Ga	60541.561	coasting	4.36	0.0	0.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.

For comparison of the spherical shaping accuracy in the test suite the orbital elements of the departure and arrival terminals are given in Table 9.8.

Table 9.8: *Orbital elements of the low-thrust legs of sub-sequence two.*

Element	Unit	Leg 1_d	Leg 1_a	Leg 2_d	Leg 2_a
a	[km]	10705874.7	10705874.7	1502562.4	1502562.4
e	[-]	0.127	0.127	0.294	0.293
i	[°]	4.6	4.6	0.4	0.5
Ω	[°]	163.8	163.8	174.7	174.6
ω	[°]	96.6	96.6	4.2	4.1
θ^1	[°]	262.5	262.5	343.9	203.3
Element	**Unit**	**Leg 3_d**	**Leg 3_a**	**Leg 4_d**	**Leg 4_a**
a	[km]	1883136.6	1883136.6	1502741.7	1502741.7
e	[-]	0.141	0.141	0.295	0.295
i	[°]	7.9	7.9	0.4	0.4
Ω	[°]	200.4	200.4	171.3	171.30
ω	[°]	280.1	280.1	8.0	8.0
θ^1	[°]	261.6	261.6	202.7	342.6

[1] True anomaly at epoch of departure of arrival

9.3 Setup - Strategy and Window

The optimisation of the sub-sequences is divided into smaller subsets of the sub-sequence. Each subset leads to a solution that is used as guideline for the following subset. For the sub-sequences of five gravity-assist manoeuvres this means that there are four total subsets to be computed. Initially, the first two flybys are used for optimisation. The solution from this is used as search direction for the next problem with the first three flybys. After optimising the latter, again the result is used as search direction for the optimisation of the first four flybys. Finally the full sub-sequence is optimised with the result of the last subset.

The reason behind splitting up the sub-sequence into smaller subsets is to find viable solutions in the smaller problems which are computed faster compared to the larger sequences (more decision variables and larger population size). Of course the solution of the smaller subset is not guaranteed to be part of the full optimal solution. Actually it is more likely that the final solution to the full problem is not equal to the initial solution found by the smaller subsets. This is, because the last flyby is not constrained to match a low-thrust leg to the next flyby. By continuing to the one flyby larger sequence, the last flyby of the previous sequence will most likely change to match an optimal trajectory to the next flyby. However, the smaller subsets do allow for promising search areas for the larger sub-sequences.

In Table 9.1 the boundary values for the decision variables were given. However, a few of them had the value 'strategy defined'. The values concerned are the epoch for the start time of the sequence and the time-of-flight values of the low-thrust legs. Lower and upper boundaries of both epoch and time-of-flight are being defined by the optimisation strategies defined in this section. Note that only the time parameters are influenced by the strategies. It was not considered effective to limit the gravity-assist manoeuvres more than by the constraints given by GTOC6. Also by only allowing changes in the phases of the moons (combination of epoch and TOF) the full GTOC6 problem was still considered relatively to the confinement of the subsets.

Two main strategies are to be considered, namely small freedom and full freedom. They refer to the freedom or amount of time that is allowed for the optimisation to look around the given epoch of a flyby for better solutions. The freedom is predefined for both strategies to be a constant multiplied with the synodical period between two consecutive flybys. Synodical periods are chosen as base for the time shift, because they define the period in which the two moons corresponding to the two flybys return to the same relative phase between them. Although the solution space is not a repeating cycle for each synodical period due to transfers

# Subset	# Flybys	Flyby 1	Flyby 2	Flyby 3	Flyby 4	Flyby 5
1	2	Small	Small			
2	3	Strict	Small	Small		
3	4	Strict	Strict	Small	Small	
4	5	Strict	Strict	Strict	Small	Small

Figure 9.1: *Example of the window shift (window size is two flybys) and for all subsets of the whole sub-sequence. Strategy is small freedom.*

to other moons, it is an indication of where similar solutions can be found. In case of a transfer between the same moon, the actual orbital period is chosen. In Equations (9.3) and (9.4) the relations are given to compute the boundaries for the new subset.

$$Epoch_{lb,\,ub} = Epoch_{winner\,subset} \pm C_{strategy} \cdot T_{syn} \tag{9.3}$$

$$TOF_{lb,\,ub} = TOF_{winner\,subset} \pm C_{strategy} \cdot T_{syn} \tag{9.4}$$

Here $Epoch_{lb,\,ub}$ and $TOF_{lb,\,ub}$ are respectively the lower and upper boundaries of the epoch and time-of-flight of the new subset problem. T_{syn} is the synodic period of the given moon transfer. Note that the latter is computed for each low-thrust leg and depends on which moons are involved in the trajectory leg. Furthermore, $Epoch_{winner\,subset}$ and $TOF_{winner\,subset}$ are the values of the optimisation solution of the previous subset. It is around this solution that the freedom will be applied. The last one is $C_{strategy}$, which is a constant that states how much freedom will be given depending on the strategy chosen. For full freedom $C_{strategy} = 2.5$, small freedom $C_{strategy} = 1.0$ and for strict freedom, which is the default for all flybys, $C_{strategy} = 0.5$. In other words, strict freedom allows the optimiser to find solutions within one cycle of the current moon transfer.

To complicate the matter, there are three windows sizes (number of flybys) for which the strategies apply. Outside the window the strict freedom is applied to the remaining flybys, which allows minimal shift in the phases of the flybys. The window moves with the last flyby in the sequence and therefore will shift one flyby with each new subset. In Figure 9.1 an overview is given of the window shifting through the sequence for each subset of the whole sub-sequence. Each row (subset) defines one optimisation problem having either two, three, four or five flybys. Small and strict indicate the freedom of the given flyby and the dark shaded area defines the window. The same principle applies for the full freedom strategy and using different window sizes.

For the initial subset of two flybys the boundaries are widened a bit more compared to the strategy settings. The epoch has a fixed offset of ± 25.0 days and the time-of-flight variable has an offset of the usual window setting multiplied by two.

The total number of test cases for the strategy and window test suite is six. Each of them will be repeated three times to account for the variation in optimiser performance with different trials. According to the tests from the previous chapter (for population size of $9D$) nearly 90 % of the trials achieved a satisfying solution within a 25 m/s deviation of the expected optimum. Important to remark is that the expected optimum is not known yet, except for the values produced by the CSU solution. However, that is a different problem compared to the model used here. Also the robustness of the model used in the test suits in the previous chapter might not result in the same robustness for the current model used for GTOC6. Therefore it is hard to say how many trials are required to get decent results. In the previous chapter trials of 50 or 10 were performed. However, 50 is by far impossible with respect to computational effort. Also 10 is considered too much for the same reason, because the problem with 10 trials had two thrust legs and one gravity assist. The current problem goes from two flybys with one low-thrust arc to a total of five flybys with four thrust arcs. This is a considerable amount of

computational effort on top of the problem with 10 trials. With the robustness of the simple model of the previous chapter in mind it is decided to have three trials per test.

9.4 Objective

Two different objective functions are optimised for the GTOC6 sequence problem. First the sequences are optimised for the total ΔV that is required for the low-thrust legs. This needs to be kept as low as possible and at the same time the gravity-assist manoeuvres need to be utilised to their full extent. The reason for first optimising ΔV is to compare the solutions with the CSU solution and to see how well the trajectories are performing considering the issues that are known with the spherical shaping method. Also a better understanding of the performance of the optimisation strategies is obtained. The objective function for ΔV, or fitness value f, is given in Equation (9.5).

$$f = \Delta V_{tot} = \sum_{i=1}^{N_{flybys}-1} (\Delta V_{leg\,i} + \Delta V_{pen\,i}) \tag{9.5}$$

Here ΔV_{tot} is the fitness value and $\Delta V_{leg\,i}$ the velocity change needed for low-thrust leg i. Furthermore, $\Delta V_{pen\,i}$ is the penalty applied to leg i due to constraint violations. In the next section more will be explained about the penalties.

The second objective function is a combination of the previous objective and the one defined for the original GTOC6 problem. It is a multi-objective function, however, the DE algorithm in PaGMO does not support multi-objective functions. Therefore the two objectives of respectively optimising for total low-thrust ΔV and for moon face mapping points, are joined in a weighted sum objective function. Note that the two objectives are contradicting each other in the optimisation: whereas ΔV needs to be minimised, the mapping points on the other hand are to be maximised. Therefore the weighted sum function in Equation (9.6) was derived.

$$f = -\left[W_{\Delta V} \cdot \frac{1}{\sum_{i=1}^{N_{flyby}-1} \left(\frac{\Delta V_{leg\,i} + \Delta V_{pen\,i}}{N_{flybys}-1} \right) + 1.0} + W_M \cdot \frac{\sum_{i=1}^{N_{flyby}-1} W_S F_{V_i}}{N_{flybys}} \right] \tag{9.6}$$

Here $W_{\Delta V}$ and W_M are respectively the weights for the ΔV and moon face mapping points. The summation of the ΔV is the same as in Equation (9.5), but is now placed in the denominator to allow for maximisation like the moon points. The placement in the denominator is also done to allow for better scaling with the moon points objective, because ΔV has a considerable large range of values, especially due to high penalties, and therefore can possibly overthrow the optimisation goal. In other words, the weights can be set to optimise for moon points in particular, but due to the large value of ΔV the moon points are of no significance in the objective value. Note the added 1.0 constant to avoid a division by zero, because a solution with only coasting arcs is possible. Also the total ΔV is divided by the number of legs ($N_{flybys} - 1$) to be able to compare the objective functions of the different subsets that have different number of flybys (normalisation). Nearly the same is done for the summation of the face values. Here the total is divided by the number of flybys, because the points are scored per flyby. The values for the weights will be discussed in the next chapter.

9.5 Penalties & Constraints

There are two penalties and one constraint that were not mentioned in the optimisation problem before. Constraints like a minimum hyperbolic excess velocity for the flybys are defined by boundaries of the optimisation problem. However the thrust constraint is checked afterwards, when the legs have been computed already. This results in considerable computational effort that is possibly wasted, because the solution is not possible in the first place. However, this can only be checked afterwards in case of thrust constraint violation. The two penalties relate to the mass penalty due to close proximity with Jupiter and the inefficiency of the spherical shaping algorithm for increasing values of the inclination and relative RAAN steps.

The mass penalty for radiation from Jupiter has been explained in Chapter 2 and does not require further elaboration. However, the black box constraints defined for the spherical shaping method is another story. After some initial testing it turned out that a limited number of trajectories proved to be viable solutions during optimisation. On top of that adding the thrust constraint, which will be discussed next, resulted in even worse results. This impeded the optimisation process significantly. To make sure the optimisation process was not obstructed by the shaping penalty it was decided to leave out this penalty. The idea of the penalty was to have less efficient solutions in the solution space, however it took away the ease of the optimisation to get through the difficult solution space filled with many local optima. Note that the D parameter suffered from undefined (NaN's) and negative values that caused ΔV penalties. However, these occurred mostly for considerable eccentric and inclined orbits and the total occurrences were minor compared to the thrust constraint violations. The latter was dominant and therefore slowed down the optimisation process significantly.

Finally there is the thrust constraint. As mentioned before the thrust can not be capped in the spherical shaping method and has to be verified afterwards. An early adoption of the thrust constraint in the GTOC6 optimisation model resulted in considerable unfavourable solutions. This was due to the implementation which as soon as a thrust leg had violated the thrust constraint, the complete solution was dropped. Dropped solutions were retained in the solution set, but were assigned very high penalty values and therefore the solution space was flattened with invalid solutions. The model was changed by letting violated solutions to be continued with the addition of a penalty proportional to the violation of the thrust constraint and also multiplied by a constant factor (100). This allowed the optimiser to continue with the solutions and providing a smoother solution space to go through.

Also it turned out that the thrust constraint set by GTOC6, though not impossible, was very strict. This also caused the optimiser to have trouble to go smoothly to the favourable solutions. In order to improve the performance of the optimiser the thrust constraint was widened by a factor of 10.0 to a maximum thrust of 1.0 N. The results was that more viable solutions were found, including more solutions that satisfied the actual GTOC6 thrust constraint. More on the results can be found in the next chapter.

Part IV

Results & Conclusions

10 Results

After a thorough analysis of the GTOC6 problem and the development of an algorithm capable of computing low-thrust multiple gravity-assist trajectories and validating the methods, the actual GTOC6 problem is tackled. The results of the two sequences that were discussed in the previous chapter are elaborated upon in this chapter. For the ΔV optimisation both sequences were considered whereas for the combined objective of ΔV and the score for moon mapping, only the first sequence is simulated. The first sequence is chosen, because it shows more challenging features like shorter orbital periods (faster dynamics) and close proximity with Jupiter resulting in high mass penalties.

10.1 Initial Results

For the optimisation of the two sub-sequences a total of 36 cases were simulated. This consisted of the two optimisation strategies times the three windows sizes and that multiplied with three trials for each sequence. The subsets with two, three, four and five flybys were given sufficient generations to be able to converge to a satisfying solution. For each subset a different number of generations was used to comply with the population size, the number of decision variables and the complexity for increasing number of gravity-assist manoeuvres. The number of generations was respectively 2500, 10000, 20000 and 40000. Although the values seem grotesque, they proved necessary for the problem.

The problems experienced considerably bad convergence and the solutions to all the simulations are not similar, even though the function values could be considered similar. All simulations required to complete the full maximum number of available generations to finish the optimisation. Even the final populations showed sufficient different individuals with considerable differences in their function values. As result no proper convergence was detected by the algorithm even though the champion of the population showed a converging trend. It was possible to increase the tolerance on the build-in convergence check by the *randDE* algorithm of PaGMO. However, a considerable high tolerance was needed to have convergence before the maximum number of generations was reached. This showed that the problem has difficulty with converging. As result the computational effort was enormous for the subsets. Each subset with two, three, four and five flybys required on average computational times of respectively 75 minutes, 14.5 hours, 50 hours and 7.5 days.

Initial testing showed no viable solutions that are below the thrust limit of 0.1 N defined by GTOC6. Possible reasoning here is that the implementation suffers from inefficient trajectories for increasing inclinations and line of nodes changes. So the spacecraft is required to do a large amount of thrusting whereas better trajectories that require less thrust are possible. For this reasoning, it is possible that the 0.1 N thrust levels are too demanding combined with the spherical shaping and the used optimisation problem and algorithm. As mentioned in the previous chapter it was decided to increase the thrust limit to 1.0 N to allow for better optimisation. The result was an improved optimisation with consistently better results. Among the results were also solutions that satisfied the original thrust constraint. This showed that allowing for more freedom in the optimisation creates a more smooth and efficient search that is not obstructed significantly by the penalties of surrounding solutions in the search space.

Another very important aspect is the following: though the spherical shaping method applies thrust, it does not with a constant (maximum) force. Most GTOC6 contesters on the other hand used reference trajectories for a local low-thrust optimiser afterwards that optimised for maximum thrust. So a large amount of efficiency is lost here due to non-efficient thrusting profiles. Also because of the non-constant nature of the spherical shaping method, there are peaks to be expected which will make it harder to get proper solutions that are satisfying the thrust constraints. On top of that spherical shaping should be used as initial guess/reference trajectory and not as final solution, for which it is not sufficient. The same approach as the participants used is recommended: to use the solutions as reference trajectories for local optimisers.

Furthermore, the DE settings used and the custom use of the 'adaptive' method *randDE* was promising for the given test problems. However, for the main problems, it became clear that the champion solutions are sometimes thrown away, whereas one would prefer to keep them. This gives the population a setback that is not wanted. Due to the randomness of the control parameters it is possible for the parameters to be chosen such that the promising individuals will still be thrown away.. However, this was not foreseen and visible in the test problem. For that particular problem it actually worked out considerably well. Therefore it should be considered to add elitism to the algorithm to keep a small selection of optimal solutions in the population.

Also note the penalty function for high levels of thrust. The maximum thrust is set to 1.0 N to allow for more solutions, however still the solution space is very thin with proper solutions, making it really hard for the optimiser to get through. It is clear from the results that ΔV values of above 1000.0 m/s are most likely affected by the penalty factor of 100 to avoid the high-thrust solutions. The effect can be seen as the huge drop in fitness value when a solution is within the thrust constraint. As said before the thrust levels are not constant and many GTOC6 participants actually showed constant maximum thrust for their solutions. Moreover they used these kind of trajectories as input for their local optimisers later on.

10.2 Optimisation of the Sequences for ΔV

The results of optimising ΔV will be explained by discussing two cases. Each case represents one of the two sequences with one of the six strategies. After that the results of the other cases are discussed followed by analysis of the overall optimisation.

10.2.1 Small Freedom and 2 Flyby Window Applied to Sequence One

The first case is the Eu-Eu-Eu-Io-Io sequence. Here a small freedom is applied around the timing of the flybys for the smallest window of two flybys. A considerable large amount of data has been produced by the optimisation and after processing the data, two tables with output are generated which are given in Tables 10.1 and 10.2. The former table gives the details on the flybys that are found by the optimisation using the defined strategy whereas the latter shows the low-thrust arcs to connect the flybys. Both tables will be discussed.

In Table 10.1 three blocks are shown where each one of them represents one trial run. A trial consists of four separate optimisation problems ranging from two to the complete five flybys of the given sequence. For each subset the flyby numbers and the corresponding moons are given. Furthermore the epoch of the flyby is shown together with the approaching hyperbolic excess velocity $V_{-\infty}$ of the spacecraft and its altitude h_{flyby} above the surface of the moon. The last column is the mass $m_{post\,flyby}$ of the spacecraft just after the flyby. This includes the mass penalties and the propellant mass for the low-thrust arc preceding the flyby.

As mentioned before the solution of the preceding subset is used to define the search area for the new subset. Here only the parameters epoch and the time-of-flight of each leg are used as a baseline value. Around the baseline the offsets are computed based on the strategy. The remaining parameters of the decision vector have their boundaries defined in Table 9.1.

Table 10.2 contains the information regarding the low-thrust arcs for all subsets and all trials. As with the previous table, the flyby numbers and corresponding moons are given. However, they now indicate the transfer trajectory and therefore two moons are supplied for each entry to indicate the departure and arrival body. Furthermore, the epoch of the start of the leg and time-of-flight are given. The column with the number of revolutions relates to how many complete (integer) revolutions are performed during the trajectory. Finally the last two columns provide the ΔV of the legs and the total of each subset is supplied as well. Note that the first column is the optimisation objective ΔV which includes the penalties for thrust constraint violation. Due to disappointing performance of sequence one related to experiencing a dominating influence of the thrust penalties, the actual calculated ΔV by the spherical shaping method is also given.

From the three trials performed with the small freedom and two flyby window strategy a few things are very noticeable in the results. Starting with Table 10.1 there is a clear distinction in the hyperbolic excess velocities between subsets with two or three flybys and four or five flybys. For the former the optimisation finds $V_{-\infty}$ near the lower boundary condition value. The gravity-assist altitudes vary over the supplied range $[50.0 - 2000.0\,\text{km}]$. On the other hand for the larger subsets (i.e. four and five flybys), $V_{-\infty}$ comes in the neighbourhood of the values found in the CSU solution. Furthermore, the flyby altitudes are significantly lower indicating that the GA's are utilised better to change the trajectory instead of using low thrust. Note that several altitudes have reached the lower boundary value of 50.0 km and have maximised the potential of the flyby with respect to the altitude. It confirms that the optimisation is indeed searching for the timings for matching the GA's to avoid using low thrust. Note that the last GA in each subset shows deviating values compared to the predecessors in the subset, especially the flyby altitude. This is explained due to the fact that the last GA is not connected to another thrust leg with a flyby. Therefore the conditions of leaving the last moon do not influence the preceding sequence. This allows for more freedom on the last flyby arrival conditions. A similar analogy holds for the first flyby. Now there is not a thrust leg in advance determining the incoming conditions of the first flyby. Therefore only what occurs after the first flyby determines the conditions for the first GA in the subset.

Though the flybys are considerably utilised, it can be seen in Table 10.2 that the optimisation finds grotesque values for ΔV. This includes the penalties for the thrust constraint violation which is clearly present with these large ΔV's. For that reason the second column with the actual ΔV's required for the low-thrust arcs is given. Immediately it can be seen that the optimisation is not able to find the coasting arcs from the CSU solution. Also the values are relatively large compared to the ΔV's found in general by CSU, which goes up till several multitudes of 10.0 m/s.

Table 10.1: *Results (3x) of subsets for the gravity-assist manoeuvres for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: small freedom and 2 flyby window]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59873.483	301.4	1743.8	1889.67
2 (16)	Eu	59880.586	301.0	1320.2	1885.24
1 (15)	Eu	59875.023	300.6	1595.9	1889.67
2 (16)	Eu	59882.129	301.0	798.4	1885.22
3 (17)	Eu	59889.235	300.9	1169.4	1882.06
1 (15)	Eu	59873.288	1819.8	59.3	1889.67
2 (16)	Eu	59880.393	2027.7	186.0	1865.84
3 (17)	Eu	59887.389	2179.7	50.0	1839.57
4 (18)	Io	59891.106	1858.9	1522.6	1811.07
1 (15)	Eu	59871.511	1733.4	725.3	1889.67
2 (16)	Eu	59878.568	1782.7	96.5	1863.58
3 (17)	Eu	59883.860	1833.7	50.0	1834.43
4 (18)	Io	59887.710	2282.8	50.0	1784.00
5 (19)	Io	59891.253	2289.2	1678.4	1771.59
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59881.681	356.3	964.6	1889.67
2 (16)	Eu	59887.006	363.9	1877.0	1886.51
1 (15)	Eu	59880.491	302.5	661.4	1889.67
2 (16)	Eu	59887.594	300.4	115.5	1884.88
3 (17)	Eu	59894.697	300.4	710.7	1880.47
1 (15)	Eu	59882.101	1584.3	710.0	1889.67
2 (16)	Eu	59887.428	1615.2	50.0	1868.07
3 (17)	Eu	59894.453	1987.3	50.0	1833.13
4 (18)	Io	59898.214	1846.5	1037.2	1801.56
1 (15)	Eu	59880.325	1640.7	1225.9	1889.67
2 (16)	Eu	59887.419	1788.5	653.3	1857.38
3 (17)	Eu	59894.480	1813.8	50.0	1830.44
4 (18)	Io	59898.308	2208.7	50.0	1772.46
5 (19)	Io	59900.970	2195.5	1383.3	1761.23
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59872.853	311.9	1030.8	1889.67
2 (16)	Eu	59878.191	306.0	586.6	1886.63
1 (15)	Eu	59873.810	309.9	1602.0	1889.67
2 (16)	Eu	59880.911	311.6	85.4	1886.34
3 (17)	Eu	59886.262	312.3	534.4	1883.03
1 (15)	Eu	59875.081	1438.0	1038.5	1889.67
2 (16)	Eu	59880.405	1468.3	50.0	1867.94
3 (17)	Eu	59887.397	1906.2	50.0	1830.41
4 (18)	Io	59891.179	1832.0	470.8	1794.90
1 (15)	Eu	59873.309	1705.6	606.3	1889.67
2 (16)	Eu	59880.377	1783.3	1010.0	1860.31
3 (17)	Eu	59887.437	1827.1	50.0	1833.06
4 (18)	Io	59891.252	2178.2	50.0	1774.87
5 (19)	Io	59893.914	2161.9	536.9	1762.28

[1] Number between parenthesis is the actual flyby number in the CSU solution.

Table 10.2: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: small freedom and 2 flyby window]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59873.483	7.10	1	2.0	2.0
				ΔV_{tot} =	2.0	2.0
1 (15-16)	Eu-Eu	59875.023	7.10	1	1.0	1.0
2 (16-17)	Eu-Eu	59882.129	7.10	1	3.0	3.0
				ΔV_{tot} =	4.0	4.0
1 (15-16)	Eu-Eu	59873.288	7.10	1	225.0	225.0
2 (16-17)	Eu-Eu	59880.393	7.00	1	255.0	255.0
3 (17-18)	Eu-Io	59887.389	3.72	1	65439.0	294.0
				ΔV_{tot} =	65919.0	774.0
1 (15-16)	Eu-Eu	59871.511	7.06	1	249.0	249.0
2 (16-17)	Eu-Eu	59878.568	5.29	1	56420.0	281.0
3 (17-18)	Eu-Io	59883.860	3.85	1	174816.0	534.0
4 (18-19)	Io-Io	59887.710	3.54	1	80.0	80.0
				ΔV_{tot} =	231565.0	1144.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59881.681	5.33	1	3.0	3.0
				ΔV_{tot} =	3.0	3.0
1 (15-16)	Eu-Eu	59880.491	7.10	1	8.0	8.0
2 (16-17)	Eu-Eu	59887.594	7.10	1	3.0	3.0
				ΔV_{tot} =	11.0	11.0
1 (15-16)	Eu-Eu	59882.101	5.33	1	198.0	198.0
2 (16-17)	Eu-Eu	59887.428	7.03	1	59029.0	347.0
3 (17-18)	Eu-Io	59894.453	3.76	1	80511.0	328.0
				ΔV_{tot} =	139738.0	873.0
1 (15-16)	Eu-Eu	59880.325	7.09	1	296.0	296.0
2 (16-17)	Eu-Eu	59887.419	7.06	1	261.0	261.0
3 (17-18)	Eu-Io	59894.480	3.83	1	238514.0	619.0
4 (18-19)	Io-Io	59898.308	2.66	1	86.0	86.0
				ΔV_{tot} =	239157.0	1262.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59872.853	5.34	1	2.0	2.0
				ΔV_{tot} =	2.0	2.0
1 (15-16)	Eu-Eu	59873.810	7.10	1	4.0	4.0
2 (16-17)	Eu-Eu	59880.911	5.35	1	5.0	5.0
				ΔV_{tot} =	9.0	9.0
1 (15-16)	Eu-Eu	59875.081	5.32	1	200.0	200.0
2 (16-17)	Eu-Eu	59880.405	6.99	1	66606.0	375.0
3 (17-18)	Eu-Io	59887.397	3.78	1	96986.0	372.0
				ΔV_{tot} =	163792.0	947.0
1 (15-16)	Eu-Eu	59873.309	7.07	1	263.0	263.0
2 (16-17)	Eu-Eu	59880.377	7.06	1	265.0	265.0
3 (17-18)	Eu-Io	59887.437	3.82	1	238087.0	620.0
4 (18-19)	Io-Io	59891.252	2.66	1	100.0	100.0
				ΔV_{tot} =	238715.0	1248.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

The large ΔV's are associated with large thrust demand during the trajectory. In Figures 10.1 - 10.4 the thrust profiles are shown for the legs in each subset for the first trial. For the subsets with two or three flybys the ΔV for the legs is low, which translates to thrust profiles that are even below the GTOC6 constraint, see Figures 10.1 and 10.2. However, when the subset increases to four and five flybys, the parameters defining the flyby trajectory change

significantly. This is also very visible in the thrust profiles found in Figures 10.3 and 10.4. The thrust magnitude goes beyond the GTOC6 constraint and as mentioned before the constraint had to be widened with a factor 10 to allow for a smooth optimisation. Although, increasing the thrust constraint was fruitful, it is still surpassed by a few legs resulting in high ΔV penalties. As a consequence, the optimisation is focussed on minimising the penalties instead of the actual ΔV of the legs. All trials experience this, and actually all strategy cases for sequence one had the same problem.

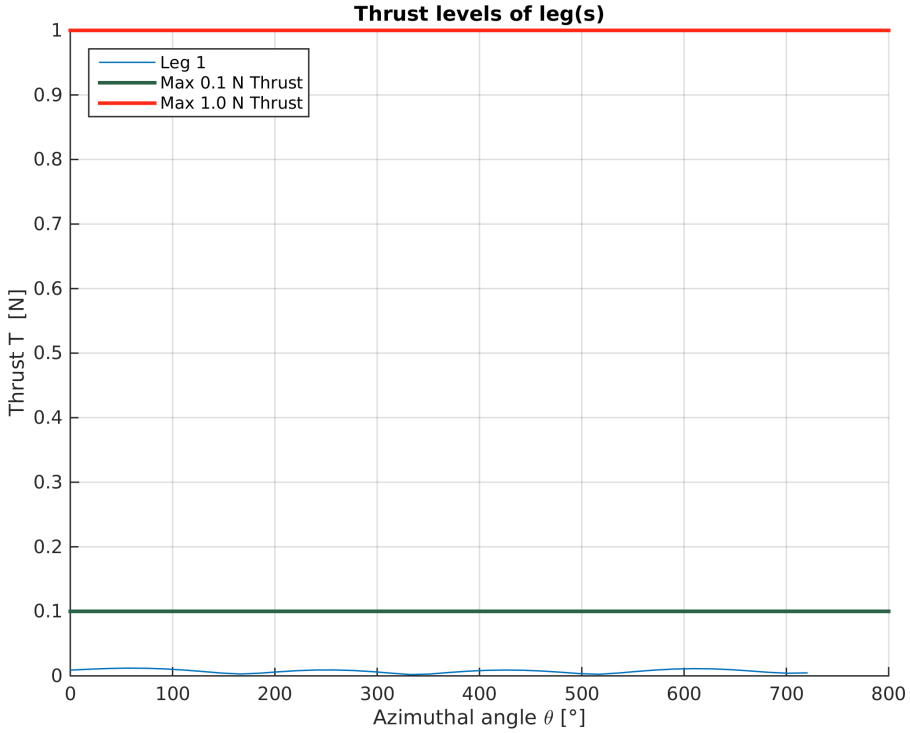

Figure 10.1: *Thrust profile of the subset Eu-Eu (trial 1) for sequence one with small freedom and 2 flyby window strategy.*

Besides the penalties, several low-thrust arcs require considerable amounts of ΔV to reach their destination. Apparently, the spherical shaping method is generating thrust arcs that require large accelerations to maintain the transfer trajectory. This could be due to the accuracy limitations of the shaping method as discussed in Chapter 4. To verify this, the orbital elements for the legs of the subset with five flybys are examined. They are given in Table 10.3.

From the data in the Table 10.3 it is remarkable that the differences in RAAN for the thrust legs are relatively small, which is required to keep the accuracy high of the spherical shaping method. For the given inclinations and RAAN's the error in ΔV is roughly 50.0 m/s according to Figure 4.12 and 4.13. On top of that is the actual ΔV required for non-inclined change of RAAN (reference trajectory). Note that the accuracy tests where performed for different inclinations and RAAN, but the remaining elements were kept the same between the departure and arrival orbit. Furthermore, legs two and three show the largest thrust violations in Figure 10.4. Though RAAN is not changed significantly, the elements argument of periapsis ω and true anomaly θ differ completely. The effect of changing the former two elements (difference in departure and arrival value) during the transfer was not taken into account during the accuracy tests and could therefore be a possible reason for the high ΔV besides the error of RAAN. This was also noted by Roegiers (2014). On top of that, both legs are in an inclined

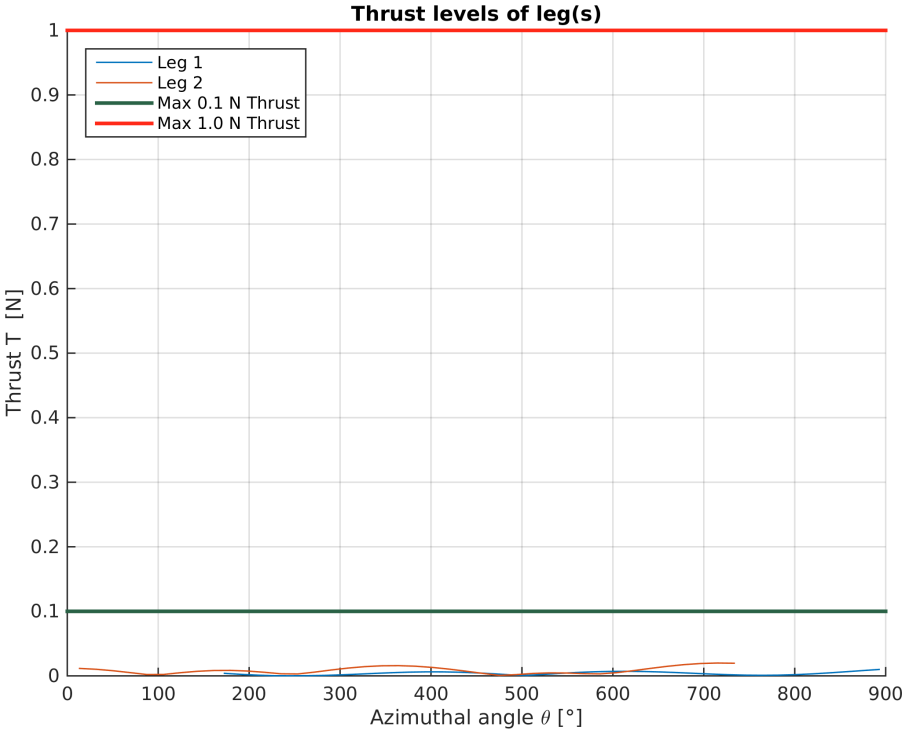

Figure 10.2: *Thrust profile of the subset Eu-Eu-Eu (trial 1) for sequence one with small freedom and 2 flyby window strategy.*

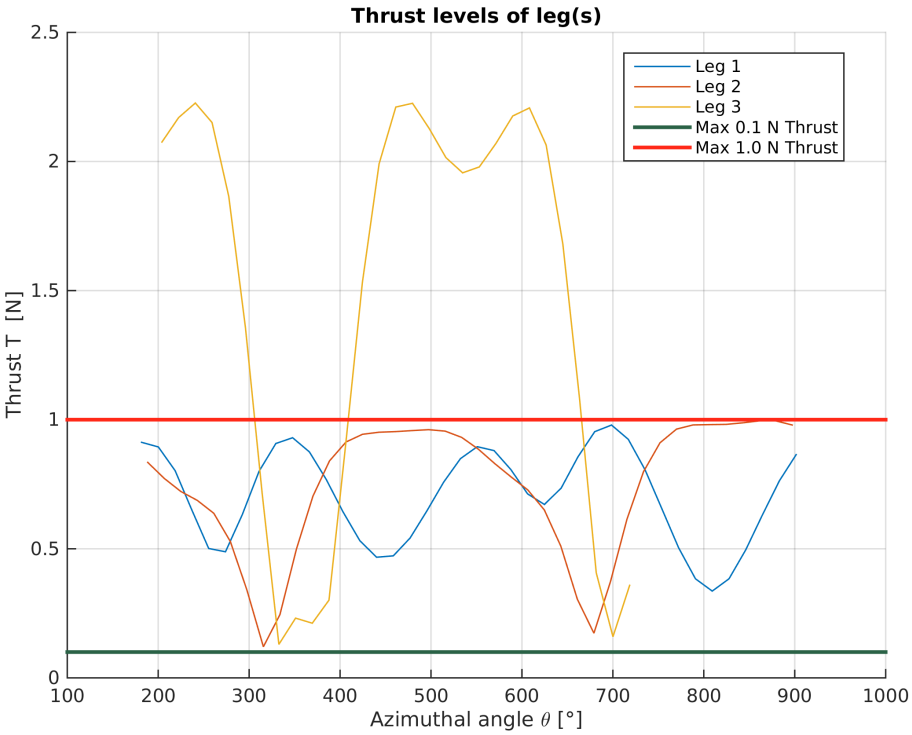

Figure 10.3: *Thrust profile of the subset Eu-Eu-Eu-Io (trial 1) for sequence one with small freedom and 2 flyby window strategy.*

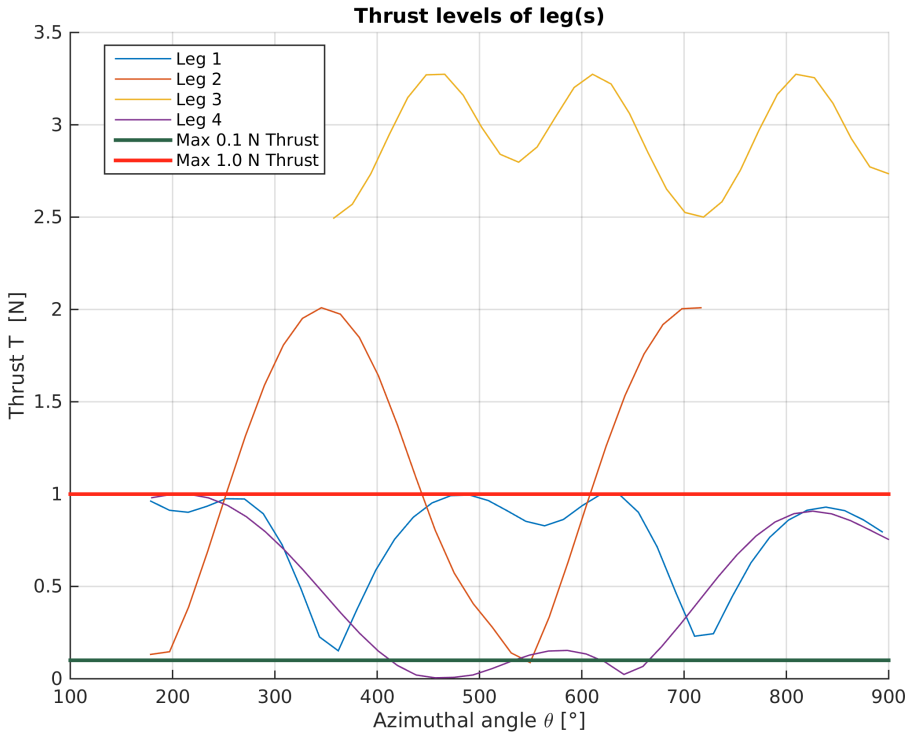

Figure 10.4: *Thrust profile of the subset Eu-Eu-Eu-Io-Io (trial 1) for sequence one with small freedom and 2 flyby window strategy.*

plane and leg three has a descent eccentricity. Also both legs require thrusting for decreasing the semi-major axis a. It is clear that the spherical shaping method is not able to find the coasting arc from the CSU solution and that it has trouble shaping the legs with proper values for ΔV which are to needed in order to fulfil the full GTOC6 problem. As consequence of the large ΔV the mass of the spacecraft also decreases faster, which translates to an additional loss of mass of $20 - 30\,\text{kg}$, due to propellant consumption, compared to the CSU solution.

Also remark that the number of complete revolutions is always one for all legs in every subset. This was set as the upper boundary condition based on the preceding literature study. This was to avoid long time-of-flights to map the moons in a relatively short time. Another reason to avoid multiple revolutions is the computational effort required for the spherical shaping method. For every revolution additional control points have to be verified and computed. These scale linear with computational effort and therefore allowing up till several revolutions could potentially result up to several times the current simulation durations as well. However, looking at the CSU solution, several legs show multiple revolutions to come back to the same moon (resonances, multitude of orbital periods). The latter can result in more favourable moon positions for the following flybys and thus lower ΔV for the thrust legs or even zero in case of a coasting arc. Finally in Figure 10.5 the trajectory is shown of trial 1.

In Figure 10.6 the evolution of the champion solutions for each trial is shown. It shows that the champions convergence to their solution for large numbers of generations. Also it can take a large number of generations for the objective to improve. Furthermore, the solutions do tend to converge to the same neighbourhood. However, note that the scale is very large due to the enormous values for ΔV. That the optimiser has difficulty with large sets of flybys can be seen in the difference in the number of generations of the trials. For example, trial one is roughly 5000 generations behind on trial three for $\Delta V = 0.75 \times 10^6\,\text{m/s}$. Though it does seem that the trials have converged after roughly 35,000 generations.

Table 10.3: *Orbital elements of the low-thrust legs of sub-sequence one.*

Element	Unit	Leg 1_d	Leg 1_a	Leg 2_d	Leg 2_a
a	[km]	$6.756\,12 \times 10^8$	$6.647\,22 \times 10^8$	$6.771\,53 \times 10^8$	$6.530\,69 \times 10^8$
e	[-]	0.07154	0.08157	0.01656	0.03551
i	[°]	6.72	6.67	7.76	7.76
ω	[°]	271.29	283.63	99.32	127.91
Ω	[°]	208.78	204.14	203.90	201.95
θ	[°]	267.32	254.65	79.20	230.50
Element	**Unit**	**Leg 3_d**	**Leg 3_a**	**Leg 4_d**	**Leg 4_a**
a	[km]	$5.606\,02 \times 10^8$	$5.227\,57 \times 10^8$	$4.237\,55 \times 10^8$	$4.235\,14 \times 10^8$
e	[-]	0.19171	0.19441	0.01444	0.01909
i	[°]	5.18	5.17	7.52	7.52
ω	[°]	172.02	176.45	113.34	105.07
Ω	[°]	202.74	202.83	202.70	202.70
θ	[°]	185.60	3.12	66.36	74.63

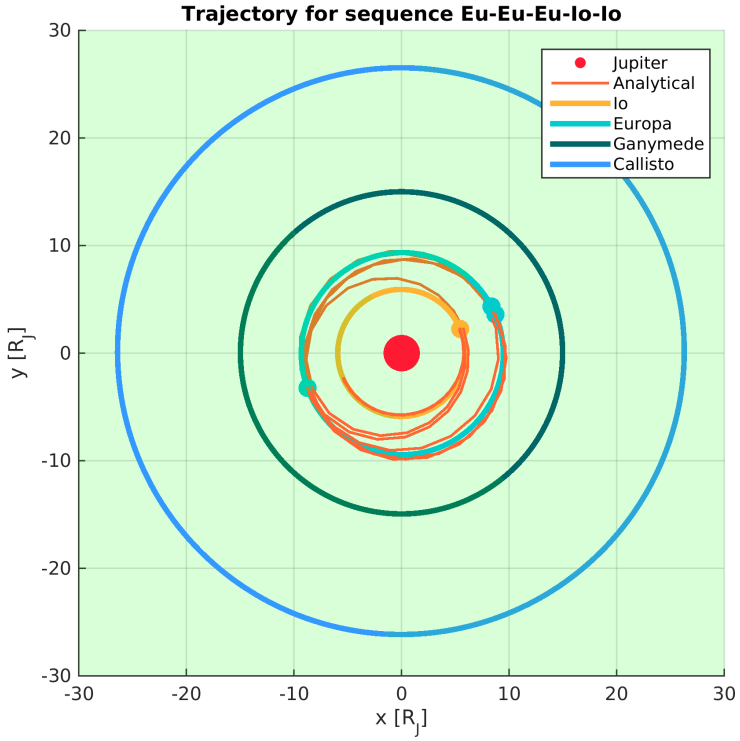

Figure 10.5: *Best trajectory solution to the Eu-Eu-Eu-Io-Io sequence (trial 1) with small freedom and two flyby window strategy.*

10.2.2 Full Freedom and 2 Flyby Window Applied to Sequence Two

The second case discussed here is the Ga-Ga-Ca-Ca-Ga sequence. For this case the full freedom is applied to the timing constraints of the optimisation problem. Also a window of two flybys is used to define the flybys with more freedom with respect to epoch and time-of-flight. Again, a considerable amount of data was generated and has been processed in similar tables as with the previous case in the previous section. In Table 10.4 the data of the flybys is given and the information on the legs of the subsets are given in Table 10.5. Both tables contain three trials.

Looking at the results a complete different picture is painted by optimisation of sequence two. First of all, there are no large values for ΔV as found in the results of sequence one. It is

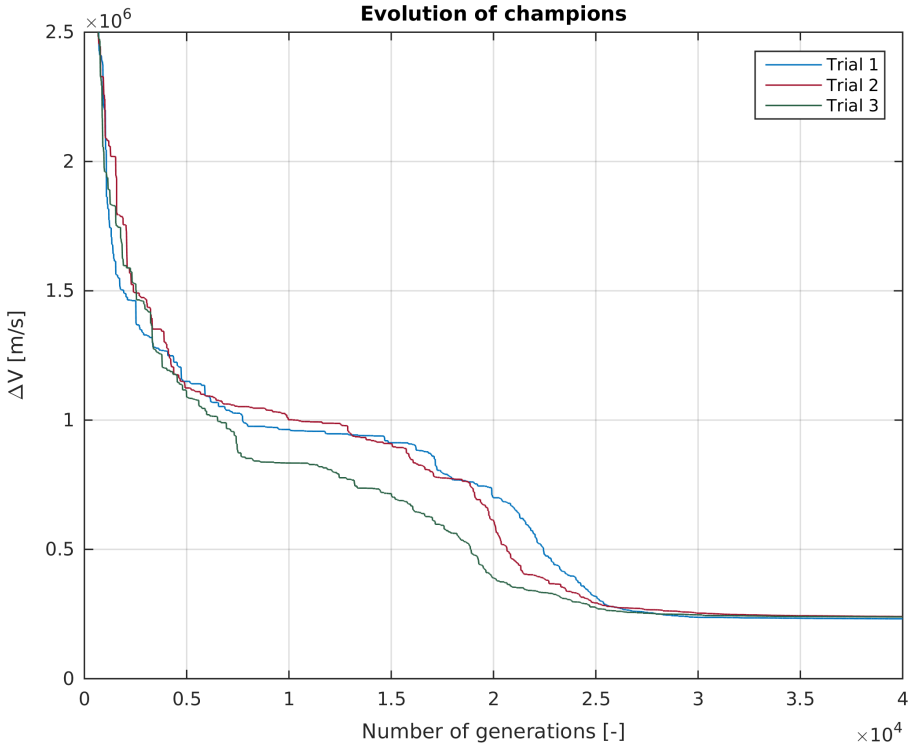

Figure 10.6: *Evolution of the champion for the Eu-Eu-Eu-Io-Io sequence with small freedom and two flyby window strategy.*

unambiguous that the optimisation has found thrust legs with near zero ΔV. Even for the large subset of five flybys two out of the three trials is able to find trajectories that utilise the full potential of flybys to change the course to the next moon. Note that the computation of ΔV suffers from accuracy due to the number of control points as mentioned in Section 4.5.3. The number of control points needed to be kept small to keep the simulation time within reasonable bounds given the complexity of the problem. The champion solutions of all the strategy results could have been verified afterwards with an increased accuracy for ΔV. However, due to time constraints this was not possible. This is left as future work and the results here are used as preliminary results. The results also confirm for the spherical shaping method, that (near) coasting arcs are possible with the RFT fix.

Continuing with the results a similar feature can be seen as with sequence one. The hyperbolic excess velocities are converging to nearly the same values for the same moon transfers (resonances). There is one small difference with sequence one. In the latter both subsets with two or three flybys differ significantly with respect to $V_{-\infty}$ whereas for sequence two it is alone the first subset of two flybys that deviates from the so called norm. The values for $V_{-\infty}$ are not the same compared to the CSU solution, but are in the relative neighbourhood.

Aside from the first subset of each trial, the other subsets have significantly lower flyby altitudes compared to the CSU solution. However, the solutions found by the optimisation are able to find (near) coasting arcs for all legs, compared to three coasting arcs and one thrust arc. Note that strictly speaking the legs are not considered coasting arcs, because ΔV is not equal to zero. However, the values are negligible small. Another remark is that the last two trials have found a similar solution and at completely different epochs. Furthermore, the first trial starts around one third of a day earlier and ends the sequence of five flybys with a smaller difference resulting in a considerable worse solution (150 m/s compared to ~ 0 m/s).

Table 10.4: *Results (3x) of subsets for the gravity-assist manoeuvres for the first sub-sequence Ga-Ga-Ca-Ca-Ga. [Settings: full freedom and 2 flyby window]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60522.177	335.6	1449.6	1098.51
2 (104)	Ga	60532.919	336.5	1537.3	1097.43
1 (103)	Ga	60524.012	1471.5	854.0	1098.51
2 (104)	Ga	60538.326	1471.5	119.7	1097.39
3 (105)	Ca	60555.856	1251.5	1316.9	1096.84
1 (103)	Ga	60523.625	1320.3	1878.3	1098.51
2 (104)	Ga	60537.928	1325.0	686.3	1096.37
3 (105)	Ca	60555.405	1070.3	122.5	1076.97
4 (106)	Ca	60588.796	1056.5	1162.1	1075.29
1 (103)	Ga	60524.166	1519.6	1082.3	1098.51
2 (104)	Ga	60538.476	1520.4	144.1	1097.17
3 (105)	Ca	60556.730	1330.4	50.0	1094.31
4 (106)	Ca	60581.844	1328.8	426.1	1090.56
5 (107)	Ga	60599.593	1667.6	1250.2	1088.16
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60529.484	328.2	955.0	1098.51
2 (104)	Ga	60540.223	329.0	710.4	1097.42
1 (103)	Ga	60527.668	1470.6	673.6	1098.51
2 (104)	Ga	60538.411	1467.6	118.3	1097.52
3 (105)	Ca	60555.979	1262.6	1119.8	1096.98
1 (103)	Ga	60527.694	1467.6	470.6	1098.51
2 (104)	Ga	60538.437	1464.7	130.4	1097.49
3 (105)	Ca	60556.047	1267.6	361.7	1096.93
4 (106)	Ca	60581.150	1255.1	973.7	1096.91
1 (103)	Ga	60524.516	1545.6	678.5	1098.51
2 (104)	Ga	60538.830	1545.6	63.2	1096.88
3 (105)	Ca	60556.882	1407.3	119.6	1096.33
4 (106)	Ca	60581.998	1399.0	632.9	1096.33
5 (107)	Ga	60599.696	1738.3	1032.6	1096.05
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60513.305	348.7	215.1	1098.51
2 (104)	Ga	60524.048	348.3	1720.4	1097.40
1 (103)	Ga	60511.794	1483.3	766.9	1098.51
2 (104)	Ga	60526.108	1483.3	201.1	1097.06
3 (105)	Ca	60543.821	1267.1	889.2	1096.51
1 (103)	Ga	60511.683	1456.0	102.6	1098.51
2 (104)	Ga	60525.997	1469.0	52.9	1096.85
3 (105)	Ca	60543.872	1248.0	313.7	1095.58
4 (106)	Ca	60568.960	1261.3	961.1	1095.18
1 (103)	Ga	60512.206	1611.0	76.0	1098.51
2 (104)	Ga	60526.520	1611.0	62.4	1096.79
3 (105)	Ca	60544.427	1346.4	50.4	1096.22
4 (106)	Ca	60569.502	1364.8	57.5	1096.22
5 (107)	Ga	60587.350	1530.8	913.8	1095.67

[1] Number between parenthesis is the actual flyby number in the CSU solution.

Compared to sequence one the second sequence does not require large ΔV and is not influenced by penalties. This is also translated to the thrust profiles of trial two which are provided in Figures 10.7 - 10.10. As mentioned before $\Delta V =\sim 0.0\,\mathrm{m/s}$ is not equal to zero and therefore not an actual coasting arc. It is not surprising that the legs require insignificant amounts to no thrust at all. The thrust constraint of 1.0 N and more importantly the GTOCT6 constraint of 0.1 N are both satisfied for all the subsets. Finally in Figure 10.11 the trajectory is shown of trial 2.

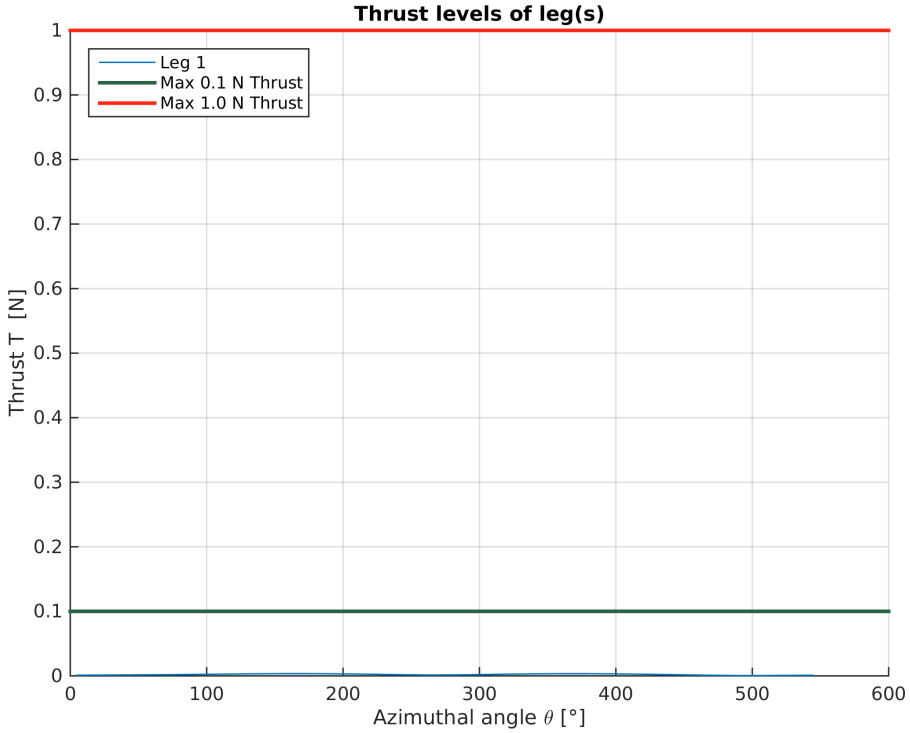

Figure 10.7: *Thrust profile of the subset Ga-Ga (trial 2) for sequence two with full freedom and 2 flyby window strategy.*

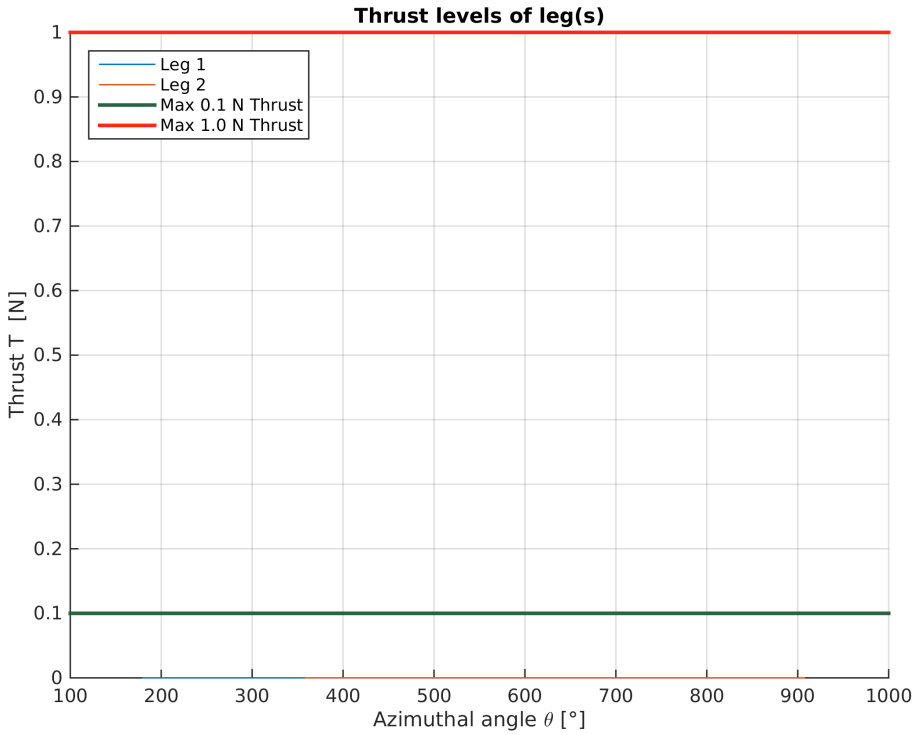

Figure 10.8: *Thrust profile of the subset Ga-Ga-Ca (trial 2) for sequence two with full freedom and 2 flyby window strategy.*

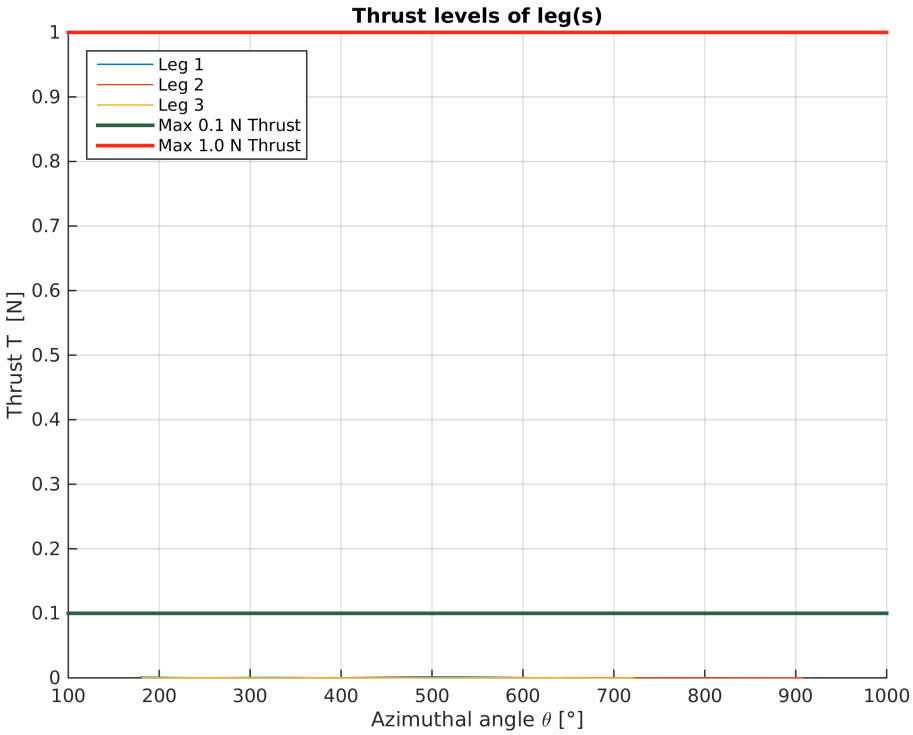

Figure 10.9: *Thrust profile of the subset Ga-Ga-Ca-Ca (trial 2) for sequence two with full freedom and 2 flyby window strategy.*

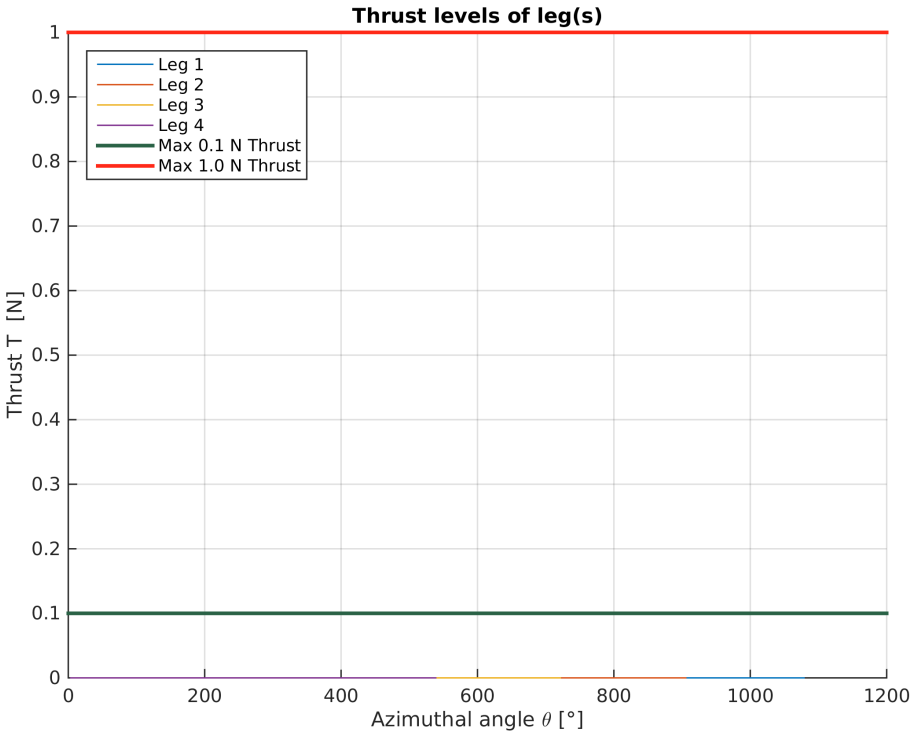

Figure 10.10: *Thrust profile of the subset Ga-Ga-Ca-Ca-Ga (trial 2) for sequence two with full freedom and 2 flyby window strategy.*

Table 10.5: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Ga-Ga-Ca-Ca-Ga. [Settings: full freedom and 2 flyby window]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60522.177	10.74	1	2.0	2.0
				$\Delta V_{tot} =$	2.0	2.0
1 (103-104)	Ga-Ga	60524.012	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60538.326	17.53	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
1 (103-104)	Ga-Ga	60523.625	14.30	1	21.0	21.0
2 (104-105)	Ga-Ca	60537.928	17.48	1	341.0	341.0
3 (105-106)	Ca-Ca	60555.405	33.39	1	31.0	31.0
				$\Delta V_{tot} =$	393.0	393.0
1 (103-104)	Ga-Ga	60524.166	14.31	1	4.0	4.0
2 (104-105)	Ga-Ca	60538.476	18.25	1	41.0	41.0
3 (105-106)	Ca-Ca	60556.730	25.11	1	67.0	67.0
4 (106-107)	Ca-Ga	60581.844	17.75	1	38.0	38.0
				$\Delta V_{tot} =$	150.0	150.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60529.484	10.74	1	2.0	2.0
				$\Delta V_{tot} =$	2.0	2.0
1 (103-104)	Ga-Ga	60527.668	10.74	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60538.411	17.57	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
1 (103-104)	Ga-Ga	60527.694	10.74	1	1.0	1.0
2 (104-105)	Ga-Ca	60538.437	17.61	1	∼0.0	∼0.0
3 (105-106)	Ca-Ca	60556.047	25.10	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	1.0	1.0
1 (103-104)	Ga-Ga	60524.516	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60538.830	18.05	1	∼0.0	∼0.0
3 (105-106)	Ca-Ca	60556.882	25.11	1	∼0.0	∼0.0
4 (106-107)	Ca-Ga	60581.998	17.70	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60513.305	10.74	1	2.0	2.0
				$\Delta V_{tot} =$	2.0	2.0
1 (103-104)	Ga-Ga	60511.794	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60526.108	17.71	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
1 (103-104)	Ga-Ga	60511.683	14.31	1	10.0	10.0
2 (104-105)	Ga-Ca	60525.997	17.86	1	13.0	13.0
3 (105-106)	Ca-Ca	60543.872	25.09	1	7.0	7.0
				$\Delta V_{tot} =$	30.0	30.0
1 (103-104)	Ga-Ga	60512.206	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60526.520	17.91	1	∼0.0	∼0.0
3 (105-106)	Ca-Ca	60544.427	25.08	1	∼0.0	∼0.0
4 (106-107)	Ca-Ga	60569.502	17.85	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

In Figure 10.12 the evolution of the champion solutions for each trial is shown. Note that trial one is missing due to data loss during the simulation. The results show that convergence to the solution of $\Delta V = 0.0\,\mathrm{m/s}$ for trial two and three takes large numbers of generations. The limit of 40,000 generations for the optimisation is just sufficient for getting to the optimal

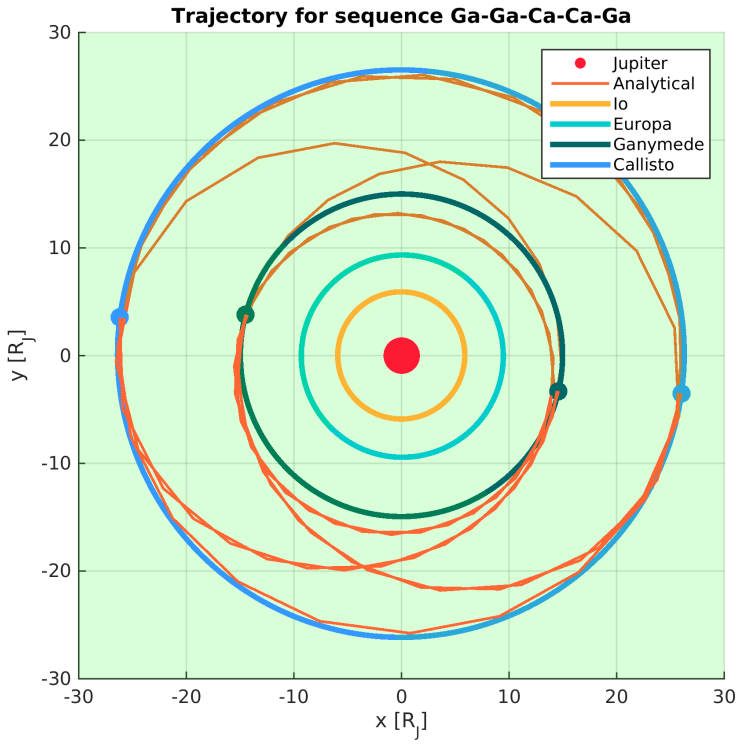

Figure 10.11: *Best trajectory solution to the Ga-Ga-Ca-Ca-Ga sequence (trial 2) with full freedom and two flyby window strategy.*

value. It shows that the five flyby problem is difficult to optimise, although the two trials here did converge in the end.

10.2.3 Result of Strategies

Before all strategies are analysed it is important to note that sequence one is dominated by thrust constraint violation penalties. Sequence two, on the other hand, has not experienced any thrust violations at all. A difference therefore exists in the goal of the optimisation. The penalties are related to the ratio of the maximum thrust violation with respect to the maximum thrust constraint, which is not a linear relation with ΔV due to the varying thrust constraint profiles, for example see Figure 10.3. Therefore the objective is shifted from optimising ΔV to optimising the thrust constraint violation penalties.

Although lowering the penalties is in essence also decreasing ΔV, it is not focussed on finding the optimal ΔV that is actually required. Instead it is optimising for the least stringent penalty until the solutions are satisfying the thrust constraint after which the objective is purely ΔV again (without penalties).

In the previous sections the results for the strategies full and small freedom with two flyby window have been shown and discussed. The remaining results of the simulations for the different strategies can be found in Appendix D. For each strategy the two tables, as seen in the previous sections, are shown with the corresponding data.

The moon transfers that involve the same moon suffer from the limited accuracy of the shaping method of the RFT fix as was with the discussion of the strategy for the first sequence. As shown before, the quality of the shaping method drops significantly when the same semi-major

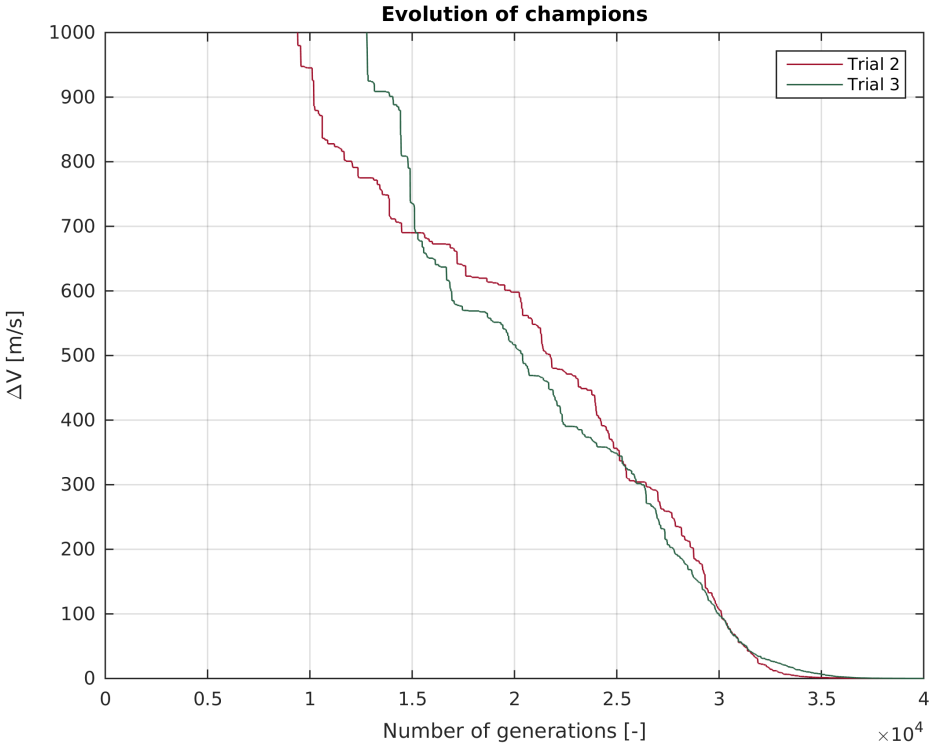

Figure 10.12: *Evolution of the champion for the Ga-Ga-Ca-Ca-Ga sequence with full freedom and two flyby window strategy.*

axis is used and the RAAN parameter changes by one degree already. This happens in several cases. Through optimisation, the population tends to converge to solutions where the RAAN is mostly rotated by the gravity assists. However, the latter does not always seem to work for moon transfers between different moons, especially after a resonance pattern while visiting the same moon, what happens in both sequences. This could also be due to insufficient convergence of the problem.

As was mentioned in the two discussed strategies in the previous sections, the hyperbolic excess velocity before the flybys $V_{-\infty}$ of the smaller subsets showed significant differences with the larger subsets. For these subsets also the flyby altitudes start dropping in order to make better use of the flyby. Several solutions show altitude drops to 50.0 km making them use the flyby to the maximum extent possible. A further distinction can be made between the two sequences for all strategies. For sequence one both subsets with two and three flybys show solutions that do not reoccur in the consecutive subsets whereas for sequence two this is only the case for the first subset. This leads to the question whether these smaller subsets are relevant for the other subsets, because they do not lead to similar results that help improve the optimisation. One of the reasons is that the first and second subset are relatively easy and provide solutions that are not viable for the larger subsets. The latter experience more difficulties with optimisation, due to the larger decision vector, the larger problem and the combination of many local optima with bad convergence performance. Therefore it can be said that the first two subsets do not necessarily improve the optimisation. Instead optimising the complete sub-sequence of five flybys at once is more time efficient, also because the results are quite varied. However, the latter is also due to the limited accuracy of the spherical shaping method.

To continue with $V_{-\infty}$ for the first sequence, both the two and three flyby window show good results for having nearly similar values of $V_{-\infty}$ for swinging by the same moon. The one

flyby window strategy on the other hand does not show this characteristic very well, which is most likely due to restricted freedom of the preceding flyby. This strategy can be translated as a patch method by adding one flyby add the time with sufficient freedom. However, the preceding flybys do not have sufficient freedom around there epochs to reach similar $V_{-\infty}$ values. On the other hand, for the second sequence all windows strategies are able to find similar $V_{-\infty}$ values, for both full and small freedom around the epochs of the flybys. As mentioned before with the two discussed test cases, the first and last flyby of a subset tend to show deviating values. Especially the last flyby shows deviating results compared to the other flybys in the subset with respect to the altitude. This was explained to be contributed to that these flybys are at the end of the sequence. Therefore they are not restricted to satisfy the next low-thrust arc.

Looking at the epochs of the first flyby of each subset of the solutions of sequence one, there are a few interesting observations. First, independent of the strategy, the epochs of the first flyby are considerably different of the various trials for the same strategy. This shows that the optimisation is probably not consistent in finding the optimal solution or there are too many local optima in which the DE algorithm gets stuck. It is also possible that the problem has not converged sufficient yet, even though a considerable large number of generations is run for each subset. For the second sequence the same observation is made regardless of the strategy and therefore it is considered to be characteristic of the problem to have many local optima. This was also expected during the preceding literature research due to the combination of continuous low-thrust legs and instantaneous gravity-assist manoeuvres. A second observation is that the epoch of the first flyby of the first subset is a leading reference value for the remaining subsets. Meaning that the consecutive subsets will start with an epoch close by the solution of the previous subset. This means that feeding the champion of the current solution has an influence on the following subsets. However, as mentioned before, the first subset (and even the second subset for the first sequence) have completely different results compared to the following subsets with respect to the utilisation of the GA's.

The time-of-flight in the solutions is often nearly equal or in the neighbourhood of the orbital period (same moon transfer) or of the synodic period (different moon transfer). An example for sequence one is the often reoccurring 2:1 (moon 7.10 days:spacecraft 3.55 days) resonance on the Eu-Eu transfer. Also the 1:1 resonance for the same moon sequence is present in several trials among the strategies. For the Eu-Io transfer the time-of-flight is in nearly all cases slightly larger than the synodical period (Table 9.3). The same or nearly the same TOF is most likely not a better configuration here, because the spacecraft does not repeat that same transfer in this sequence of the optimisation. Surprisingly for the second sequence not one of the TOF represents a resonance transfer. Although some of the TOF's come close to the orbital periods or synodic periods, there is not one that is nearly equal. This introduces an interesting question related to moon mapping of this sequence, because resonances are often used to map two opposite faces on the moon by going forth and back between two resonances Casalino (2012, 2014).

From the results it can be seen that the first two subsets, with two and three GA's, most of them give very good solutions. It is however when the sequence increases to four or five GA's that the quality drops significantly. Also the parameters seem to change more than by a slight deviation: like the hyperbolic excess velocity changes significantly and also the flyby altitudes start dropping in order to make better use of the flyby. Several solutions show altitude drops to 50.0 km making them use the flyby to the maximum extent possible.

To add to the subset story, especially the first two subsets of the sub-sequences seem not to contribute efficiently to the solution of the final subset. One of the reasons is that the first two subsets are relatively too easy and provide solutions that are not viable for the larger

subsets. The latter experiences more difficulties for optimisation, due to the larger decision vector, the larger problem and the combination of many local optima with bad convergence performance. Therefore it can be said that the first two subsets do not necessarily improve the optimisation, maybe even doing the complete sub-sequence of five flybys at once is more time efficient, also because the results are quite varied. However, the latter is also due to the limited accuracy of the spherical shaping method.

To further elaborate on the larger subsets with four and five flybys. Both subsets show that the problem is severely complex resulting, in case of sequence one, in enormous ΔV values. This is due to the dependency of the flybys and the low-thrust arcs. Each flyby that is being added to the subset increases the complexity and decreases the chance of finding proper trajectories in the system. The dependency on the previous and the following legs and flybys determines essentially the effective maximum number of flybys that can be optimised in one series. It is clear that for sequence one the subset with five flybys, but also four, has difficulties trying to find proper trajectories that do not need considerable amounts of ΔV. For sequence two, both subsets are still capable of finding proper solutions. However for larger sequences this might not work out anymore. A possible solution to this problem is either to split up the full sequence of the GTOC6 problem in smaller subsets of, for example, a maximum of five flybys. Another possibility would be to decouple the low-thrust arcs from the flybys. This can be done by introducing coasting arcs and powered gravity-assist manoeuvres. After finding the trajectory solution, the relative small thrust corrections during the powered flyby can be removed by replacing the coasting arc with a low-thrust arc. This approach has also been used by Novak (2012) to solve the Ea-Ea-Apolla trajectory discussed in Chapter 8.

So far the characteristics of the results of the strategies have been discussed. To get a better picture of the most suitable strategy for optimising ΔV, the objective function value is analysed of all strategies. In Figure 10.13 the results are shown for all strategies of the first sequence. It is clear from the graph that the subsets with four and five flybys have very large values compared to the smaller subsets. What is more difficult to see clearly is an optimal strategy for tackling complex low-thrust gravity-assist trajectories. The three trials (same color in graph) that are performed per strategy show no conclusive result for any of the strategies. Each of them displays different behaviour in the sense that one trial performs relatively well and another trial performs the worst.

In Figure 10.14 the same sequence and strategies are plotted, but now without the thrust constraint penalties. The graph displays the actual ΔV for the subsets. As mentioned before, optimising ΔV with penalties is not the same as optimising for the actual ΔV, which can be seen in the last figure. For example, the full freedom and three flyby window strategy has two trials that relatively do not perform well for optimisation of the former. However, looking at ΔV without the penalties there are now two trials of the same strategy that perform relatively well. Though, there is a difference in optimisation objective and therefore the latter results, without penalties, can not be used as conclusive results.

The second sequence showed much better results compared to sequence one. In Figure 10.15 the overview is given of the ΔV results for all strategies. One remarkable observation compared to the results of sequence one, is that the several trials of different strategies show a significant improvement of results going from the subset with four flybys to the subset of five flybys. In case of sequence one the results only became worse for the subset of five flybys.

Again the results show a varied outcome for the three trials of each strategy. Also the full freedom with three flyby window strategy does not perform well, relatively speaking. However, this is the case for the subset of four flybys. The consecutive subset of five flybys is improved significantly for two trials, whereas the result of the other trial became worse. Same as with

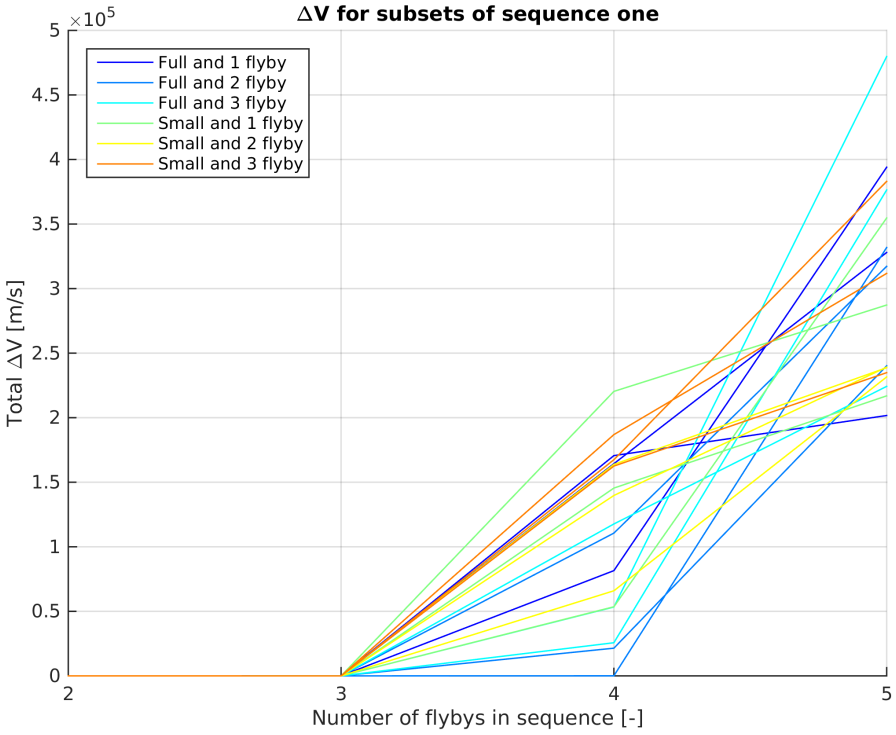

Figure 10.13: *The total ΔV of the subsets for all strategies for sequence one. Three trials per subset are displayed with the same color.*

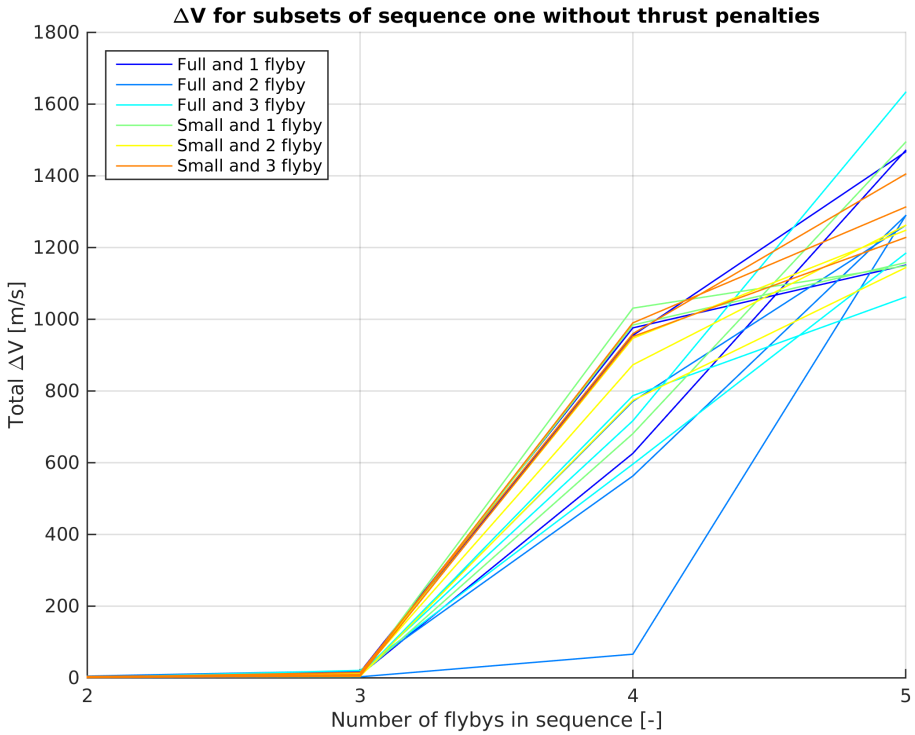

Figure 10.14: *The total ΔV of the subsets without thrust constraint violation penalties for all strategies for sequence one. Three trials per subset are displayed with the same color.*

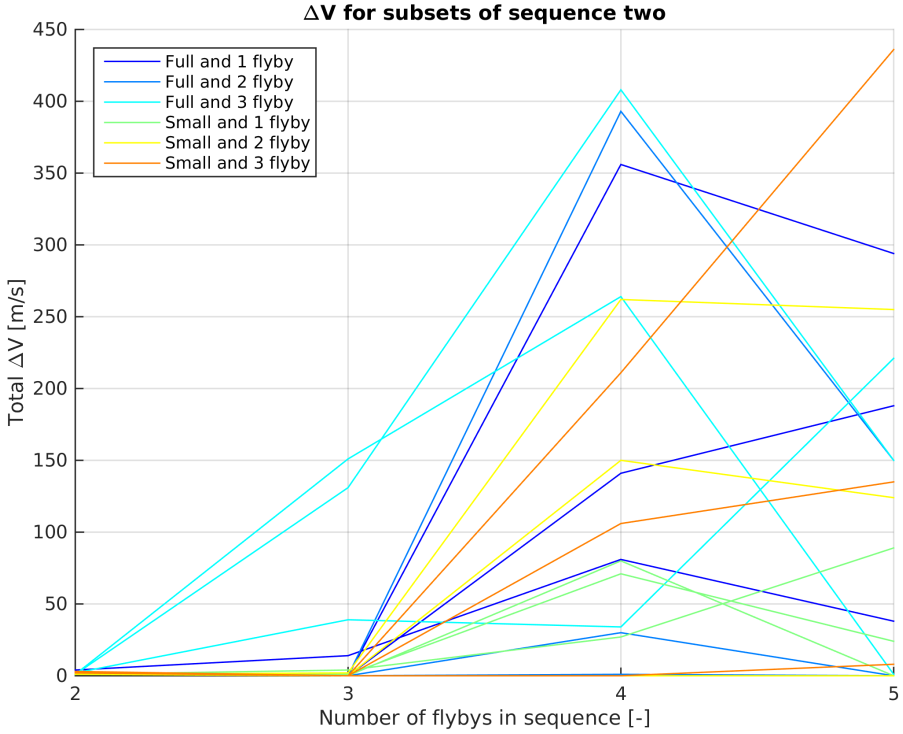

Figure 10.15: *The total ΔV of the subsets for all strategies for sequence two. Three trials per subset are displayed with the same color.*

sequence one it is hard to determine the performance of the different strategies. Therefore a closer look is taken at the robustness of the strategies.

10.2.4 Robustness of Strategies

To see how well the strategies performed the robustness is checked. Due to the varying solutions and not having one singular optimal value for which the robustness can be defined for as reference level, a different solution has to found. Here the best two trials out of three are taken and the difference is used to determine the level of robustness. The closer the difference to zero the more robust the strategy is to find optimal solutions. To define the two best solutions the total sum is taken of the ΔV of each subset of the sequence. In other words, not just the solution of the five flybys sequence counts, but also the smaller flyby sequence sizes. Although sequence one does not experience improvement in ΔV after adding a flyby (always increasing), this is the case for sequence two. To avoid taking only trials that perform very well on the five flybys subset, but not well on all of the preceding smaller subsets, it is decided to take the summation of ΔV of each subset in a trial. Doing so results in the two best trials of each strategy test for which the differences are shown in Figures 10.16 and 10.17 for respectively sequence one and two.

For sequence one it follows that the small freedom with two flyby strategy performs the most optimal. Furthermore, full freedom with either one or three flyby window gives the worst robustness. Closely followed by the small freedom with one flyby strategy, it can be concluded that for the first sequence using a one flyby window is not recommended. A window of two flybys seems to show the most promising results for robustness. Note that the robustness here is defined as the two best trials out of three which is a very limited selection. However, due to

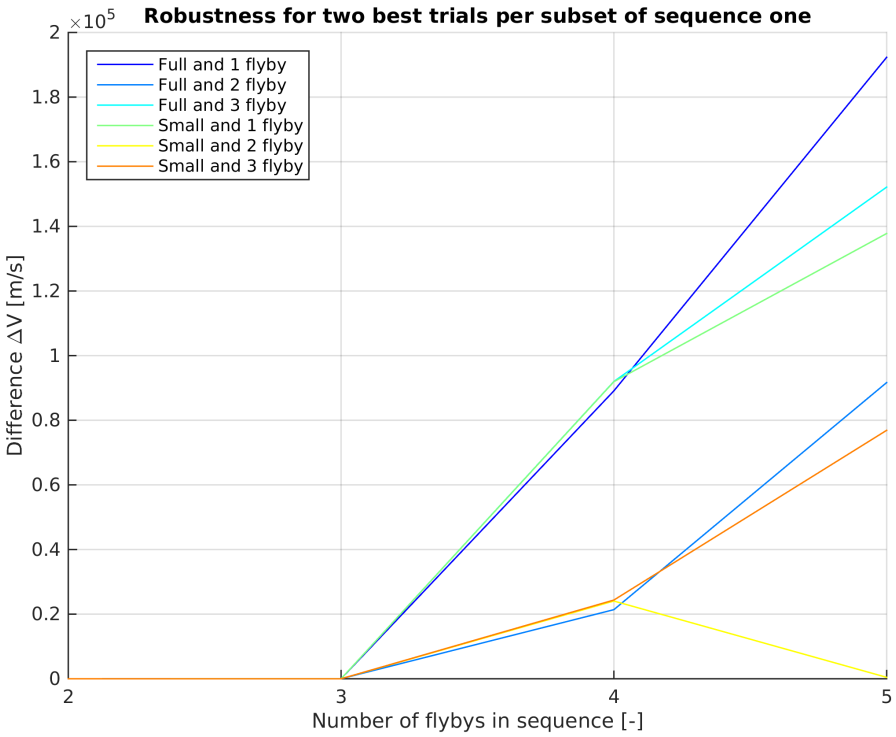

Figure 10.16: *Robustness of the strategies defined as the difference in ΔV of the two best trials per subset for sequence one.*

the required computational resources, as mentioned before, it was decided to stick with three trials.

Sequence two displays slightly different results compared to sequence one which are given in Figure 10.17. It is clear though that the full freedom with the three flyby strategy is the weakest of all strategies. Followed by the same freedom, but now with only one flyby window. Also the other three flyby window (small freedom) does not perform well compared to the others. This leads to the conclusions that sequence two does not favour the three flyby window strategy. Combining the latter with the conclusion of sequence one leaves one with the two strategies with a window of two flybys as potential optimal strategies. In case of sequence two the preference is the full freedom strategy whereas for sequence one it is the small freedom strategy. Note, as mentioned before that the number of trials is potentially not sufficient to give a conclusive answer to which strategy is the most optimal. However, based on these results in this section a window of two flybys is the optimal solution and both full and small freedom are able to perform well.

For the strategy choice for the optimisation of ΔV and moon face points in the next section a decision needs to be made. It was decided early on to optimise the first sequence, because of the fast dynamics and the possible challenge for the optimisation of the moon face points. However, this was decided when the sequences were not fixed yet halfway the thesis. The optimisation would be influenced by Europa due to the additional points that are awarded, because of its scientific interest. Though, with the sequence fixed this does not matter, still sequence one is selected for the next simulation. For this reason it is decide to go with the small freedom with two flyby window strategy, because it performed relatively well for sequence one.

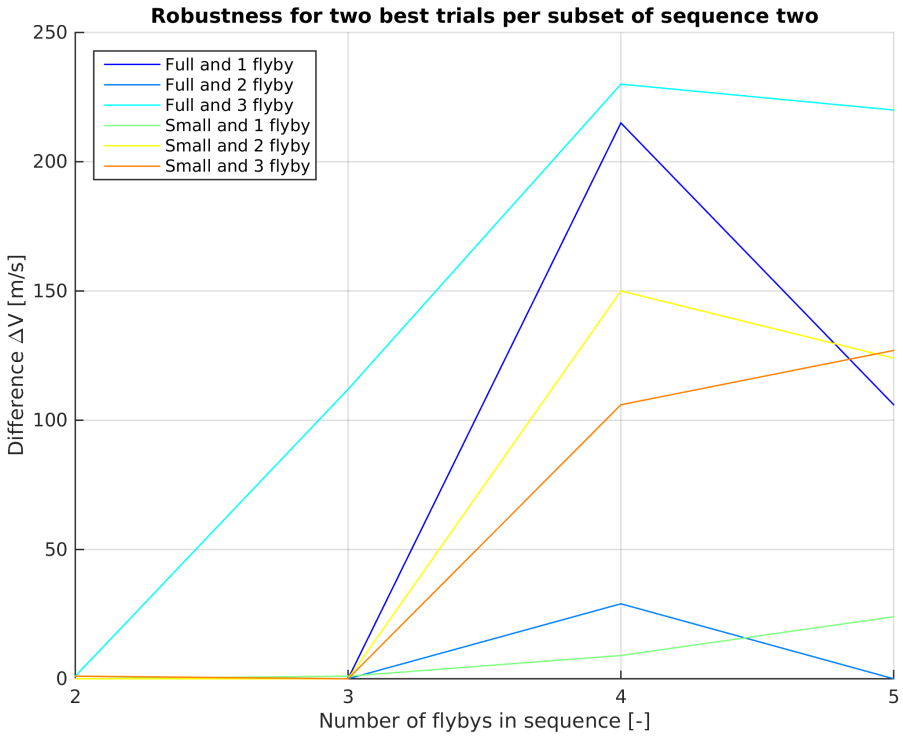

Figure 10.17: *Robustness of the strategies defined as the difference in ΔV of the two best trials per subset for sequence two.*

10.3 Optimisation of the Sequence One for ΔV and Moon Face Points

The optimisation for ΔV and moon face points is performed with a weighted sum objective function (Equation (9.6)), because the DE algorithm does not support multi-objective optimisation. For the problem three different sets of weights were used to solve the same problem (sequence one) again, but now also including the mapping of the moons. The three sets are $W_{\Delta V} = 10.0$ and $W_M = 1.0$, $W_{\Delta V} = 1.0$ and $W_M = 1.0$ and the last $W_{\Delta V} = 1.0$ and $W_M = 10.0$. With the first set the focus is put on optimising for ΔV and a little for the points. The last set of weights is the other way around, the focus is now on the moon points. Obviously, the second set is a compromise of both.

Furthermore, from the previous section the small freedom with two flyby strategy was selected. Note that only the first sequence is tested though, due to time constraints. In the last section of Appendix D the results are given for sequence one. In the following section the case with the weight set $W_{\Delta V} = 1.0$ and $W_M = 10.0$ is discussed, followed by the discussion on the results of the multi-objective optimisation.

10.3.1 $W_{\Delta V} = 1.0$ and $W_M = 10.0$ Applied to Sequence One

The last optimisation case is to test whether the designed low-thrust gravity-assist model can be applied to find the most valuable moon faces present in the sequence. In this section the results will be discussed, that was focussed mainly on optimising the moon face points due to the weight $W_M = 10.0$. Optimisation of ΔV is considered to be less of importance in this case. The results of the optimisation can be found in Tables 10.6 and 10.7.

The same kind of data is shown in the tables as was the case with the tables in the previous section. There are two additional columns in Table 10.6 that give the face number and the corresponding value (points) that were mapped with the flyby. Furthermore, in Table 10.7 besides the usual total ΔV values, the function value f for the combined objectives of ΔV and moon face points is given. Note that the more negative f is the more optimal the result is.

The first thing that is noticed from the results is again the very large ΔV values. This time the values are even larger as compared to the similar optimisation of ΔV in Section 10.2.1. The explanation is that the optimisation is more focussed on trying to find trajectories that reach the high scoring faces of the moons, which is done considerably well when looking at the column of the face values in Table 10.6. Even better, each subset within each trial of this test case has reached the maximum score possible. Europa has faces with a maximum value of six whereas Io has a maximum face value of three. Therefore the set of weights has performed well for the moon face points. However, this is at the cost of more ΔV.

Compared to the test case in Section 10.2.1, the TOF, $V_{-\infty}$ and h_{flyby} show more variation in the values to reach certain moon faces. Again the first two subsets confirm that their values in $V_{-\infty}$ and h_{flyby} are very different from the larger subsets. It is also interesting to see that the initial two subsets tend to often find face number 25, whereas in the final two subsets they disappear. Except for the subset with five flybys in trial one. Furthermore, the ΔV is a factor five or six higher. Though the actual ΔV required is only a factor two larger, the reason for the high values is related to the thrust levels. In Figures 10.19 - 10.22 the thrust profiles are given of each subset. From this follows that for the larger subsets (four and five flybys) all legs fail for the thrust constraint. Also the magnitude of the thrust is high and therefore explains the enormous ΔV. Also the consequence is that the mass of the spacecraft decreases real fast due to the excessive amounts of propellant used for these manoeuvres. Finally in Figure 10.18 the trajectory is shown of trial 2.

10.3.2 Performance of Sets of Weights

Looking at the results it is immediately clear that the relative set of weights have a desirable effect with respect to one point, scoring moon points. In each solution the maximum score is achieved, however, the cost is low-thrust arcs that violate the thrust constraint severely causing considerable large penalties for ΔV. Apparently the weights are not chosen carefully to allow for proper ΔV optimisation. The focus is too much on moon points, causing the optimiser to be less strict finding proper ΔV solutions. Therefore the optimisation with respect to moon points does not reveal a considerable amount of insight, except that the ΔV is influenced by the weights. However, the better set of weights can be derived from the results.

Instead of looking at the objective value f, the total ΔV is used to check the performance of the optimisation. This is because the maximum number of points is achieved for all cases and therefore only ΔV is changing the objective value. However ΔV is more clear and insightful than an objective value that combines ΔV and moon face points. In Figure 10.23 the total ΔV is shown per subset. It is clear that being fully focussed on the moon mapping with the weights $W_{\Delta V} = 1.0$ and $W_M = 10.0$ causes the highest ΔV results. However, the solutions to $W_{\Delta V} = 1.0$ and $W_M = 1.0$ follow closely. The last set is $W_{\Delta V} = 10.0$ and $W_M = 1.0$ which has a varied performance divided among the other results. With the given sets the performance between optimising ΔV and moon face points has not been fully explored. This would require more tuning of the weights to get better ratios to optimise for. However, due to time constraints this is not performed, but it is considered for future work. To add on that, it

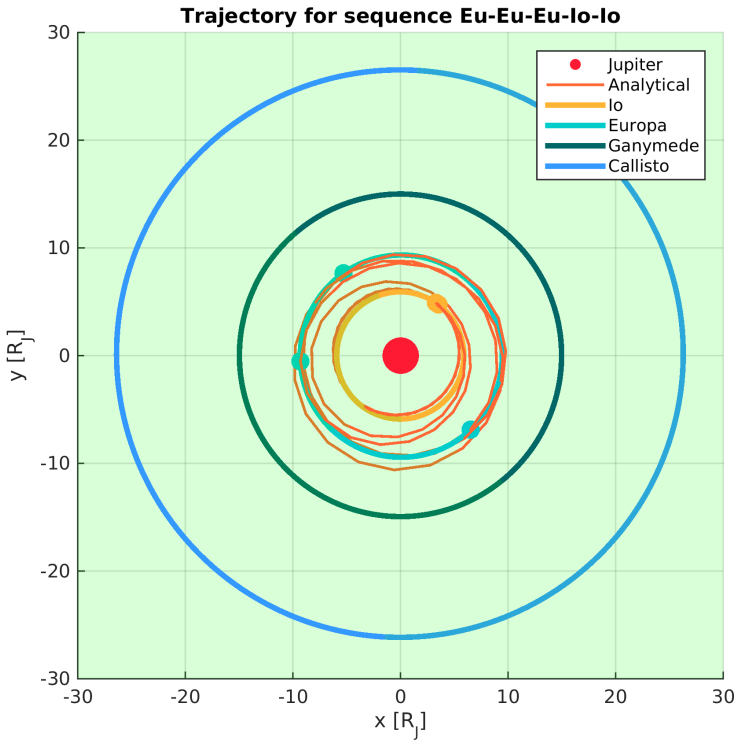

Figure 10.18: *Best trajectory solution to the Eu-Eu-Eu-Io-Io sequence (trial 2) with* $W_{\Delta V} = 1.0$ *and* $W_M = 10.0$.

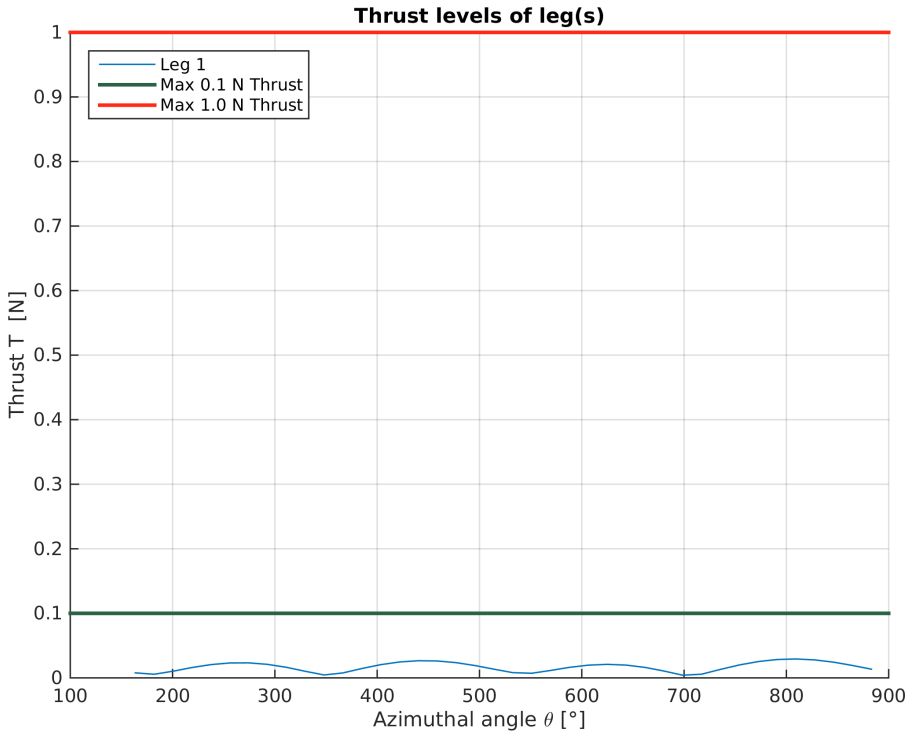

Figure 10.19: *Thrust profile of the subset Eu-Eu (trial 1) for sequence one with moon mapping.*

Table 10.6: *Results (3x) of subsets for the gravity-assist manoeuvres for the sub-sequence Eu-Eu-Eu-Io-Io. [Settings: small freedom and 2 flyby window and $W_{\Delta V} = 1.0$ and $W_M = 10.0$]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	# Face	Face value	$m_{post\ flyby}$ [kg]
1 (15)	Eu	59885.256	302.3	452.3	18	6	1889.67
2 (16)	Eu	59892.358	301.8	1532.4	24	6	1886.31
1 (15)	Eu	59886.738	305.6	1914.3	20	6	1889.67
2 (16)	Eu	59890.301	300.0	56.8	25	6	1885.75
3 (17)	Eu	59893.858	300.0	898.2	18	6	1883.23
1 (15)	Eu	59885.139	439.4	95.3	24	6	1889.67
2 (16)	Eu	59890.310	565.7	1499.2	26	6	1861.02
3 (17)	Eu	59894.264	1232.9	50.4	22	6	1817.12
4 (18)	Io	59898.084	2176.6	1478.6	19	3	1736.04
1 (15)	Eu	59884.302	507.6	472.1	16	6	1889.67
2 (16)	Eu	59889.006	624.9	1574.3	25	6	1850.76
3 (17)	Eu	59894.355	1078.3	52.6	23	6	1802.83
4 (18)	Io	59898.372	1261.9	50.0	22	3	1686.28
5 (19)	Io	59900.763	1317.6	800.2	26	3	1654.42
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	# Face	Face value	$m_{post\ flyby}$ [kg]
1 (15)	Eu	59883.221	303.8	108.4	25	6	1889.67
2 (16)	Eu	59890.324	305.1	967.8	18	6	1886.48
1 (15)	Eu	59881.526	301.4	767.4	25	6	1889.67
2 (16)	Eu	59888.629	300.0	547.3	18	6	1886.27
3 (17)	Eu	59895.730	300.0	856.4	24	6	1883.00
1 (15)	Eu	59882.606	380.1	1998.5	24	6	1889.67
2 (16)	Eu	59889.247	417.5	1999.7	17	6	1865.08
3 (17)	Eu	59893.704	1723.3	268.1	16	6	1795.55
4 (18)	Io	59898.045	1868.8	1099.1	20	3	1732.46
1 (15)	Eu	59883.125	1121.8	1914.1	16	6	1889.67
2 (16)	Eu	59888.537	900.1	74.5	17	6	1852.04
3 (17)	Eu	59894.353	1212.9	50.4	21	6	1786.63
4 (18)	Io	59898.505	1501.3	1288.4	15	3	1671.26
5 (19)	Io	59902.034	1494.2	384.5	24	3	1661.33
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	# Face	Face value	$m_{post\ flyby}$ [kg]
1 (15)	Eu	59865.445	357.8	1594.5	25	6	1889.67
2 (16)	Eu	59872.545	355.7	124.4	19	6	1886.22
1 (15)	Eu	59866.397	304.0	716.6	18	6	1889.67
2 (16)	Eu	59873.496	303.2	53.6	25	6	1884.93
3 (17)	Eu	59880.602	300.0	1339.5	19	6	1879.99
1 (15)	Eu	59865.985	369.3	414.0	19	6	1889.67
2 (16)	Eu	59872.868	318.8	351.4	24	6	1864.82
3 (17)	Eu	59879.875	1236.8	52.1	20	6	1785.03
4 (18)	Io	59883.890	1840.8	1319.2	19	3	1703.98
1 (15)	Eu	59864.636	469.1	1488.1	20	6	1889.67
2 (16)	Eu	59871.666	1161.9	50.0	17	6	1828.72
3 (17)	Eu	59876.934	1645.8	54.1	19	6	1754.36
4 (18)	Io	59880.576	1583.2	440.9	20	3	1654.66
5 (19)	Io	59883.209	1548.6	1036.5	19	3	1630.17

[1] Number between parenthesis is the actual flyby number in the CSU solution.

might be very useful to try out an actual multi-objective optimiser. PaGMO contains such an optimiser in the flavour of MOEA/D (Mambrini and Izzo, 2014), which creates a Pareto front with solutions to the sub-problems. Here the sub-problems are each separate objective function. The optimal Pareto front that follows from the optimisation is the line that denotes the optimal solution to the multiple objectives. The user than has to define the compromise of which objective is more important and read the corresponding optimal solution from the

Table 10.7: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: full freedom and 2 flyby window and $W_{\Delta V} = 1.0$ and $W_M = 10.0$]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59885.256	7.10	1	6.0	6.0
$f =$	-10.1527			$\Delta V_{tot} =$	6.0	6.0
1 (15-16)	Eu-Eu	59886.738	3.56	1	12.0	12.0
2 (16-17)	Eu-Eu	59890.301	3.56	1	11.0	11.0
$f =$	-10.0794			$\Delta V_{tot} =$	23.0	23.0
1 (15-16)	Eu-Eu	59885.139	5.17	1	42776.0	255.0
2 (16-17)	Eu-Eu	59890.310	3.95	1	200383.0	455.0
3 (17-18)	Eu-Io	59894.264	3.82	1	551468.0	882.0
$f =$	-8.7500038			$\Delta V_{tot} =$	794627.0	1592.0
1 (15-16)	Eu-Eu	59884.302	4.70	1	91306.0	383.0
2 (16-17)	Eu-Eu	59889.006	5.35	1	114024.0	488.0
3 (17-18)	Eu-Io	59894.355	4.02	1	931540.0	1297.0
3 (18-19)	Io-Io	59898.372	2.39	1	147156.0	340.0
$f =$	-8.0000031			$\Delta V_{tot} =$	1284026.0	2508.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59883.221	7.10	1	4.0	4.0
$f =$	-10.2066			$\Delta V_{tot} =$	4.0	4.0
1 (15-16)	Eu-Eu	59881.526	7.10	1	6.0	6.0
2 (16-17)	Eu-Eu	59888.629	7.10	1	5.0	5.0
$f =$	-10.1499			$\Delta V_{tot} =$	11.0	11.0
1 (15-16)	Eu-Eu	59882.606	6.64	1	25157.0	231.0
2 (16-17)	Eu-Eu	59889.247	4.46	1	451260.0	721.0
3 (17-18)	Eu-Io	59893.704	4.34	1	273688.0	676.0
$f =$	-8.7500040			$\Delta V_{tot} =$	750105.0	1628.0
1 (15-16)	Eu-Eu	59883.125	5.41	1	71265.0	368.0
2 (16-17)	Eu-Eu	59888.537	5.82	1	367653.0	681.0
3 (17-18)	Eu-Io	59894.353	4.15	1	1101802.0	1267.0
3 (18-19)	Io-Io	59898.505	3.53	1	67.0	67.0
$f =$	-8.0000026			$\Delta V_{tot} =$	1540787.0	2384.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59865.445	7.10	1	7.0	7.0
$f =$	-10.1319			$\Delta V_{tot} =$	7.0	7.0
1 (15-16)	Eu-Eu	59866.397	7.10	1	7.0	7.0
2 (16-17)	Eu-Eu	59873.496	7.11	1	9.0	9.0
$f =$	-10.1120			$\Delta V_{tot} =$	16.0	16.0
1 (15-16)	Eu-Eu	59865.985	6.88	1	217.0	217.0
2 (16-17)	Eu-Eu	59872.868	7.00	1	299949.0	814.0
3 (17-18)	Eu-Io	59879.875	4.02	1	580679.0	898.0
$f =$	-8.7500034			$\Delta V_{tot} =$	880845.0	1929.0
1 (15-16)	Eu-Eu	59864.636	7.03	1	149539.0	617.0
2 (16-17)	Eu-Eu	59871.666	5.27	1	328594.0	787.0
3 (17-18)	Eu-Io	59876.934	3.64	1	813988.0	1134.0
3 (18-19)	Io-Io	59880.576	2.63	1	80575.0	258.0
$f =$	-8.0000029			$\Delta V_{tot} =$	1372696.0	2796.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

Pareto front. The multi-objective optimiser is not part of this thesis, but recommended for further research.

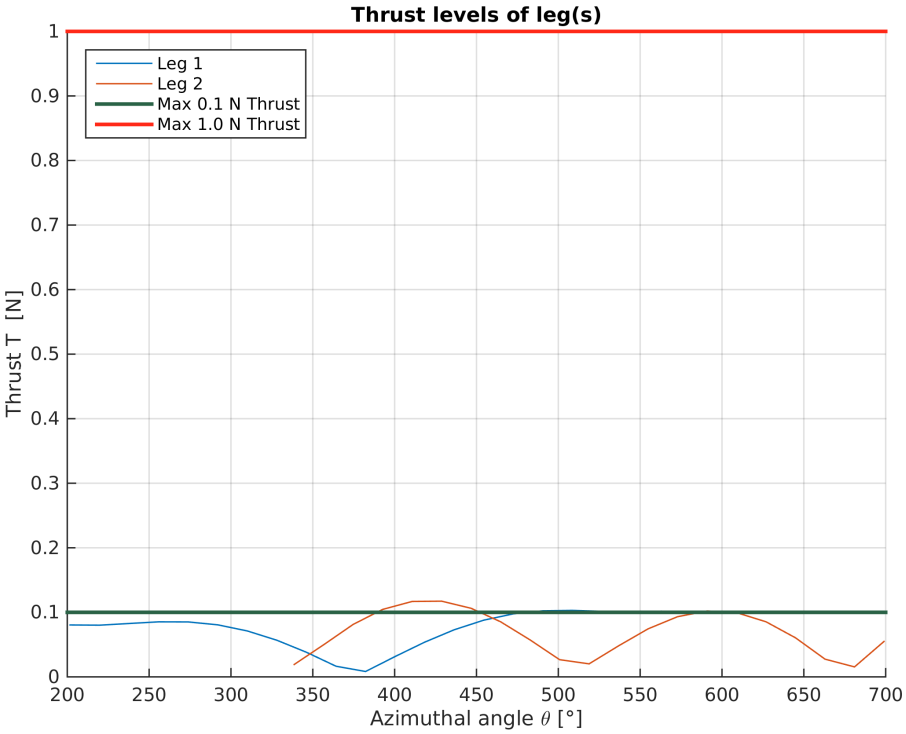

Figure 10.20: *Thrust profile of the subset Eu-Eu-Eu (trial 1) for sequence one with moon mapping.*

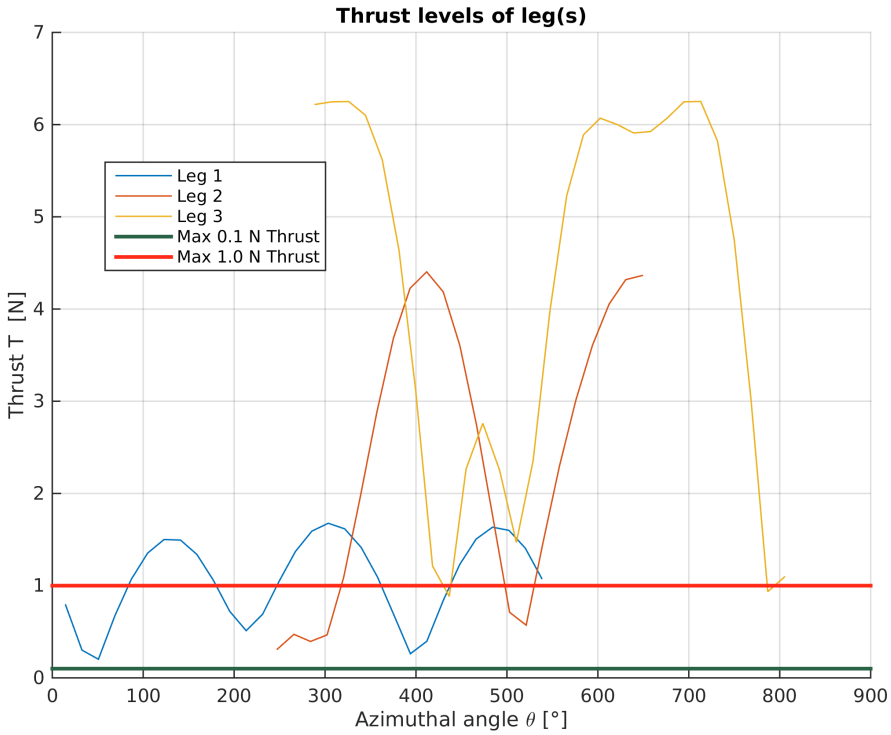

Figure 10.21: *Thrust profile of the subset Eu-Eu-Eu-Io (trial 1) for sequence one with moon mapping.*

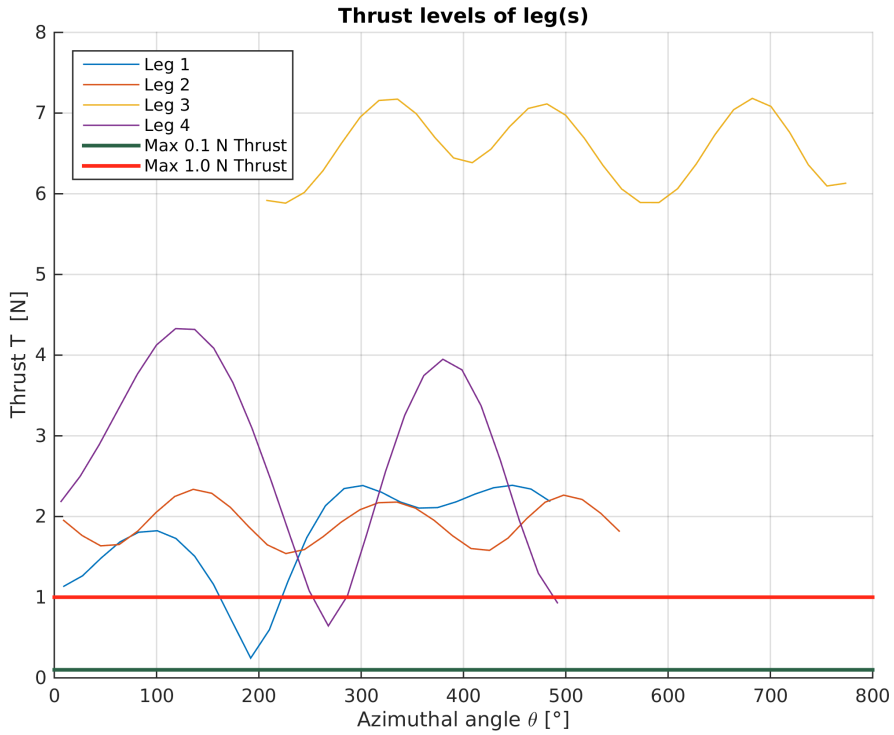

Figure 10.22: *Thrust profile of the subset Eu-Eu-Eu-Io-Io (trial 1) for sequence one with moon mapping.*

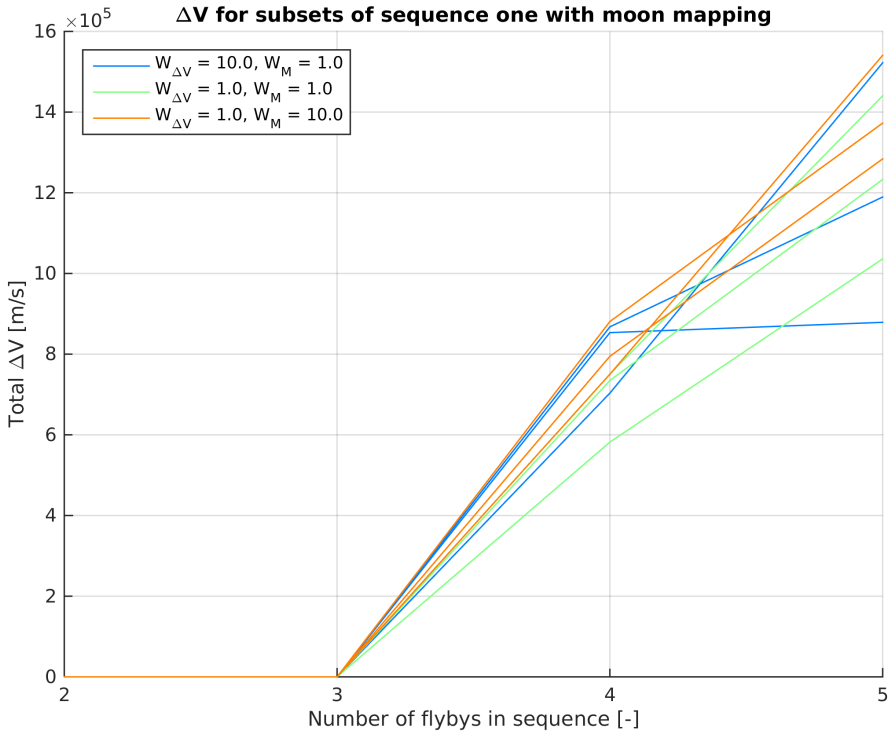

Figure 10.23: *The total ΔV of the subsets for all strategies for sequence one with moon mapping.*

11 Conclusions & Recommendations

This chapter is the closure of the thesis and gives the conclusions of this research. Also recommendations are mentioned and future work is proposed to improve the research and methodology.

11.1 Conclusions

The focus of this thesis is on preliminary trajectory optimisation in complex moon systems. More specifically, the main interest is focussed on finding effective ways to optimise moon sequences using low-thrust propulsion and flybys. This was phrased into the following research question which will be answered here:

Evaluate the quality of different optimisation strategies for Galilean moon mapping trajectories using one-body dynamics, gravity-assist manoeuvres and low-thrust propulsion.

The thesis work started with an extensive analysis of the current best known solution to the GTOC6 problem by the Chinese CSU. From this followed various interesting observations. First the low-thrust propulsion is used very limited during the mapping phase of GTOC6. The flybys take care of the largest changes in the orbital parameters. Furthermore, the CSU solution was able to map nearly all the moon faces using inclinations up till 12.0° and eccentricities up till 0.35.

After the analysis the spherical shaping method was verified to see whether it was able to create trajectories with sufficient accuracy, similar to the ones of CSU. The spherical shaping method developed for Tudat by Roegiers (2014) had several drawbacks. First the thrust could not be constrained and therefore had to be checked afterwards. During optimisation this resulted in a large number of solutions being discarded afterwards for violating the thrust constraint. The overall performance of the method is that trajectories can be made for inclinations up till 45.0° and eccentricities up till 0.45. Furthermore, the method experiences inaccuracies in ΔV for increasing inclinations and for differing and varying values for departure and arrival RAAN, as well as the argument of periapsis and the true anomaly. Roegiers (2014) developed a reference frame transformation that improved the accuracy. Also other problems like not being able to have a coasting arc in an inclined plane were solved with the fix. However, the spherical shaping method still suffers from limited accuracy.

A low-thrust multiple gravity-assist optimisation model was developed for the GTOC6 problem. Here the gravity-assist model from Tudat was used and also the half-plane test to determine which moon face was mapped by the flyby was developed and incorporated into the model. With the model an extensive test was performed to select a suitable DE algorithm. It followed from two test cases that the large complexity for gravity-assist sequences favours random selected control parameters during the evolution. This feature was made possible due to an incorrect implementation of the adaptive algorithm jDE while testing the algorithms. The implementation is named randDE. Furthermore, it followed that the combination of low-thrust arcs and gravity-assist manoeuvres would cause problems with convergence due to the large decision vectors and the slow convergence for the test problems.

With the selected optimisation algorithm randDE the GTOC6 problem was prepared. A framework was developed to test several optimisation strategies. Two sequences of the CSU solution were taken to be tested for two different optimisation cases. One on the optimisation of ΔV and another one on the optimisation of ΔV and the moon face points.

Several optimisation strategies with different window sizes have been applied allowing for a wide variation of the freedom around the epochs of the flybys. With sequence one, Eu-Eu-Eu-Io-Io, showing very unsatisfying solutions due to thrust constraint violation. Sequence two, Ga-Ga-Ca-Ca-Ga showed more promising results that were similar to the CSU solution. From the results it followed that the strategies made it possible to find the gravity assists needed to avoid using low-thrust propulsion. However, not very consistent answers were found, and the initial two subsets of the computations for the remaining subsets are most of the time not a leading factor for the optimisation. Therefore the two subsets should be considered to be removed. However, the impact of that on the epochs of the remaining subsets has not been tested. For the optimisation with the different strategies there was not a conclusive answer to which strategy is the most optimal. From the results did follow that using a window of two flybys is the most suitable option. Whether the full or small freedom should be applied was left in the middle. For the final problem the small freedom strategy was used due to the fact that that strategy performed well in sequence one.

Furthermore, sequences with four or five GA's are considered complex and the size of the decision vector and problem create performance issues for (rand)DE. The current approach of increasing the subsets and optimising everything again is not very efficient and does likely not yield better results compared to just running the five GA problem in the first place. However, this was not verified.

Also using low-thrust in this stage of optimisation might not be the most suitable solution for tackling these large complex moon problems. Especially, because the low thrust is not used for major trajectory changes. This is done by the flybys and therefore it might be more preferable to use Lambert (ballistic) arcs with powered gravity assists to scope the search space. This also decouples the legs from the flybys in order to improve the performance of finding proper trajectories. Afterwards the powered gravity assist can be removed by replacing the Lambert arcs with low-thrust arcs using a local optimiser. Compared to ballistic legs, the low-thrust approach creates significant complexity for preliminary optimisation. The continuous spectrum of solutions of the low-thrust arc combined with the infinitely possibilities for the conditions before the flyby make the solution space a true maze for the optimiser.

The final optimisation of ΔV and the moon face points turned out to work very well for mapping the faces of the moons. In all cases the maximum score was achieved for the moon faces. However, since the first sequence was used again, enormous values for ΔV were found. From the results it was concluded that the weight sets for the weighted sum function were not adequate enough to optimise the problem. This should be further investigated and it is recommended to do this with an actual multi-objective optimiser, like MOEA/D.

To summarise, the spherical shaping method is a decent and elegant algorithm for shaping low-thrust trajectories. However, the performance in its current implementation in Tudat is insufficient for complex low-thrust multiple gravity-assist problems with respect to computational effort and accuracy. Though for the optimisation of the second sequence very good results were achieved by several trials. These solutions required almost no thrust at all and therefore were not affected by the accuracy issues of the spherical shaping method. However, this was not the case for sequence one. From this it is also concluded that such complex moon problems should preferably be solved with ballistic arcs combined with powered gravity assists followed by a local optimiser as mentioned above. With respect to the optimisation strategy, using a two flyby window allowed for the more optimal solutions compared to the one and three flyby windows strategies. For the full or small freedom around the epochs of the flybys the result was a non-conclusive answer, because they both performed relatively well in one sequence and on average in the other sequence.

11.2 Recommendations

This thesis is a thorough analysis of optimisation of trajectories in complex moon systems. However, sufficient elements could have been done different or improved this thesis. These form the recommendations listed below .

Spherical Shaping Method

The first recommendation involves improvements to the spherical shaping code with respect to nodal change (RAAN). Large errors are introduced that have to be overcome with penalty functions to keep the unwanted solutions away. Preferably the low-thrust method should give solutions that have zero error as in that all solutions are equally treated by optimisation algorithms. The spherical shaping method is rather new and the same holds for the implementation of Tudat which already came with recommendations by Roegiers (2014). Several improvements on the shortcomings of the spherical shaping method are

1 find an improved solution for the reference frame transformation fix to avoid excessive ΔV for the trajectories. This could be partly the reason due to the assumption of only out-of-plane thrust and tangential thrusting. Radial thrusting is not present and although tangential is considered more efficient for escaping objects and to get far away with minimal effort, it might be that radial thrusting is the missing key element for better trajectories while rotating RAAN.

2 find a way to incorporate thrust limitations, currently the optimisation algorithm has to filter out the valid trajectories after computing them. Together with the many constraints, this severely limits the freedom of the optimiser. It is better that the spherical method is able to find proper trajectories that fit the maximum thrust profile.

3 use time as an independent variable for numerical verification (propagation). Currently, only the angular values can be converted after shaping the trajectory to the corresponding time steps. It would be convenient to create solutions with a certain fixed time step.

4 increase the accuracy ΔV computation without increasing the computation time significantly.

Following the previous recommendation, based on the results of this thesis it is recommended that the spherical shaping method in its current implementation is not suitable for problems where the departure and arrival conditions differ significantly. Even in the case of the low-thrust multiple gravity-assist trajectories it is debatable if the spherical shaping should be

used. The flyby can overcome the accuracy problems by taking the responsibility for the required change of the trajectory, though this is essentially taking out the low-thrust arc.

Optimisation

As mentioned before, the randDE algorithm is not really adaptive. The control parameters are defined randomly for every generation. This worked out well for the test problem used for DE tuning. However, it was assumed that the performance would be similar for the larger subsets. The latter are in essence different problems and therefore the randDE algorithm could actually perform worse compared to the actual adaptive algorithms of PaGMO.

The next point is that the convergence of the optimisation will be stopped after a certain maximum number of generations. This will be at the cost of more suitable solutions or better solutions, that could impact the remainder of the sequence. However, at the cost of computation time. Therefore a compromise is necessary to fulfil the optimisation requirement. Especially for the results of sequence one it is questionable, whether the maximum number of generations was sufficient.

With respect to optimisation, different optimisation algorithms should be tried out. For this thesis this was too elaborate and time consuming to test and verify. However it is shown by Izzo et al. (2009) that mixing different algorithms in many cases has similar if not an improved outcome. Also because the advantages are used of the many methods. This is a general optimisation recommendation, but especially applies for trajectory optimisation. Also, because of the complex and large search space and the resulting solution space with many local optima, it is recommended to not stick to one particular algorithm. Furthermore, PaGMO is equipped with these capabilities and is relatively easy to use and should definitely be looked into by future students researching complex trajectories. On top of that, from the results on tuning the DE algorithm it was concluded that adaptive algorithms should have the preference, because they save a significant amount of time on tuning which in most cases is very problem dependent.

Sequence optimisation

Although the optimisation model with its decision vector was designed well. There is always room for improvement. In specific the flyby angles defining the orientation of the hyperbolic excess velocity could be improved. Instead of angles defined in the Jovicentric frame the local moon centred frame should be used. This might improve the search space. The current angles have all kinds of values, because the angles are 'dependent' on position. At the normalised position [1,0,0] a 90 degrees rotation angle (planar) equals a similar velocity boost as a 270 degrees rotation at the position [-1,0,0]. Due to time constraints and efficiency of the problem algorithm it was decided not to work this out. Also the efficiency of the algorithm could actually be decreased, because additional computations are required to transform from moon centred to Jovicentric frame.

A suggestion to improve the optimisation of the subsets of the sequence is, to use the champion of the previous subset completely instead of using alone the epoch and TOF values as guidance for the next subset. The champion could be injected into the initial population of the next subset to advance the evolution. However, it should be noted that since the champion is based on a different problem (less flybys/legs), an adverse effect is also possible due to the injection of the champion.

Different sets of weights should be tried out for the multi-objective optimisation problem (moon face points). The sets used in this thesis were all strongly focussed on maximising the moon points. Also an actual multi-objective optimiser should be considered, like MOEA/D to get a better result and insight into the multi-objective problem.

Finally, optimising five flybys or even four is requiring enormous amounts of computational effort. Therefore, it should be looked into using a moving window going through the sequence. In other words the flybys that were outside the window should be fixed instead of given a small freedom. This might result in similar results while saving expensive computational time.

Branch and Bound

Branch and bound was originally planned to be used to generate the sequences of flybys. However, this part is not performed at all due to time constraints and being a considerable load of work together with the preceding work for a Master's thesis. This could become a thesis on its own to find optimal sequences. Same holds for the different moon hopping techniques as defined in the literature research (Hoving, 2014).

A

Galilean Moon System Data

This appendix holds the data of the Galilean moons and Jupiter needed for orbital calculations. The Keplerian orbit elements of the satellites can be found in Table A.1 whereas the physical constants of the moons are given in Table A.2. The remaining constants are displayed in Table A.3.

Table A.1: *Keplerian orbit elements of the Galilean Satellites at Epoch* $= 58\,849.0\,\mathrm{MJD}$. *(Petropoulos, 2012)*

Orbit Element	Unit	Io	Europa
Semi-major axis a	[km]	422029.68714001	671224.23712681
Eccentricity e	[-]	$4.308524661773 \times 10^{-3}$	$9.384699662601 \times 10^{-3}$
Inclination i	[°]	$40.11548686966 \times 10^{-3}$	0.46530284284480
LAN Ω	[°]	-79.640061742992	-132.15817268686
Arg. of pericentre ω	[°]	37.991267683987	-79571640035051
Mean anomaly M_0	[°]	286.85240405645	318.00776678240
Orbit Element	**Unit**	**Ganymede**	**Callisto**
Semi-major axis a	[km]	1070587.4692374	1883136.6167305
Eccentricity e	[-]	$1.953365822716 \times 10^{-3}$	$7.337063799028 \times 10^{-3}$
Inclination i	[°]	0.13543966756582	0.253554332731555
LAN Ω	[°]	-50.793372416917	86.723916616548
Arg. of pericentre ω	[°]	-42.876495018307	-160.76003434076
Mean anomaly M_0	[°]	220.59841030407	321.07650614246

Table A.2: *Satellite physcial constants. (Petropoulos, 2012)*

Satellite	Radius R_S [km]	μ_S [km^3/s^2]
Io	1826.5	5959.916
Europa	1561.0	3202.739
Ganymede	2634.0	9887.834
Callisto	2408.0	7179.289

Table A.3: *Other constants and conversions. (Petropoulos, 2012)*

Parameter	Unit	Value
Gravitational parameter of Jupiter μ	[km^3/s^2]	126686534.92180
Jupiter radius R_J	[km]	71492.0
Standard acceleration due to gravity g	[m/s^2]	9.80665
Day	[s]	86400
Year	[days]	365.25

B

The Football Grid

This appendix relates to the performance index used to map the Galilean satellites. The moons are mapped by dividing the surface into a football grid (uniform icosahedron) which has 32 faces, consisting of 20 pentagons and 12 hexagons. The coordinates of the vertices used in the 32 faces are defined in the body-fixed reference frame and are given in Table B.1. In Table B.2 the vertices can be found that make up each pentagon and hexagon. The sides of the pentagons and hexagons have a length of 2 and the vertices lie on a sphere. The golden ratio $p = \phi = \left(1 + \sqrt{5}\right)/2$ is used to define the radius of this sphere which equals to $r = \sqrt{9\phi + 10}$.

Table B.1: *Football grid coordinates,* $p = \phi = \left(1 + \sqrt{5}\right)/2$. *(Petropoulos, 2012)*

Vertex number	b_1	b_2	b_3
1	-3p	-1	0
2	-3p	1	0
3	-$(1+2p)$	-2	-p
4	-$(1+2p)$	-2	p
5	-$(1+2p)$	2	-p
6	-$(1+2p)$	2	p
7	-$(2+p)$	-1	-2p
8	-$(2+p)$	-1	2p
9	-$(2+p)$	1	-2p
10	-$(2+p)$	1	2p
11	-2p	-$(2+p)$	-1
12	-2p	-$(2+p)$	1
13	-2p	$(2+p)$	-1
14	-2p	$(2+p)$	1
15	-2	-p	-$(1+2p)$
16	-2	-p	$(1+2p)$
17	-2	p	-$(1+2p)$
18	-2	p	$(1+2p)$
19	-p	-$(1+2p)$	-2
20	-p	-$(1+2p)$	2
21	-p	$(1+2p)$	-2
22	-p	$(1+2p)$	2
23	-1	-2p	-$(2+p)$
24	-1	-2p	$(2+p)$
25	-1	0	-3p
26	-1	0	3p
27	-1	2p	-$(2+p)$
28	-1	2p	$(2+p)$
29	0	-3p	-1
30	0	-3p	1

Vertex number	b_1	b_2	b_3
31	0	3p	-1
32	0	3p	1
33	1	-2p	-$(2+p)$
34	1	-2p	$(2+p)$
35	1	0	-3p
36	1	0	3p
37	1	2p	-$(2+p)$
38	1	2p	$(2+p)$
39	p	-$(1+2p)$	-2
40	p	-$(1+2p)$	2
41	p	$(1+2p)$	-2
42	p	$(1+2p)$	2
43	2	-p	-$(1+2p)$
44	2	-p	$(1+2p)$
45	2	p	-$(1+2p)$
46	2	p	$(1+2p)$
47	2p	-$(2+p)$	-1
48	2p	-$(2+p)$	1
49	2p	$(2+p)$	-1
50	2p	$(2+p)$	1
51	$(2+p)$	-1	-2p
52	$(2+p)$	-1	2p
53	$(2+p)$	1	-2p
54	$(2+p)$	1	2p
55	$(1+2p)$	-2	-p
56	$(1+2p)$	-2	p
57	$(1+2p)$	2	-p
58	$(1+2p)$	2	p
59	3p	-1	0
60	3p	1	0

Table B.2: *List of vertices that make up each face of the football grid. (Petropoulos, 2012)*

Face number	Vertex numbers					
1	59	60	58	54	52	56
2	52	54	46	36	44	
3	18	10	8	16	26	
4	2	6	10	8	4	1
5	9	5	2	1	3	7
6	17	9	7	15	25	
7	43	51	53	45	35	
8	51	55	59	60	57	53
9	60	58	50	49	57	
10	58	54	46	38	42	50
11	4	8	16	24	20	12
12	1	4	12	11	3	
13	7	3	11	19	23	15
14	53	57	49	41	37	45
15	41	49	50	42	32	31
16	21	31	32	22	14	13
17	32	42	38	28	22	
18	38	28	18	26	36	46
19	24	34	44	36	26	16
20	20	24	34	40	30	
21	19	11	12	20	30	29
22	39	29	30	40	48	47
23	23	19	29	39	33	
24	23	33	43	35	25	15
25	37	27	17	25	35	45
26	37	41	31	21	27	
27	13	14	6	2	5	
28	14	22	28	18	10	6
29	48	40	34	44	52	56
30	47	48	56	59	55	
31	33	39	47	55	51	43
32	27	21	13	5	9	17

C

Leg Optimisation Results

This appendix holds the results of the test suite for tuning the DE algorithm and the adaptive variant. For each figure the settings for F and CR are found on the x- and y-axis in case of normal DE. In case of adaptive DE the 10 schemes applied are shown on the axis. The crossover scheme is set on the x-axis whereas the selection and mutation schemes can be found on the y-axis. The colour shows the number of occurrences (i.e. percentage) that the threshold has been passed for that setting and is an indication of the robustness of the tested algorithm. Inside the coloured circles there are two values. The top one equals the previous percentage value in case the colour is not clear. The bottom value indicates the number of generations until the threshold value has been reached. This threshold value is either set to the expected optimum or the that same with a certain deviation added.

C.1 Ea-Ea-Apollo with threshold of 8764.34 m/s

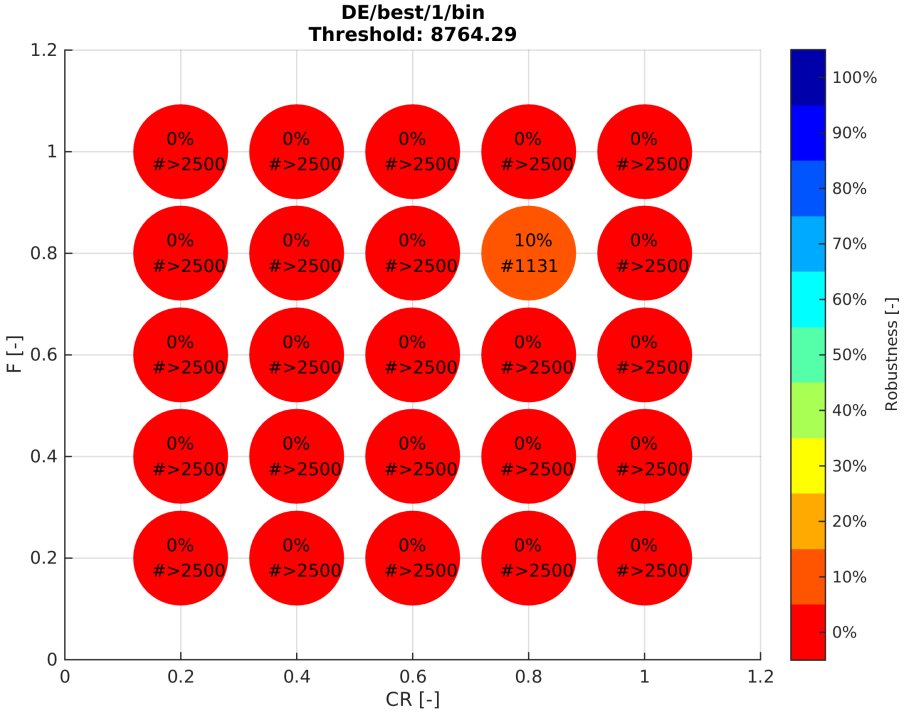

Figure C.1: *Optimisation tuning results for the best/1/bin scheme. Threshold for* $\Delta V = 8764.29\,\mathrm{m/s}$.

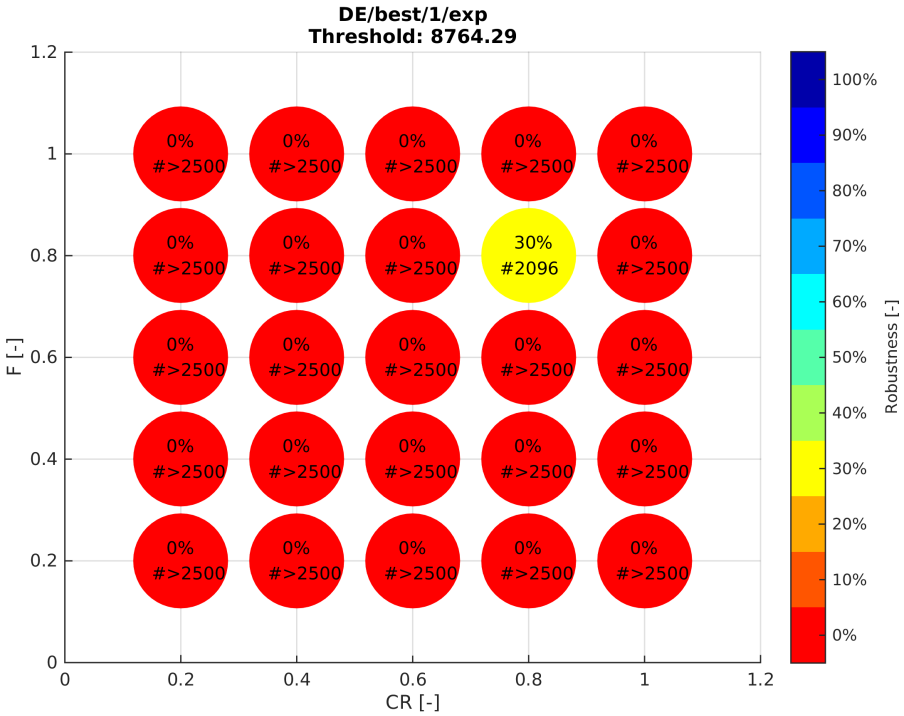

Figure C.2: *Optimisation tuning results for the best/1/exp scheme. Threshold for* $\Delta V = 8764.29\,\mathrm{m/s}$.

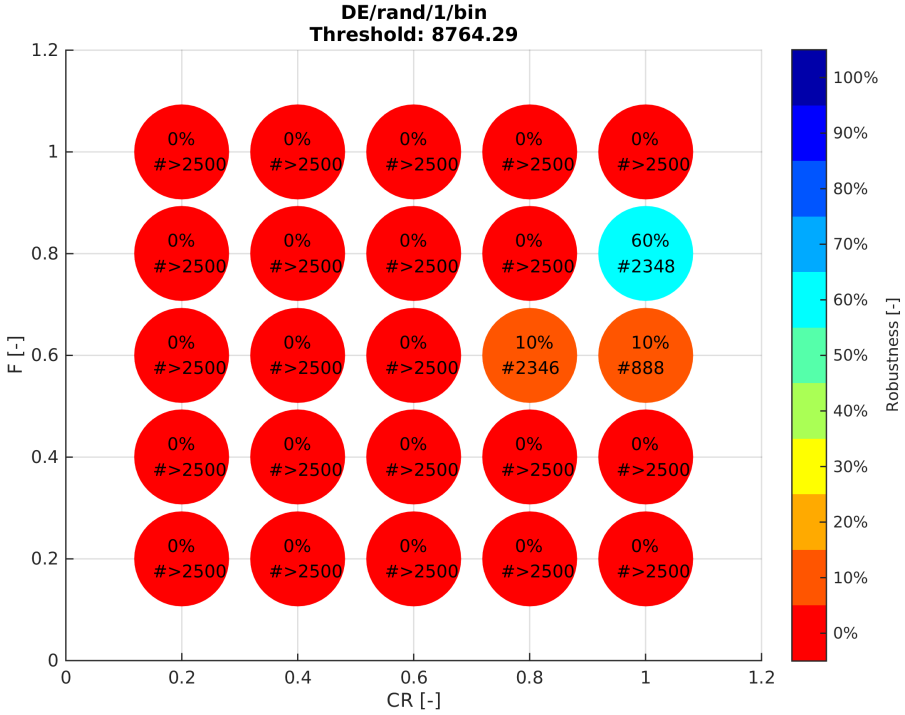

Figure C.3: *Optimisation tuning results for the rand/1/bin scheme. Threshold for* $\Delta V = 8764.29\,\mathrm{m/s}$.

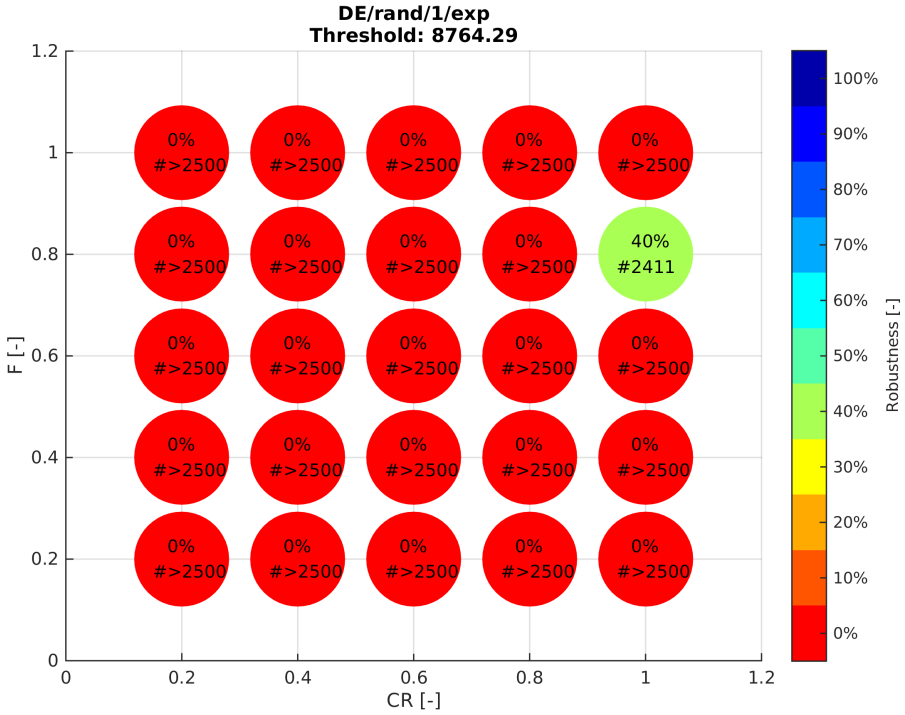

Figure C.4: *Optimisation tuning results for the rand/1/exp scheme. Threshold for* $\Delta V = 8764.29\,\mathrm{m/s}$.

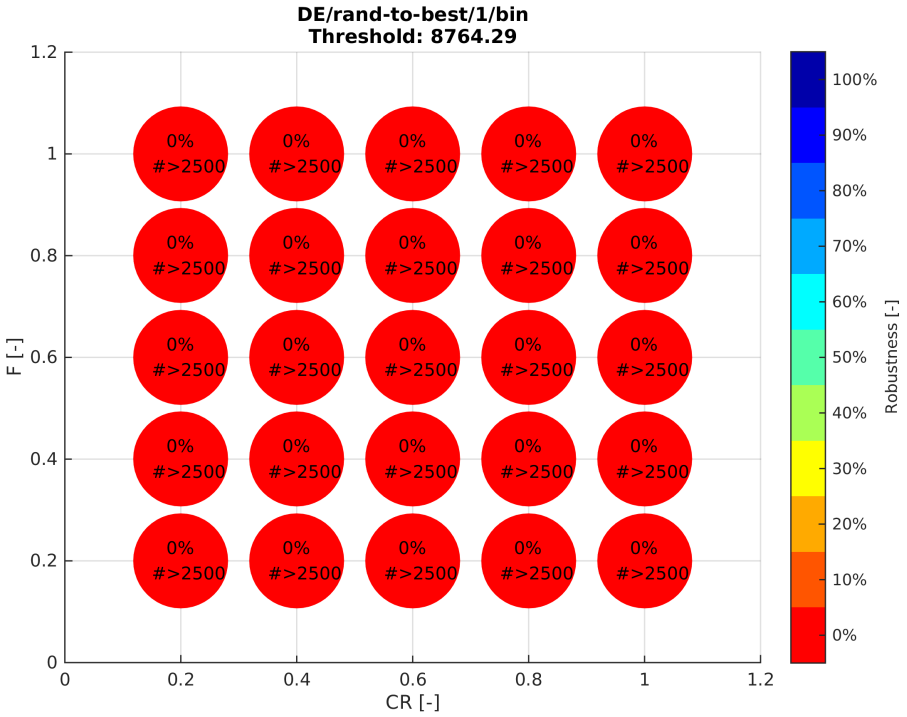

Figure C.5: *Optimisation tuning results for the rand-to-best/1/bin scheme. Threshold for* $\Delta V = 8764.29\,\mathrm{m/s}$.

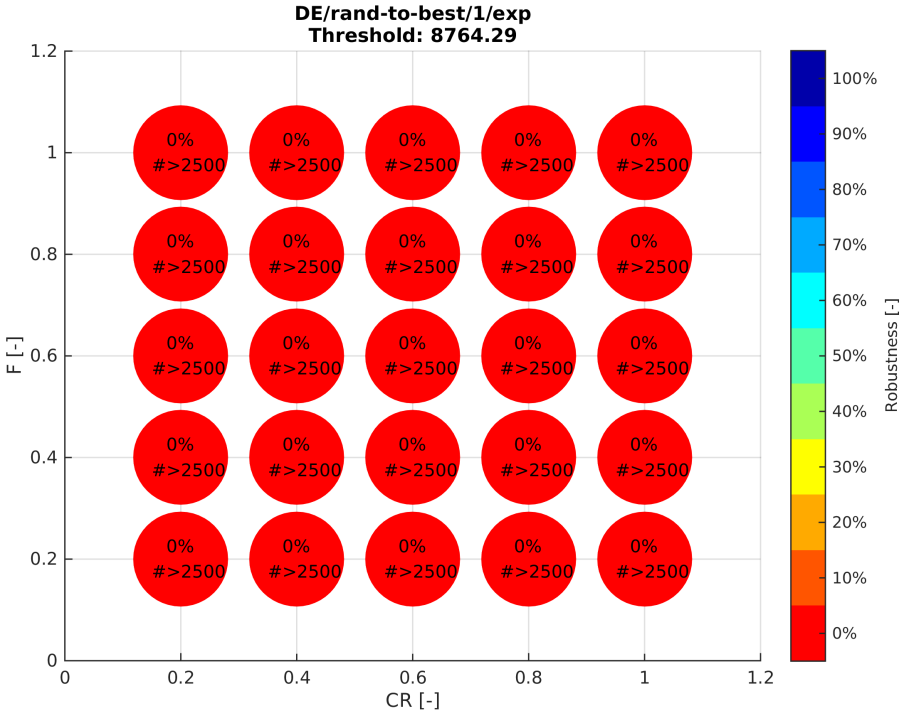

Figure C.6: *Optimisation tuning results for the rand-to-best/1/exp scheme. Threshold for* $\Delta V = 8764.29\,\mathrm{m/s}$.

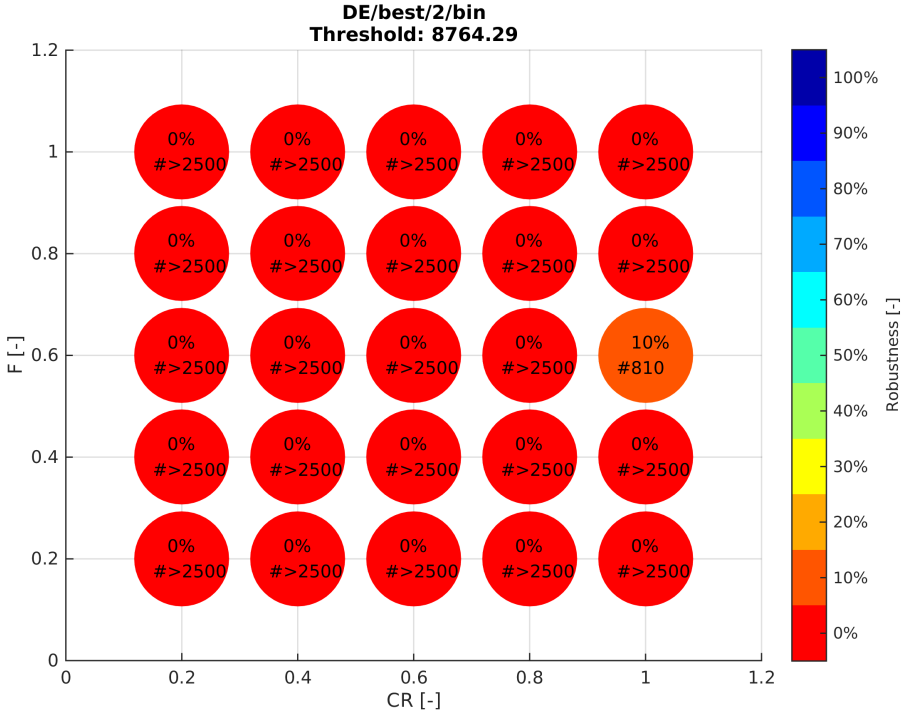

Figure C.7: *Optimisation tuning results for the best/2/bin scheme. Threshold for* $\Delta V = 8764.29\,\mathrm{m/s}$.

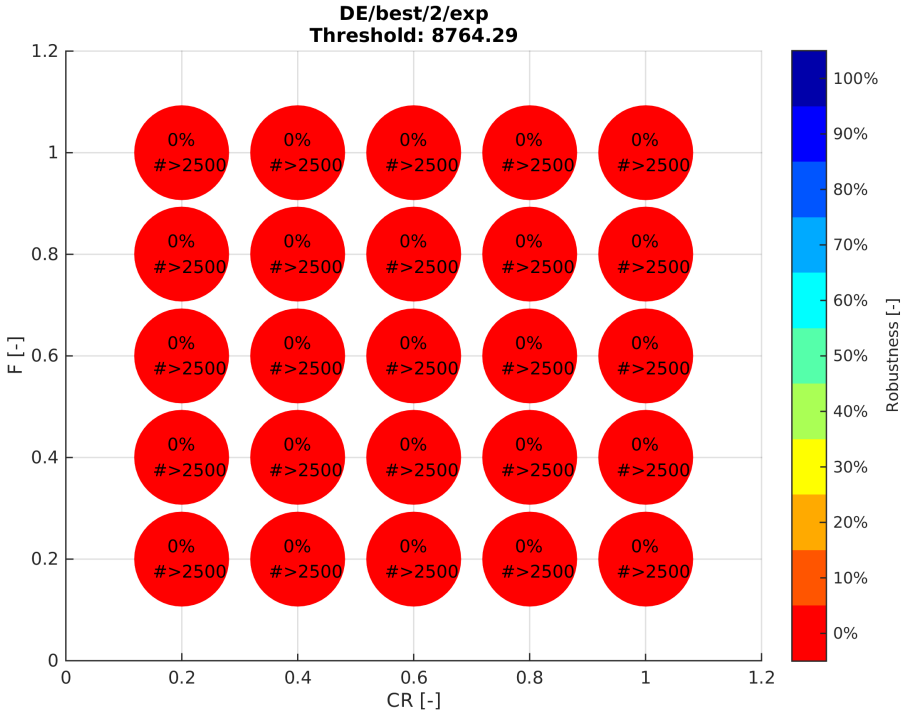

Figure C.8: *Optimisation tuning results for the best/2/exp scheme. Threshold for* $\Delta V = 8764.29\,\mathrm{m/s}$.

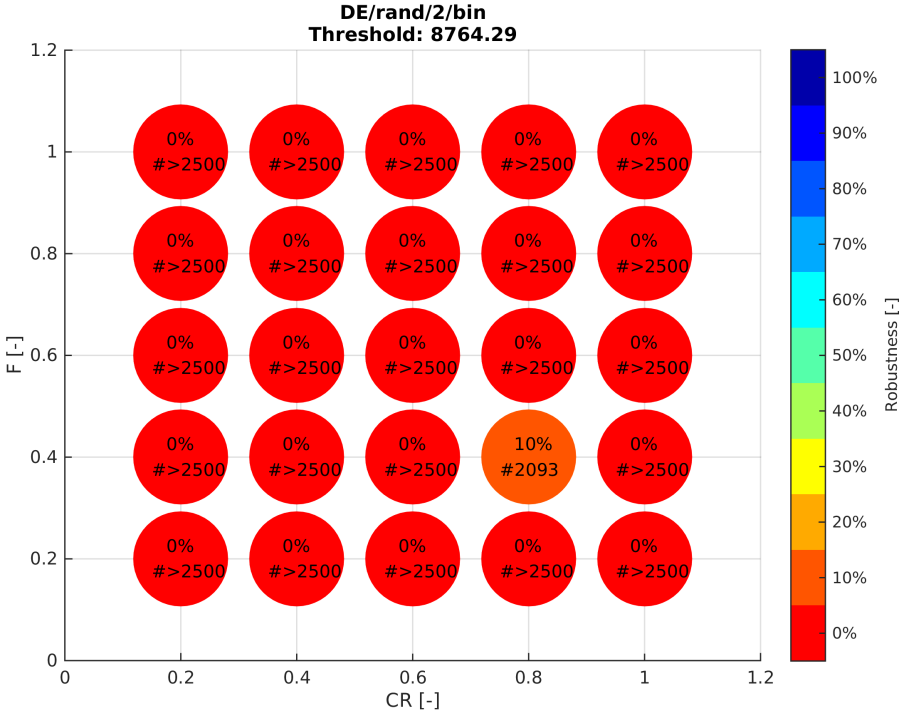

Figure C.9: *Optimisation tuning results for the rand/2/bin scheme. Threshold for* $\Delta V = 8764.29\,\mathrm{m/s}$.

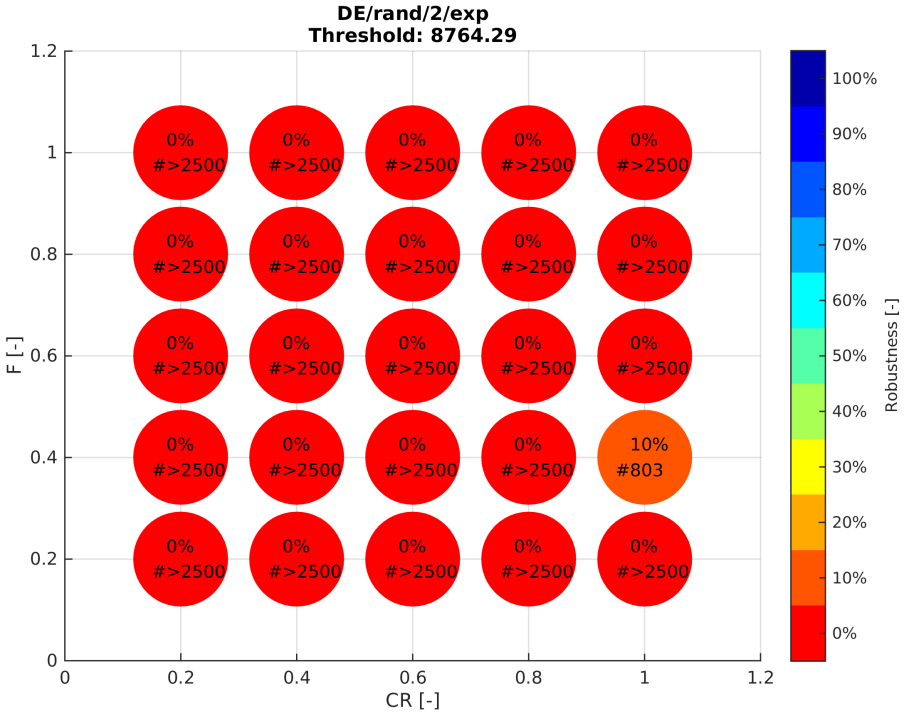

Figure C.10: *Optimisation tuning results for the rand/2/exp scheme. Threshold for* $\Delta V = 8764.29\,\mathrm{m/s}$.

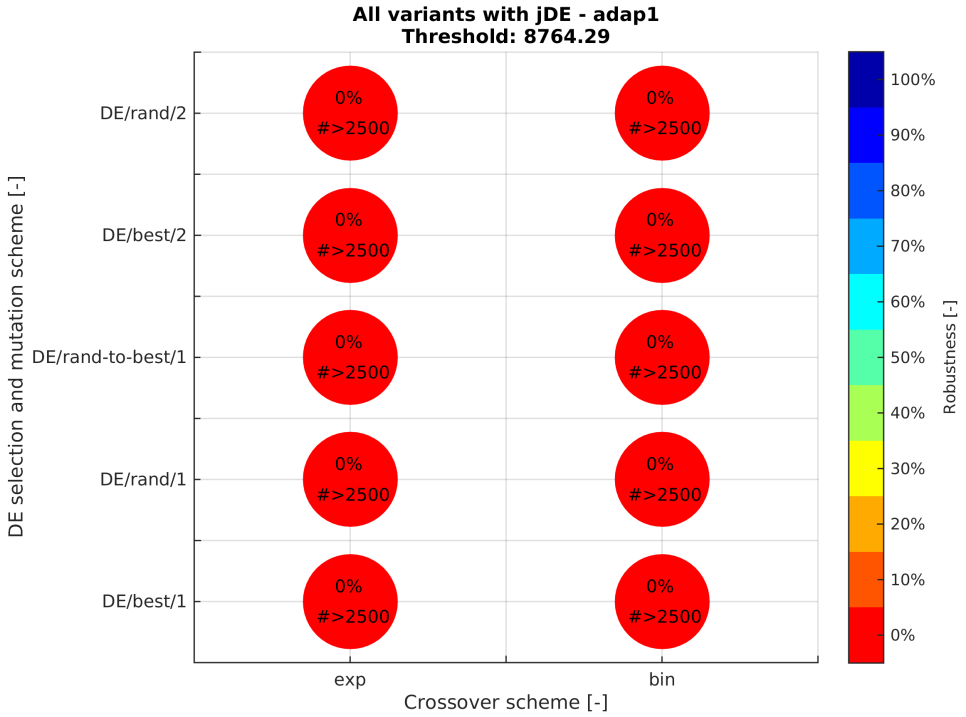

Figure C.11: *Optimisation tuning results for jDE (adaptive, scheme 1). Threshold for* $\Delta V = 8764.29\,\mathrm{m/s}$.

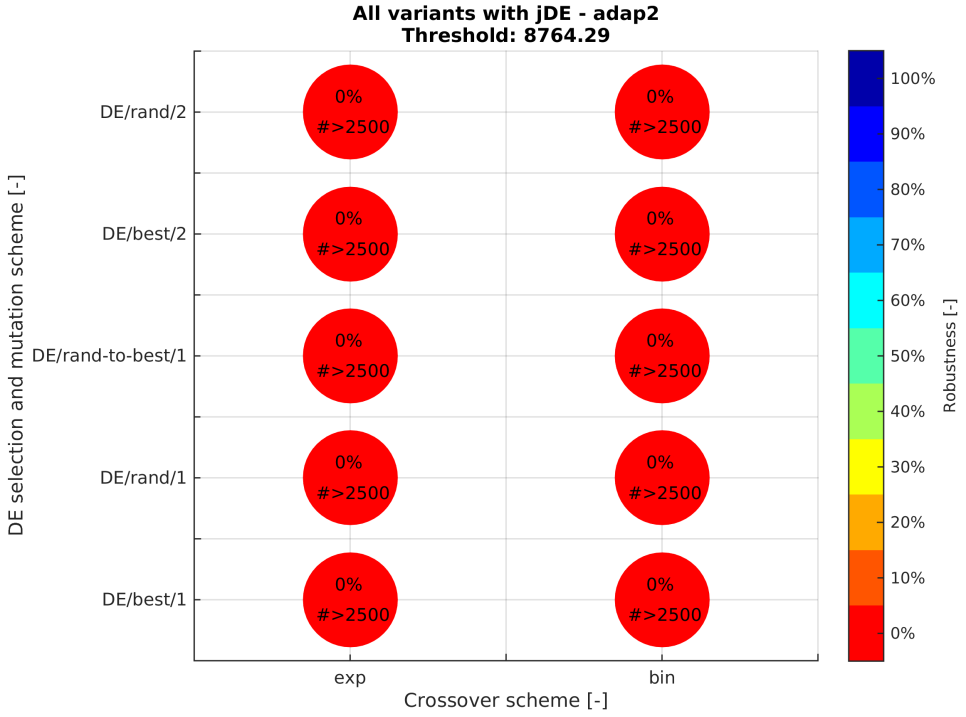

Figure C.12: *Optimisation tuning results for jDE (adaptive, scheme 2). Threshold for* $\Delta V = 8764.29\,\mathrm{m/s}$.

C.2 Ea-Ea-Apollo with threshold of 8764.34 + 25 m/s

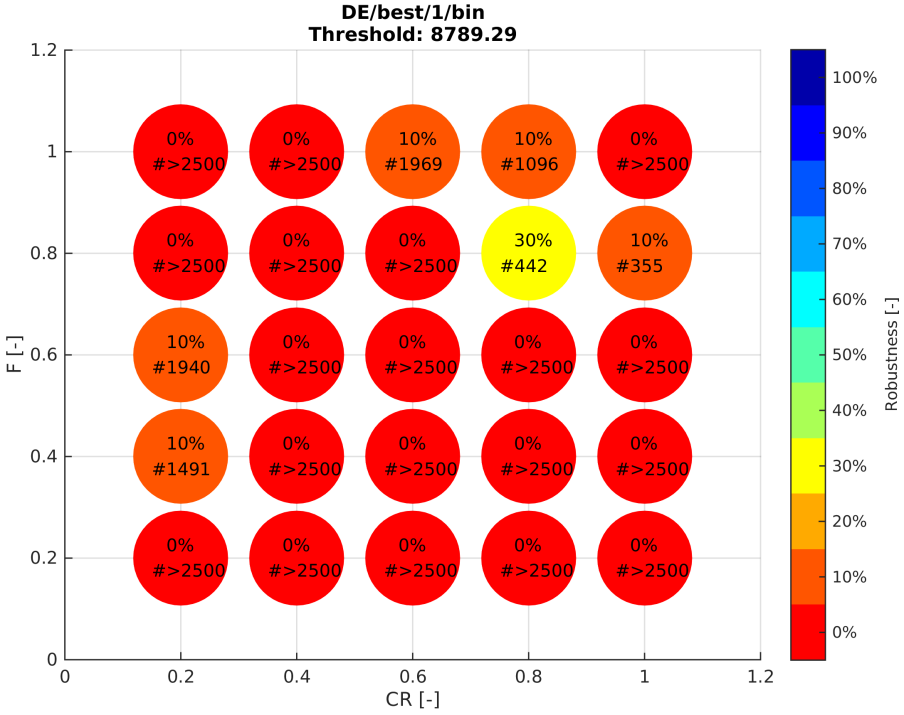

Figure C.13: *Optimisation tuning results for the best/1/bin scheme. Threshold for* $\Delta V = 8764.29 + 25.0\,\mathrm{m/s}$.

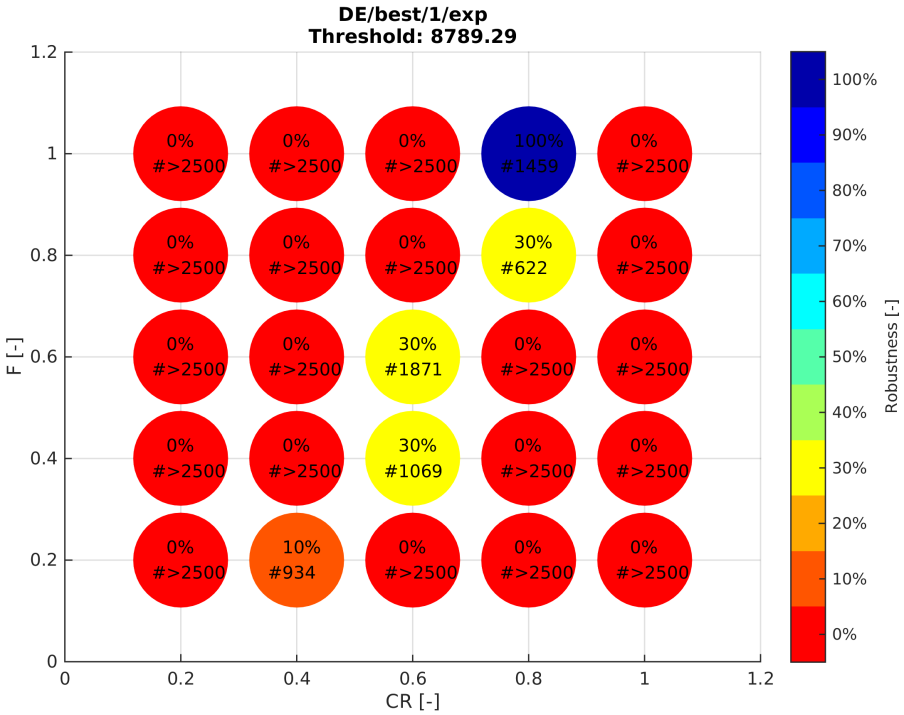

Figure C.14: *Optimisation tuning results for the best/1/exp scheme. Threshold for* $\Delta V = 8764.29 + 25.0\,\mathrm{m/s}$.

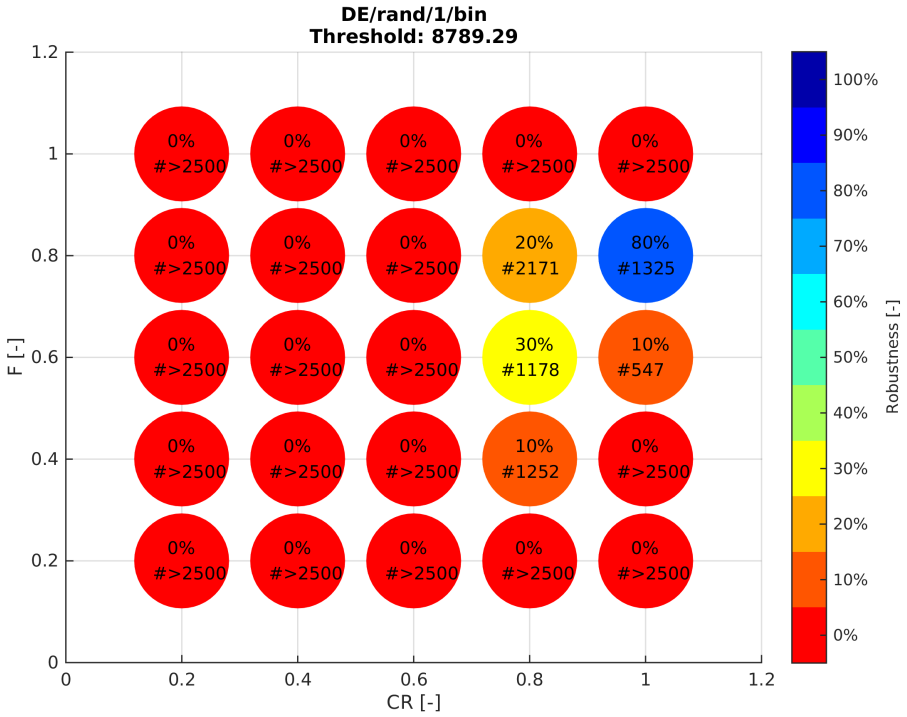

Figure C.15: *Optimisation tuning results for the rand/1/bin scheme. Threshold for* $\Delta V = 8764.29 + 25.0\,\mathrm{m/s}$.

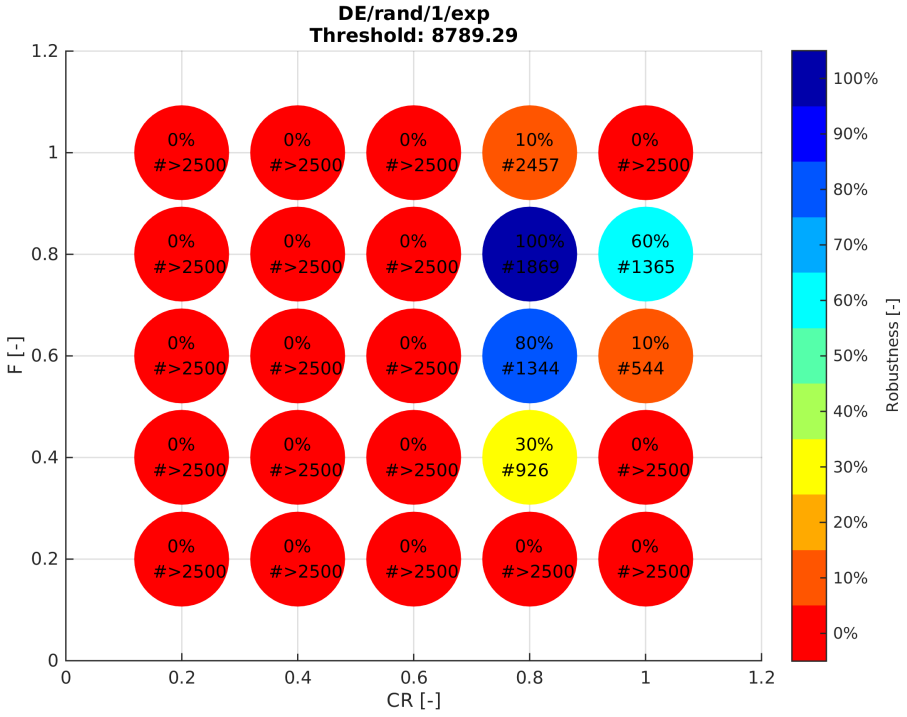

Figure C.16: *Optimisation tuning results for the rand/1/exp scheme. Threshold for* $\Delta V = 8764.29 + 25.0\,\mathrm{m/s}$.

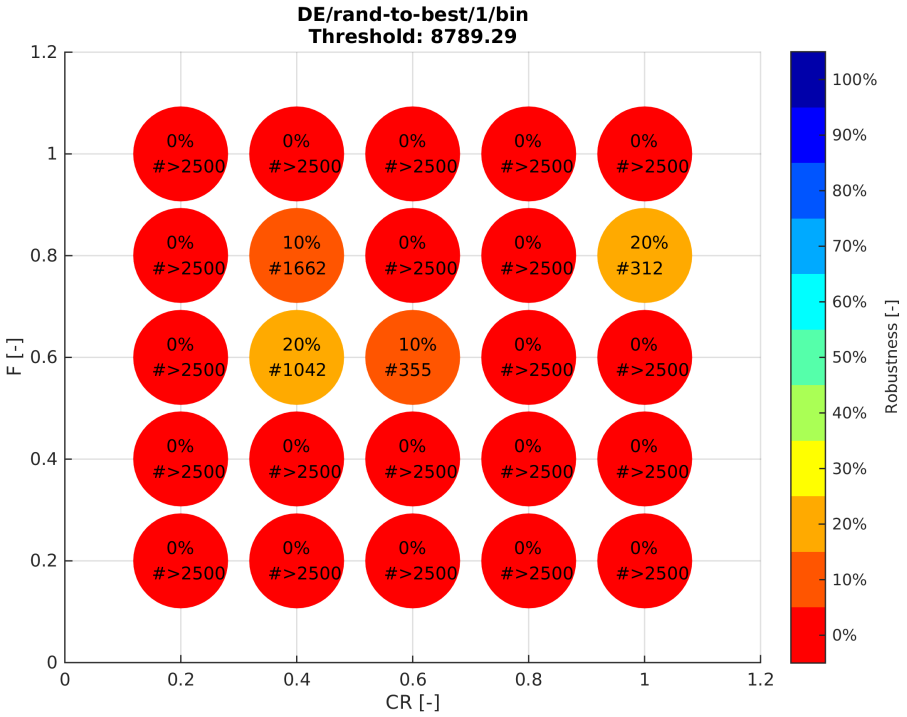

Figure C.17: *Optimisation tuning results for the rand-to-best/1/bin scheme. Threshold for* $\Delta V = 8764.29 + 25.0\,\mathrm{m/s}$.

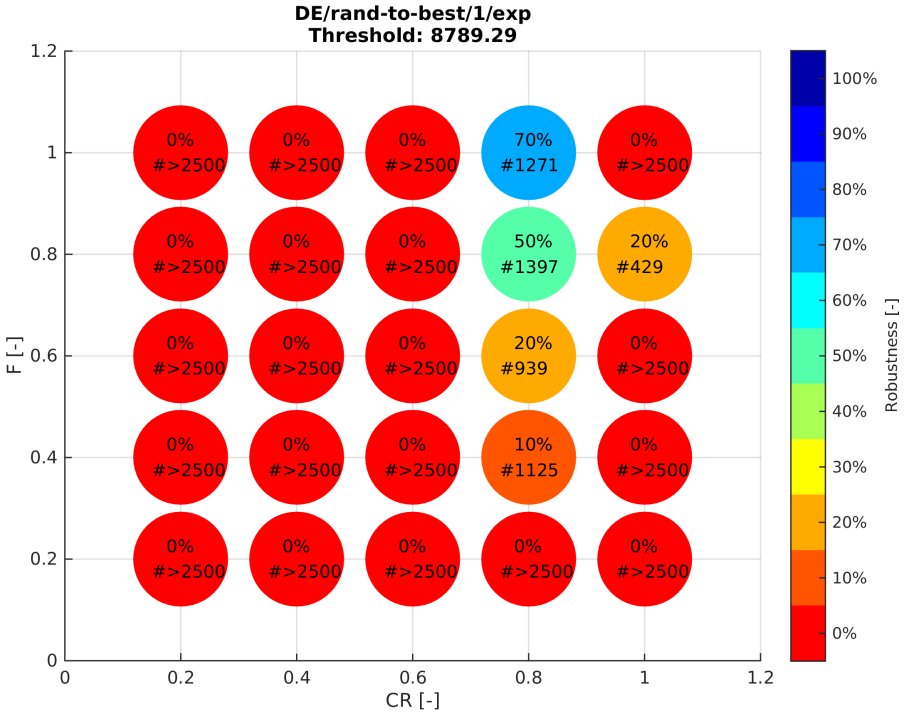

Figure C.18: *Optimisation tuning results for the rand-to-best/1/exp scheme. Threshold for* $\Delta V = 8764.29 + 25.0\,\mathrm{m/s}$.

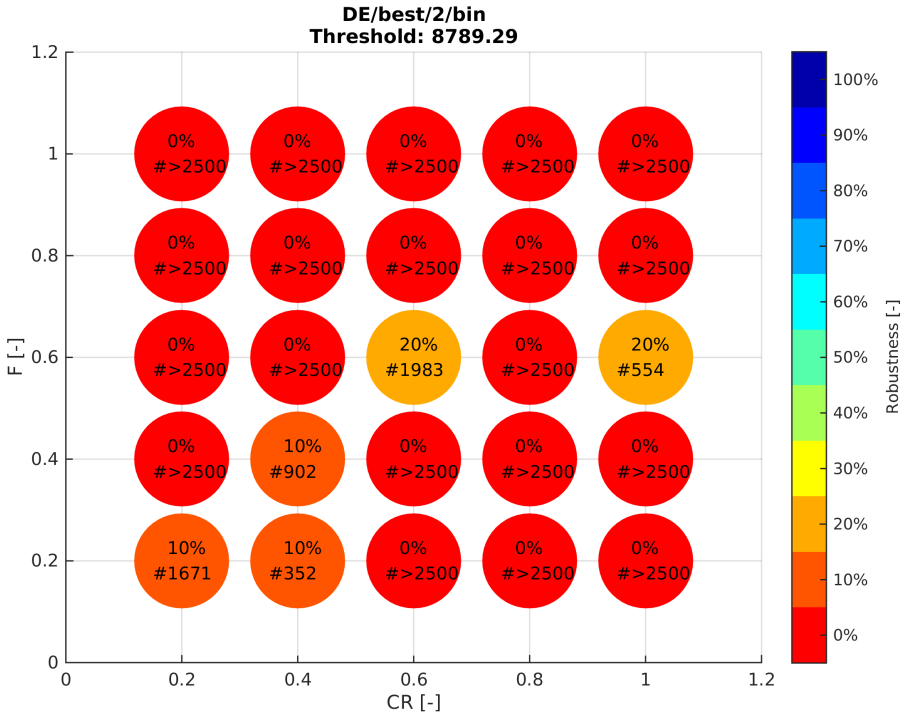

Figure C.19: *Optimisation tuning results for the best/2/bin scheme. Threshold for* $\Delta V = 8764.29 + 25.0\,\mathrm{m/s}$.

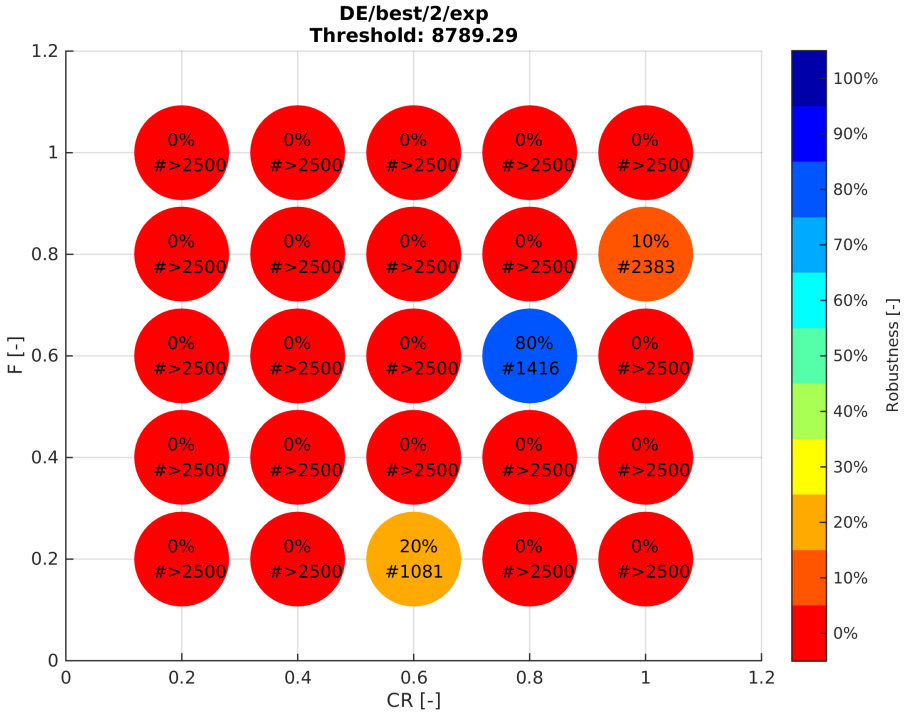

Figure C.20: *Optimisation tuning results for the best/2/exp scheme. Threshold for* $\Delta V = 8764.29 + 25.0\,\mathrm{m/s}$.

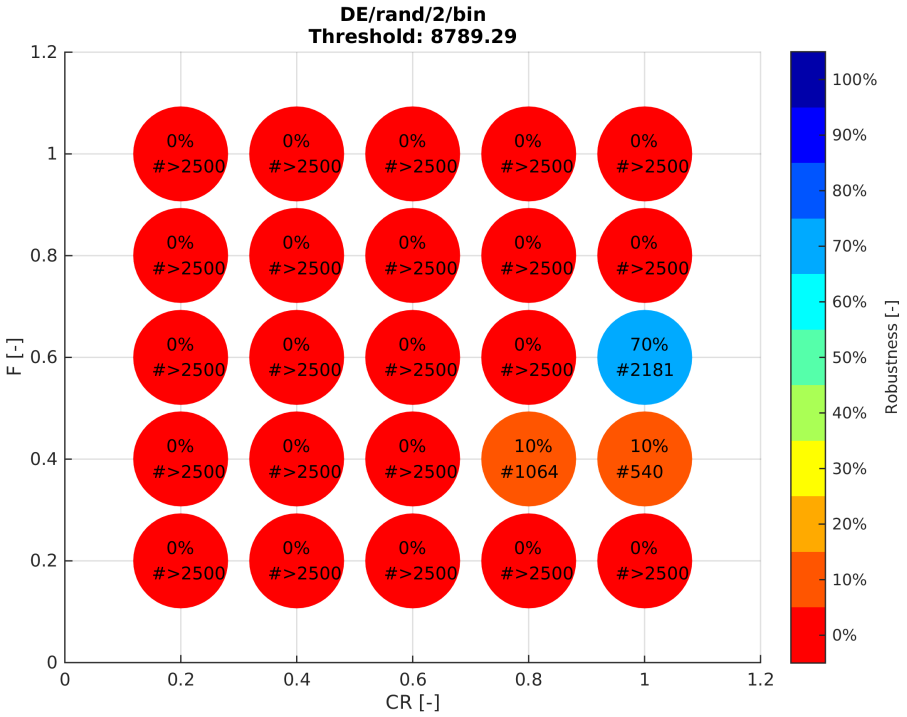

Figure C.21: *Optimisation tuning results for the rand/2/bin scheme. Threshold for* $\Delta V = 8764.29 + 25.0\,\mathrm{m/s}$.

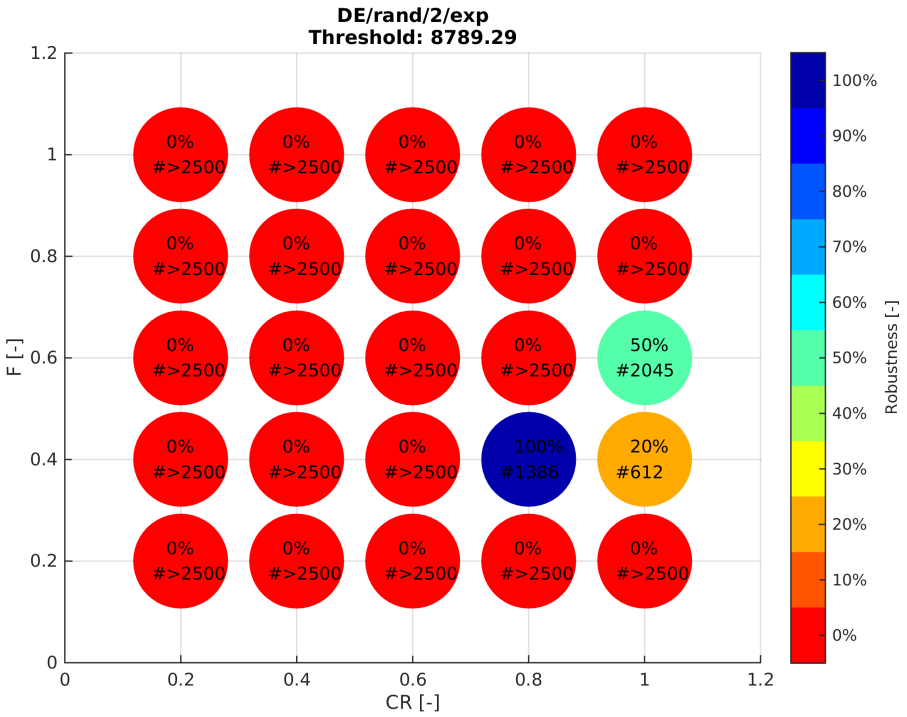

Figure C.22: *Optimisation tuning results for the rand/2/exp scheme. Threshold for* $\Delta V = 8764.29 + 25.0\,\mathrm{m/s}$.

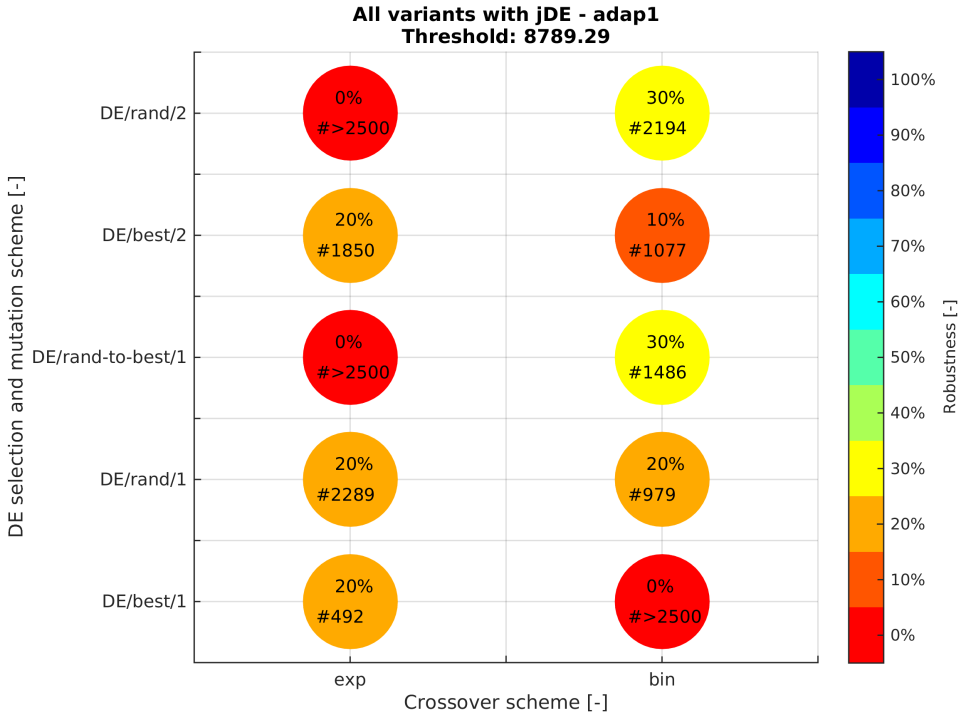

Figure C.23: *Optimisation tuning results for jDE (adaptive, scheme 1). Threshold for* $\Delta V = 8764.29 + 25.0\,\mathrm{m/s}$.

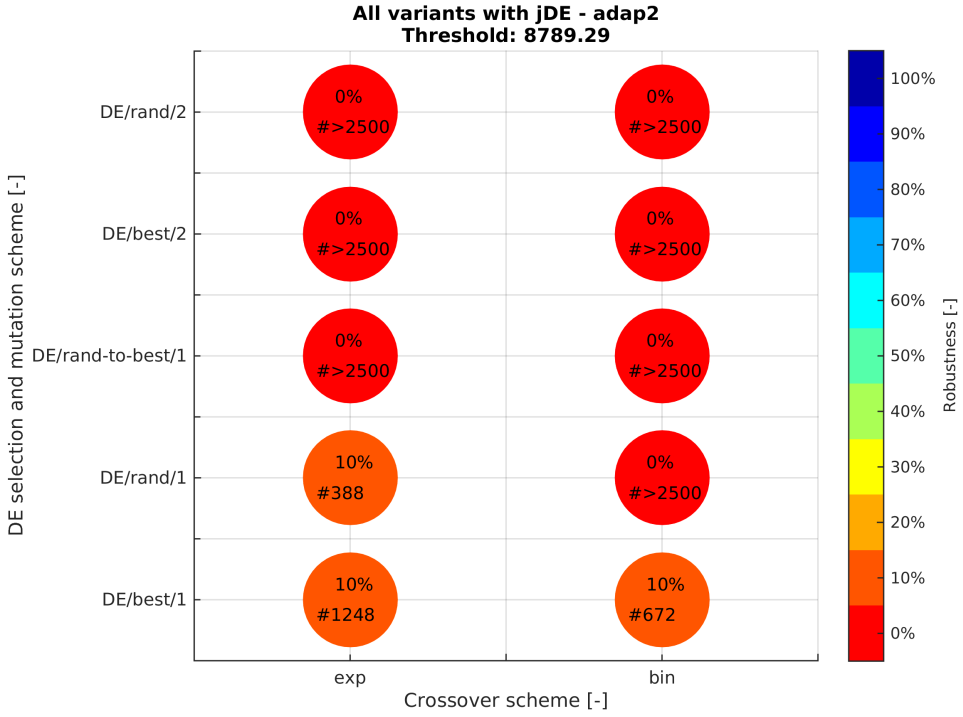

Figure C.24: *Optimisation tuning results for jDE (adaptive, scheme 2). Threshold for* $\Delta V = 8764.29 + 25.0\,\mathrm{m/s}$.

D

Sequence Optimisation Results

This appendix holds the results of the optimisation strategy test suite for GTOC6.

D.1 Optimisation of Sequence Eu-Eu-Eu-Io-Io for ΔV

D.1.1 Full Freedom and One Flyby Window

Table D.1: *Results (3x) of subsets for the gravity-assist manoeuvres for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: full freedom and 1 flyby window]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59884.083	413.2	1886.6	1889.67
2 (16)	Eu	59889.434	411.8	1521.5	1886.61
1 (15)	Eu	59883.234	304.9	899.4	1889.67
2 (16)	Eu	59888.557	300.0	50.0	1886.22
3 (17)	Eu	59895.662	300.0	102.8	1881.93
1 (15)	Eu	59882.187	1438.8	50.7	1889.67
2 (16)	Eu	59887.478	1449.8	50.0	1867.61
3 (17)	Eu	59894.459	1917.1	50.0	1828.46
4 (18)	Io	59898.223	1804.6	1874.1	1794.05
1 (15)	Eu	59882.187	971.7	50.0	1889.67
2 (16)	Eu	59887.465	1183.1	50.0	1858.40
3 (17)	Eu	59894.488	1633.0	50.0	1814.73
4 (18)	Io	59898.310	1902.3	50.0	1756.39
5 (19)	Io	59900.972	1888.2	1064.8	1743.07
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59898.322	498.9	1906.2	1889.67
2 (16)	Eu	59901.876	499.3	1619.1	1886.44
1 (15)	Eu	59899.902	302.6	1322.8	1889.67
2 (16)	Eu	59903.458	302.9	75.6	1887.97
3 (17)	Eu	59910.549	300.0	1218.7	1882.37
1 (15)	Eu	59899.851	1413.4	255.7	1889.67
2 (16)	Eu	59905.141	1426.6	50.0	1867.37
3 (17)	Eu	59912.123	1925.5	50.0	1827.03
4 (18)	Io	59915.877	1801.9	450.8	1792.17
1 (15)	Eu	59898.315	1763.0	76.0	1889.67
2 (16)	Eu	59905.373	1805.7	205.6	1865.10
3 (17)	Eu	59912.295	1856.1	50.0	1834.88
4 (18)	Io	59916.098	2049.1	50.0	1780.34
5 (19)	Io	59919.640	2080.4	441.9	1771.36
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59864.881	305.1	783.4	1889.67
2 (16)	Eu	59870.191	304.5	1168.3	1886.60
1 (15)	Eu	59863.338	300.0	363.3	1889.67
2 (16)	Eu	59866.894	301.6	526.2	1887.93
3 (17)	Eu	59873.998	301.3	699.8	1884.58
1 (15)	Eu	59862.750	1936.8	124.5	1889.67
2 (16)	Eu	59866.310	2028.7	50.6	1879.02
3 (17)	Eu	59873.266	2231.4	50.0	1850.60
4 (18)	Io	59876.987	1866.1	1627.1	1825.90
1 (15)	Eu	59861.015	1077.5	61.1	1889.67
2 (16)	Eu	59866.285	1083.9	50.0	1867.72
3 (17)	Eu	59873.293	1620.1	50.0	1824.63
4 (18)	Io	59877.086	1761.4	50.0	1758.53
5 (19)	Io	59879.747	1738.9	511.1	1743.92

[1] Number between parenthesis is the actual flyby number in the CSU solution.

Table D.2: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: full freedom and 1 flyby window]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59884.083	5.35	1	3.0	3.0
				ΔV_{tot} =	3.0	3.0
1 (15-16)	Eu-Eu	59883.234	5.32	1	6.0	6.0
2 (16-17)	Eu-Eu	59888.557	7.11	1	2.0	2.0
				ΔV_{tot} =	8.0	8.0
1 (15-16)	Eu-Eu	59882.187	5.29	1	203.0	203.0
2 (16-17)	Eu-Eu	59887.478	6.98	1	72709.0	392.0
3 (17-18)	Eu-Io	59894.459	3.76	1	91439.0	360.0
				ΔV_{tot} =	164351.0	955.0
1 (15-16)	Eu-Eu	59882.187	7.05	1	289.0	289.0
2 (16-17)	Eu-Eu	59887.465	7.02	1	87338.0	441.0
3 (17-18)	Eu-Io	59894.488	3.82	1	240237.0	628.0
4 (18-19)	Io-Io	59898.310	2.66	1	109.0	109.0
				ΔV_{tot} =	327973.0	1467.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59898.322	3.55	1	4.0	4.0
				ΔV_{tot} =	4.0	4.0
1 (15-16)	Eu-Eu	59899.902	3.55	1	3.0	3.0
2 (16-17)	Eu-Eu	59903.458	7.09	1	13.0	13.0
				ΔV_{tot} =	16.0	16.0
1 (15-16)	Eu-Eu	59899.851	5.29	1	206.0	206.0
2 (16-17)	Eu-Eu	59905.141	6.98	1	76729.0	405.0
3 (17-18)	Eu-Io	59912.123	3.75	1	93684.0	365.0
				ΔV_{tot} =	170619.0	976.0
1 (15-16)	Eu-Eu	59898.315	7.01	1	233.0	233.0
2 (16-17)	Eu-Eu	59905.373	6.92	1	297.0	297.0
3 (17-18)	Eu-Io	59912.295	3.80	1	201108.0	566.0
4 (18-19)	Io-Io	59916.098	3.54	1	55.0	55.0
				ΔV_{tot} =	201693.0	1151.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59864.881	5.31	1	2.0	2.0
				ΔV_{tot} =	2.0	2.0
1 (15-16)	Eu-Eu	59863.338	3.56	1	3.0	3.0
2 (16-17)	Eu-Eu	59866.894	7.10	1	5.0	5.0
				ΔV_{tot} =	8.0	8.0
1 (15-16)	Eu-Eu	59862.750	3.56	1	99.0	99.0
2 (16-17)	Eu-Eu	59866.310	6.96	1	34836.0	276.0
3 (17-18)	Eu-Io	59873.266	3.72	1	46590.0	251.0
				ΔV_{tot} =	81525.0	626.0
1 (15-16)	Eu-Eu	59861.015	5.27	1	202.0	202.0
2 (16-17)	Eu-Eu	59866.285	7.01	1	76104.0	434.0
3 (17-18)	Eu-Io	59873.293	3.79	1	294095.0	711.0
4 (18-19)	Io-Io	59877.086	2.66	1	23611.0	124.0
				ΔV_{tot} =	394012.0	1471.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

D.1.2 Full Freedom and Two Flyby Window

Table D.3: *Results (3x) of subsets for the gravity-assist manoeuvres for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: full freedom and 2 flyby window]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59871.760	351.9	263.5	1889.67
2 (16)	Eu	59877.070	352.1	1889.3	1886.34
1 (15)	Eu	59870.083	300.1	53.3	1889.67
2 (16)	Eu	59875.439	301.6	50.5	1885.39
3 (17)	Eu	59882.543	300.0	494.6	1882.27
1 (15)	Eu	59869.674	2630.9	1079.8	1889.67
2 (16)	Eu	59876.766	2741.3	1601.6	1871.54
3 (17)	Eu	59883.790	2896.9	50.0	1847.57
4 (18)	Io	59887.481	2031.6	1064.9	1831.00
1 (15)	Eu	59869.717	1549.6	836.9	1889.67
2 (16)	Eu	59876.807	1736.3	262.5	1857.29
3 (17)	Eu	59883.868	1759.9	50.0	1830.43
4 (18)	Io	59887.703	2147.2	50.0	1772.17
5 (19)	Io	59890.364	2129.7	1208.0	1763.22
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59901.742	351.9	274.5	1889.67
2 (16)	Eu	59908.847	351.5	834.2	1885.21
1 (15)	Eu	59901.021	308.9	685.1	1889.67
2 (16)	Eu	59908.126	308.8	138.4	1885.27
3 (17)	Eu	59915.232	308.6	1239.9	1880.95
1 (15)	Eu	59900.264	2343.6	160.3	1889.67
2 (16)	Eu	59907.366	2363.5	204.5	1884.72
3 (17)	Eu	59912.164	2363.7	50.0	1880.16
4 (18)	Io	59915.848	1945.1	991.8	1876.82
1 (15)	Eu	59901.981	1346.2	203.2	1889.67
2 (16)	Eu	59907.312	1350.7	50.0	1869.99
3 (17)	Eu	59912.177	1642.3	50.0	1838.35
4 (18)	Io	59915.945	1757.3	50.0	1776.58
5 (19)	Io	59918.605	1725.8	818.6	1760.43
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59878.027	358.0	1633.7	1889.67
2 (16)	Eu	59885.130	360.0	1745.1	1885.01
1 (15)	Eu	59877.036	308.3	1576.5	1889.67
2 (16)	Eu	59882.386	307.1	235.9	1886.66
3 (17)	Eu	59889.490	303.9	97.1	1881.74
1 (15)	Eu	59877.164	1400.4	210.2	1889.67
2 (16)	Eu	59882.495	1404.1	50.0	1869.98
3 (17)	Eu	59887.374	1752.4	50.0	1836.94
4 (18)	Io	59891.170	1791.3	1164.8	1811.04
1 (15)	Eu	59876.348	1425.0	126.4	1889.67
2 (16)	Eu	59881.667	1398.9	50.0	1867.84
3 (17)	Eu	59886.870	1676.4	50.0	1832.61
4 (18)	Io	59891.025	1915.4	50.0	1773.39
5 (19)	Io	59893.577	1946.2	765.7	1762.59

[1] Number between parenthesis is the actual flyby number in the CSU solution.

Table D.4: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: full freedom and 2 flyby window]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59871.760	5.31	1	5.0	5.0
				$\Delta V_{tot} =$	5.0	5.0
1 (15-16)	Eu-Eu	59870.083	5.36	1	15.0	15.0
2 (16-17)	Eu-Eu	59875.439	7.10	1	5.0	5.0
				$\Delta V_{tot} =$	20.0	20.0
1 (15-16)	Eu-Eu	59869.674	7.09	1	167.0	167.0
2 (16-17)	Eu-Eu	59876.766	7.02	1	231.0	231.0
3 (17-18)	Eu-Io	59883.790	3.69	1	21054.0	165.0
				$\Delta V_{tot} =$	21452.0	563.0
1 (15-16)	Eu-Eu	59869.717	7.09	1	299.0	299.0
2 (16-17)	Eu-Eu	59876.807	7.06	1	260.0	260.0
3 (17-18)	Eu-Io	59883.868	3.84	1	239553.0	622.0
4 (18-19)	Io-Io	59887.703	2.66	1	60.0	60.0
				$\Delta V_{tot} =$	240172.0	1241.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59901.742	7.11	1	1.0	1.0
				$\Delta V_{tot} =$	1.0	1.0
1 (15-16)	Eu-Eu	59901.021	7.11	1	2.0	2.0
2 (16-17)	Eu-Eu	59908.126	7.11	1	1.0	1.0
				$\Delta V_{tot} =$	3.0	3.0
1 (15-16)	Eu-Eu	59900.264	7.10	1	18.0	18.0
2 (16-17)	Eu-Eu	59907.366	4.80	1	25.0	25.0
3 (17-18)	Eu-Io	59912.164	3.68	1	23.0	23.0
				$\Delta V_{tot} =$	66.0	66.0
1 (15-16)	Eu-Eu	59901.981	5.33	1	179.0	179.0
2 (16-17)	Eu-Eu	59907.312	4.87	1	44307.0	311.0
3 (17-18)	Eu-Io	59912.177	3.77	1	256198.0	658.0
4 (18-19)	Io-Io	59915.945	2.66	1	31154.0	141.0
				$\Delta V_{tot} =$	331838.0	1289.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59878.027	7.10	1	4.0	4.0
				$\Delta V_{tot} =$	4.0	4.0
1 (15-16)	Eu-Eu	59877.036	5.35	1	2.0	2.0
2 (16-17)	Eu-Eu	59882.386	7.10	1	5.0	5.0
				$\Delta V_{tot} =$	7.0	7.0
1 (15-16)	Eu-Eu	59877.164	5.33	1	179.0	179.0
2 (16-17)	Eu-Eu	59882.495	4.88	1	60006.0	326.0
3 (17-18)	Eu-Io	59887.374	3.80	1	50400.0	266.0
				$\Delta V_{tot} =$	110585.0	771.0
1 (15-16)	Eu-Eu	59876.348	5.32	1	202.0	202.0
2 (16-17)	Eu-Eu	59881.667	5.20	1	79835.0	351.0
3 (17-18)	Eu-Io	59886.870	4.15	1	237114.0	619.0
4 (18-19)	Io-Io	59891.025	2.55	1	90.0	90.0
				$\Delta V_{tot} =$	317241.0	1262.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

D.1.3 Full Freedom and Three Flyby Window

Table D.5: *Results (3x) of subsets for the gravity-assist manoeuvres for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: full freedom and 3 flyby window]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59890.030	383.4	558.0	1889.67
2 (16)	Eu	59895.344	379.1	929.5	1886.62
1 (15)	Eu	59898.187	300.1	1266.2	1889.67
2 (16)	Eu	59905.293	300.3	496.5	1885.31
3 (17)	Eu	59908.853	300.0	1061.4	1882.12
1 (15)	Eu	59898.369	1397.1	193.8	1889.67
2 (16)	Eu	59903.703	1393.9	50.0	1869.95
3 (17)	Eu	59908.580	1756.4	50.0	1836.56
4 (18)	Io	59912.357	1790.3	1017.6	1809.62
1 (15)	Eu	59897.900	1810.5	1200.3	1889.67
2 (16)	Eu	59903.230	1766.1	2000.0	1867.81
3 (17)	Eu	59908.538	1810.1	50.0	1842.69
4 (18)	Io	59912.415	2292.1	50.0	1791.90
5 (19)	Io	59915.075	2270.5	577.2	1780.28
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59882.662	370.0	111.2	1889.67
2 (16)	Eu	59889.768	368.5	1176.3	1886.59
1 (15)	Eu	59888.901	300.0	1549.0	1889.67
2 (16)	Eu	59896.003	300.0	197.5	1886.34
3 (17)	Eu	59903.101	302.7	425.9	1882.95
1 (15)	Eu	59887.458	2090.0	780.8	1889.67
2 (16)	Eu	59894.555	2206.0	241.2	1871.56
3 (17)	Eu	59901.525	2283.6	50.0	1842.27
4 (18)	Io	59905.247	1886.5	1538.4	1816.48
1 (15)	Eu	59885.681	1174.5	265.7	1889.67
2 (16)	Eu	59892.650	1349.5	1847.1	1865.36
3 (17)	Eu	59897.698	1501.7	50.0	1815.13
4 (18)	Io	59901.768	1736.4	50.0	1742.38
5 (19)	Io	59904.428	1706.6	936.9	1728.89
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59869.426	305.0	310.1	1889.67
2 (16)	Eu	59874.768	300.0	146.4	1886.71
1 (15)	Eu	59874.445	304.1	323.3	1889.67
2 (16)	Eu	59877.992	301.7	51.8	1886.35
3 (17)	Eu	59883.324	306.2	1945.2	1882.10
1 (15)	Eu	59873.266	2492.4	1829.1	1889.67
2 (16)	Eu	59880.349	2593.9	792.2	1873.28
3 (17)	Eu	59887.345	2670.0	50.0	1845.87
4 (18)	Io	59891.041	1983.5	1102.4	1827.89
1 (15)	Eu	59873.019	1106.0	1494.0	1889.67
2 (16)	Eu	59878.337	1093.9	51.7	1875.33
3 (17)	Eu	59883.419	1508.5	50.0	1838.65
4 (18)	Io	59887.517	1876.2	53.2	1776.16
5 (19)	Io	59890.059	1891.1	1029.7	1769.24

[1] Number between parenthesis is the actual flyby number in the CSU solution.

Table D.6: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: full freedom and 3 flyby window]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59895.344	5.31	1	2.0	2.0
				ΔV_{tot} =	2.0	2.0
1 (15-16)	Eu-Eu	59898.187	7.11	1	2.0	2.0
2 (16-17)	Eu-Eu	59905.293	3.56	1	5.0	5.0
				ΔV_{tot} =	7.0	7.0
1 (15-16)	Eu-Eu	59898.369	5.33	1	180.0	180.0
2 (16-17)	Eu-Eu	59903.703	4.88	1	62562.0	330.0
3 (17-18)	Eu-Io	59908.580	3.78	1	54926.0	277.0
				ΔV_{tot} =	117668.0	787.0
1 (15-16)	Eu-Eu	59897.900	5.33	1	202.0	202.0
2 (16-17)	Eu-Eu	59903.230	5.31	1	45738.0	237.0
3 (17-18)	Eu-Io	59908.538	3.88	1	178245.0	535.0
4 (18-19)	Io-Io	59912.415	2.66	1	88.0	88.0
				ΔV_{tot} =	224273.0	1062.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59882.662	7.11	1	3.0	3.0
				ΔV_{tot} =	3.0	3.0
1 (15-16)	Eu-Eu	59888.901	7.10	1	6.0	6.0
2 (16-17)	Eu-Eu	59896.003	7.10	1	7.0	7.0
				ΔV_{tot} =	13.0	13.0
1 (15-16)	Eu-Eu	59887.458	7.10	1	166.0	166.0
2 (16-17)	Eu-Eu	59894.555	6.97	1	287.0	287.0
3 (17-18)	Eu-Io	59901.525	3.72	1	52890.0	264.0
				ΔV_{tot} =	53343.0	717.0
1 (15-16)	Eu-Eu	59885.681	6.97	1	231.0	231.0
2 (16-17)	Eu-Eu	59892.650	5.05	1	126904.0	512.0
3 (17-18)	Eu-Io	59897.698	4.07	1	330858.0	776.0
4 (18-19)	Io-Io	59901.768	2.66	1	21796.0	114.0
				ΔV_{tot} =	479789.0	1633.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59869.426	5.34	1	1.0	1.0
				ΔV_{tot} =	1.0	1.0
1 (15-16)	Eu-Eu	59874.445	3.54	1	6.0	6.0
2 (16-17)	Eu-Eu	59877.992	5.33	1	15.0	15.0
				ΔV_{tot} =	21.0	21.0
1 (15-16)	Eu-Eu	59873.266	7.08	1	149.0	149.0
2 (16-17)	Eu-Eu	59880.349	7.00	1	267.0	267.0
3 (17-18)	Eu-Io	59887.345	3.70	1	25257.0	180.0
				ΔV_{tot} =	25673.0	596.0
1 (15-16)	Eu-Eu	59873.019	5.32	1	121.0	121.0
2 (16-17)	Eu-Eu	59878.337	5.08	1	102054.0	365.0
3 (17-18)	Eu-Io	59883.419	4.10	1	274208.0	653.0
4 (18-19)	Io-Io	59887.517	2.54	1	47.0	47.0
				ΔV_{tot} =	376430.0	1184.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

D.1.4 Small Freedom and One Flyby Window

Table D.7: *Results (3x) of subsets for the gravity-assist manoeuvres for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: small freedom and 1 flyby window]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59895.887	342.0	106.7	1889.67
2 (16)	Eu	59902.993	341.4	830.3	1886.69
1 (15)	Eu	59895.865	300.6	1811.6	1889.67
2 (16)	Eu	59902.965	301.0	134.8	1885.18
3 (17)	Eu	59906.522	301.4	1016.1	1883.40
1 (15)	Eu	59895.487	1822.4	552.0	1889.67
2 (16)	Eu	59902.580	1913.9	83.5	1865.53
3 (17)	Eu	59907.834	1978.8	50.0	1835.78
4 (18)	Io	59912.195	1852.2	1911.2	1786.51
1 (15)	Eu	59895.721	1870.7	1358.8	1889.67
2 (16)	Eu	59902.785	1795.8	1257.3	1869.96
3 (17)	Eu	59908.040	1816.8	50.0	1836.33
4 (18)	Io	59912.199	2017.5	50.0	1780.86
5 (19)	Io	59914.757	2023.4	1142.4	1773.09
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59867.011	330.7	1996.8	1889.67
2 (16)	Eu	59872.353	336.2	380.5	1886.73
1 (15)	Eu	59867.026	302.7	1478.1	1889.67
2 (16)	Eu	59874.130	304.4	650.6	1886.42
3 (17)	Eu	59881.235	305.7	1293.3	1883.40
1 (15)	Eu	59866.221	1998.3	532.4	1889.67
2 (16)	Eu	59873.291	2161.5	61.0	1868.79
3 (17)	Eu	59880.291	2325.9	50.0	1845.96
4 (18)	Io	59884.015	1917.6	1082.6	1819.91
1 (15)	Eu	59866.432	1010.9	757.3	1889.67
2 (16)	Eu	59873.526	1261.4	76709	1861.29
3 (17)	Eu	59880.358	1562.8	50.0	1819.40
4 (18)	Io	59884.174	1794.5	50.0	1756.29
5 (19)	Io	59886.835	1778.5	1014.8	1741.88
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59868.130	325.6	1797.6	1889.67
2 (16)	Eu	59875.235	324.1	630.7	1886.47
1 (15)	Eu	59869.596	301.5	310.9	1889.67
2 (16)	Eu	59876.701	301.3	51.8	1885.33
3 (17)	Eu	59880.254	300.0	1684.7	1882.29
1 (15)	Eu	59868.328	921.9	51.5	1889.67
2 (16)	Eu	59875.409	1231.7	1703.1	1856.28
3 (17)	Eu	59880.288	1670.4	50.0	1820.82
4 (18)	Io	59884.110	1773.4	1638.8	1790.28
1 (15)	Eu	59867.905	1959.9	1308.7	1889.67
2 (16)	Eu	59874.995	1878.4	1627.2	1857.67
3 (17)	Eu	59880.305	1917.8	50.0	1831.35
4 (18)	Io	59884.150	2259.9	50.0	1781.92
5 (19)	Io	59886.811	2243.6	1017.6	1770.73

[1] Number between parenthesis is the actual flyby number in the CSU solution.

Table D.8: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: small freedom and 1 flyby window]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59895.887	7.10	1	1.0	1.0
				ΔV_{tot} =	1.0	1.0
1 (15-16)	Eu-Eu	59895.865	7.10	1	4.0	4.0
2 (16-17)	Eu-Eu	59902.965	3.56	1	4.0	4.0
				ΔV_{tot} =	8.0	8.0
1 (15-16)	Eu-Eu	59895.487	7.09	1	229.0	229.0
2 (16-17)	Eu-Eu	59902.580	5.26	1	60518.0	293.0
3 (17-18)	Eu-Io	59907.834	4.36	1	159608.0	509.0
				ΔV_{tot} =	220355.0	1031.0
1 (15-16)	Eu-Eu	59895.721	7.06	1	183.0	183.0
2 (16-17)	Eu-Eu	59902.785	5.26	1	79897.0	334.0
3 (17-18)	Eu-Io	59908.040	4.16	1	207140.0	576.0
4 (18-19)	Io-Io	59912.199	2.56	1	56.0	56.0
				ΔV_{tot} =	287276.0	1149.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59867.011	5.34	1	1.0	1.0
				ΔV_{tot} =	1.0	1.0
1 (15-16)	Eu-Eu	59867.026	7.10	1	4.0	4.0
2 (16-17)	Eu-Eu	59874.130	7.10	1	2.0	2.0
				ΔV_{tot} =	6.0	6.0
1 (15-16)	Eu-Eu	59866.221	7.07	1	195.0	195.0
2 (16-17)	Eu-Eu	59873.291	7.00	1	219.0	219.0
3 (17-18)	Eu-Io	59880.291	3.73	1	53123.0	267.0
				ΔV_{tot} =	53537.0	681.0
1 (15-16)	Eu-Eu	59866.432	7.09	1	270.0	270.0
2 (16-17)	Eu-Eu	59873.526	6.83	1	75755.0	423.0
3 (17-18)	Eu-Io	59880.358	3.82	1	278564.0	679.0
4 (18-19)	Io-Io	59884.174	2.66	1	122.0	122.0
				ΔV_{tot} =	354711.0	1494.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59868.130	7.10	1	4.0	4.0
				ΔV_{tot} =	4.0	4.0
1 (15-16)	Eu-Eu	59869.596	7.10	1	2.0	2.0
2 (16-17)	Eu-Eu	59876.701	3.55	1	3.0	3.0
				ΔV_{tot} =	5.0	5.0
1 (15-16)	Eu-Eu	59868.328	7.08	1	311.0	311.0
2 (16-17)	Eu-Eu	59875.409	4.88	1	73539.0	354.0
3 (17-18)	Eu-Io	59880.288	3.82	1	71623.0	319.0
				ΔV_{tot} =	145473.0	984.0
1 (15-16)	Eu-Eu	59867.905	7.09	1	299.0	299.0
2 (16-17)	Eu-Eu	59874.995	5.31	1	48340.0	251.0
3 (17-18)	Eu-Io	59880.305	3.85	1	168191.0	524.0
4 (18-19)	Io-Io	59884.150	2.66	1	84.0	84.0
				ΔV_{tot} =	216914.0	1158.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

D.1.5 Small Freedom and Three Flyby Window

Table D.9: *Results (3x) of subsets for the gravity-assist manoeuvres for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: small freedom and 3 flyby window]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59878.226	330.9	217.4	1889.67
2 (16)	Eu	59885.332	330.3	359.3	1886.70
1 (15)	Eu	59878.725	300.1	1667.6	1889.67
2 (16)	Eu	59882.280	301.1	51.3	1888.09
3 (17)	Eu	59887.589	302.6	1597.3	1885.04
1 (15)	Eu	59878.659	1459.2	102.3	1889.67
2 (16)	Eu	59883.949	1469.0	50.0	1867.40
3 (17)	Eu	59890.931	1916.8	50.0	1829.18
4 (18)	Io	59894.708	1825.5	770.9	1794.41
1 (15)	Eu	59878.539	1685.5	716.6	1889.67
2 (16)	Eu	59883.834	1775.3	1849.8	1864.96
3 (17)	Eu	59890.899	1774.2	50.0	1837.01
4 (18)	Io	59894.755	2233.1	50.0	1780.15
5 (19)	Io	59897.416	2217.0	1134.9	1765.48
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59887.747	304.5	972.3	1889.67
2 (16)	Eu	59893.097	303.9	1791.9	1886.67
1 (15)	Eu	59890.411	397.6	924.5	1889.67
2 (16)	Eu	59897.512	394.7	1152.5	1886.63
3 (17)	Eu	59902.835	388.5	1297.8	1882.86
1 (15)	Eu	59889.269	1450.4	1664.9	1889.67
2 (16)	Eu	59894.558	1459.3	50.0	1867.45
3 (17)	Eu	59901.535	1898.8	50.0	1829.47
4 (18)	Io	59905.312	1821.2	1417.5	1793.72
1 (15)	Eu	59889.425	1136.1	288.3	1889.67
2 (16)	Eu	59894.697	1153.1	51.5	1868.64
3 (17)	Eu	59901.584	1558.4	52.1	1831.51
4 (18)	Io	59905.387	1737.5	50.0	1763.70
5 (19)	Io	59908.048	1725.3	1005.6	1749.76
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59858.994	327.9	239.7	1889.67
2 (16)	Eu	59864.342	324.0	738.7	1886.41
1 (15)	Eu	59859.504	300.0	1181.4	1889.67
2 (16)	Eu	59866.610	301.0	1433.4	1885.60
3 (17)	Eu	59871.958	302.4	1420.3	1882.11
1 (15)	Eu	59857.727	1327.2	140.1	1889.67
2 (16)	Eu	59862.990	1322.2	50.0	1868.16
3 (17)	Eu	59869.747	1795.2	50.0	1819.08
4 (18)	Io	59873.546	1805.5	501.1	1790.91
1 (15)	Eu	59857.544	1012.3	216.9	1889.67
2 (16)	Eu	59862.819	1039.8	93.9	1867.67
3 (17)	Eu	59869.777	1601.9	50.0	1826.46
4 (18)	Io	59873.454	1948.9	50.0	1770.70
5 (19)	Io	59876.961	1933.8	793.6	1757.19

[1] Number between parenthesis is the actual flyby number in the CSU solution.

Table D.10: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: small freedom and 3 flyby window]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59878.226	7.10	1	1.0	1.0
				$\Delta V_{tot} =$	1.0	1.0
1 (15-16)	Eu-Eu	59878.725	3.56	1	3.0	3.0
2 (16-17)	Eu-Eu	59882.280	5.31	1	2.0	2.0
				$\Delta V_{tot} =$	5.0	5.0
1 (15-16)	Eu-Eu	59878.659	5.29	1	206.0	206.0
2 (16-17)	Eu-Eu	59883.949	6.98	1	69338.0	382.0
3 (17-18)	Eu-Io	59890.931	3.78	1	92978.0	364.0
				$\Delta V_{tot} =$	162522.0	952.0
1 (15-16)	Eu-Eu	59878.539	5.30	1	231.0	231.0
2 (16-17)	Eu-Eu	59883.834	7.07	1	270.0	270.0
3 (17-18)	Eu-Io	59890.899	3.86	1	234259.0	604.0
4 (18-19)	Io-Io	59894.755	2.66	1	123.0	123.0
				$\Delta V_{tot} =$	234883.0	1228.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59887.747	5.35	1	2.0	2.0
				$\Delta V_{tot} =$	2.0	2.0
1 (15-16)	Eu-Eu	59890.411	7.10	1	5.0	5.0
2 (16-17)	Eu-Eu	59897.512	5.32	1	10.0	10.0
				$\Delta V_{tot} =$	15.0	15.0
1 (15-16)	Eu-Eu	59889.269	5.29	1	205.0	205.0
2 (16-17)	Eu-Eu	59894.558	6.98	1	69151.0	380.0
3 (17-18)	Eu-Io	59901.535	3.78	1	98059.0	375.0
				$\Delta V_{tot} =$	167415.0	960.0
1 (15-16)	Eu-Eu	59889.425	5.27	1	192.0	192.0
2 (16-17)	Eu-Eu	59894.697	6.89	1	65258.0	370.0
3 (17-18)	Eu-Io	59901.584	3.80	1	317501.0	727.0
4 (18-19)	Io-Io	59905.387	2.66	1	116.0	116.0
				$\Delta V_{tot} =$	383067.0	1405.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59858.994	5.35	1	4.0	4.0
				$\Delta V_{tot} =$	4.0	4.0
1 (15-16)	Eu-Eu	59859.504	7.10	1	2.0	2.0
2 (16-17)	Eu-Eu	59866.610	5.35	1	7.0	7.0
				$\Delta V_{tot} =$	9.0	9.0
1 (15-16)	Eu-Eu	59857.727	5.26	1	197.0	197.0
2 (16-17)	Eu-Eu	59862.990	5.76	1	125569.0	499.0
3 (17-18)	Eu-Io	59869.747	3.80	1	61115.0	294.0
				$\Delta V_{tot} =$	186881.0	990.0
1 (15-16)	Eu-Eu	59857.544	5.28	1	185.0	185.0
2 (16-17)	Eu-Eu	59862.819	6.96	1	81163.0	413.0
3 (17-18)	Eu-Io	59869.777	3.68	1	230304.0	595.0
4 (18-19)	Io-Io	59873.454	3.51	1	120.0	120.0
				$\Delta V_{tot} =$	311772.0	1313.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

D.2 Optimisation of Sequence Ga-Ga-Ca-Ca-Ga for ΔV

D.2.1 Full Freedom and One Flyby Window

Table D.11: *Results (3x) of subsets for the gravity-assist manoeuvres for the first sub-sequence Ga-Ga-Ca-Ca-Ga. [Settings: full freedom and 1 flyby window]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60514.521	314.3	431.3	1098.51
2 (104)	Ga	60525.265	314.7	581.3	1097.42
1 (103)	Ga	60515.381	1462.0	496.0	1098.51
2 (104)	Ga	60526.120	1467.4	60.8	1097.50
3 (105)	Ca	60543.255	1246.9	1797.3	1096.93
1 (103)	Ga	60515.123	1329.9	883.0	1098.51
2 (104)	Ga	60525.865	1327.6	50.0	1093.25
3 (105)	Ca	60543.439	1090.0	544.6	1077.88
4 (106)	Ca	60566.22	1084.9	884.4	1077.35
1 (103)	Ga	60515.669	1399.7	200.5	1098.51
2 (104)	Ga	60526.405	1405.4	50.0	1097.46
3 (105)	Ca	60544.811	1286.1	50.0	1087.83
4 (106)	Ca	60569.852	1297.8	267.2	1080.94
5 (107)	Ga	60587.379	1485.7	1584.7	1080.43
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60485.715	354.1	144.7	1098.51
2 (104)	Ga	60500.026	355.0	1557.6	1096.93
1 (103)	Ga	60486.464	1448.3	457.9	1098.51
2 (104)	Ga	60500.778	1448.3	81.9	1097.41
3 (105)	Ca	60518.504	1258.1	667.0	1096.87
1 (103)	Ga	60486.774	1396.1	954.6	1098.51
2 (104)	Ga	60501.078	1404.3	122.7	1096.41
3 (105)	Ca	60518.396	1169.0	50.0	1089.49
4 (106)	Ca	60543.384	1155.8	1120.8	1088.50
1 (103)	Ga	60486.700	1478.9	769.0	1098.51
2 (104)	Ga	60501.006	1478.9	66.0	1097.08
3 (105)	Ca	60519.387	1258.1	74.4	1089.66
4 (106)	Ca	60544.404	1244.3	76.6	1088.29
5 (107)	Ga	60562.298	1437.4	1235.7	1086.20
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60488.667	310.9	1696.9	1098.51
2 (104)	Ga	60499.394	310.8	1411.8	1097.27
1 (103)	Ga	60486.687	1467.8	621.0	1098.51
2 (104)	Ga	60500.993	1467.5	107.4	1096.24
3 (105)	Ca	60518.084	1258.7	1363.8	1095.69
1 (103)	Ga	60486.422	1441.2	1113.1	1098.51
2 (104)	Ga	60500.736	1441.2	50.0	1097.40
3 (105)	Ca	60518.258	1230.5	50.0	1092.31
4 (106)	Ca	60543.244	1218.5	1273.1	1092.30
1 (103)	Ga	60486.860	1473.8	802.0	1098.51
2 (104)	Ga	60501.171	1472.8	61.0	1096.72
3 (105)	Ca	60519.107	1342.6	135.4	1095.91
4 (106)	Ca	60544.115	1324.1	58.8	1095.76
5 (107)	Ga	60562.084	1554.4	1680.9	1093.70

[1] Number between parenthesis is the actual flyby number in the CSU solution.

Table D.12: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Ga-Ga-Ca-Ca-Ga. [Settings: full freedom and 1 flyby window]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60514.521	10.74	1	2.0	2.0
				$\Delta V_{tot} =$	2.0	2.0
1 (103-104)	Ga-Ga	60515.381	10.74	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60526.120	17.14	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
1 (103-104)	Ga-Ga	60515.123	10.74	1	77.0	77.0
2 (104-105)	Ga-Ca	60525.865	17.57	1	269.0	269.0
3 (105-106)	Ca-Ca	60543.439	22.78	1	10.0	10.0
				$\Delta V_{tot} =$	356.0	356.0
1 (103-104)	Ga-Ga	60515.669	10.74	1	1.0	1.0
2 (104-105)	Ga-Ca	60526.405	18.41	1	164.0	164.0
3 (105-106)	Ca-Ca	60544.811	25.04	1	125.0	125.0
4 (106-107)	Ca-Ga	60569.852	17.53	1	4.0	4.0
				$\Delta V_{tot} =$	294.0	294.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60485.715	14.31	1	2.0	2.0
				$\Delta V_{tot} =$	2.0	2.0
1 (103-104)	Ga-Ga	60486.464	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60500.778	17.73	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
1 (103-104)	Ga-Ga	60486.774	14.30	1	8.0	8.0
2 (104-105)	Ga-Ca	60501.078	17.32	1	115.0	115.0
3 (105-106)	Ca-Ca	60518.396	24.99	1	18.0	18.0
				$\Delta V_{tot} =$	141.0	141.0
1 (103-104)	Ga-Ga	60486.700	14.31	1	6.0	6.0
2 (104-105)	Ga-Ca	60501.006	18.38	1	124.0	124.0
3 (105-106)	Ca-Ca	60519.387	25.02	1	25.0	25.0
4 (106-107)	Ca-Ga	60544.404	17.89	1	33.0	33.0
				$\Delta V_{tot} =$	188.0	188.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60488.667	10.73	1	4.0	4.0
				$\Delta V_{tot} =$	4.0	4.0
1 (103-104)	Ga-Ga	60486.687	14.31	1	14.0	14.0
2 (104-105)	Ga-Ca	60500.993	17.09	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	14.0	14.0
1 (103-104)	Ga-Ga	60486.42	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60500.736	17.52	1	81.0	81.0
3 (105-106)	Ca-Ca	60518.258	24.99	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	81.0	81.0
1 (103-104)	Ga-Ga	60486.860	14.31	1	3.0	3.0
2 (104-105)	Ga-Ca	60501.171	17.94	1	5.0	5.0
3 (105-106)	Ca-Ca	60519.107	25.00	1	3.0	3.0
4 (106-107)	Ca-Ga	60544.115	17.97	1	27.0	27.0
				$\Delta V_{tot} =$	38.0	38.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

D.2.2 Full Freedom and Three Flyby Window

Table D.13: *Results (3x) of subsets for the gravity-assist manoeuvres for the first sub-sequence Ga-Ga-Ca-Ca-Ga. [Settings: full freedom and 3 flyby window]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60495.180	315.9	1695.4	1098.51
2 (104)	Ga	60509.491	315.7	150.7	1097.44
1 (103)	Ga	60487.262	1465.3	1856.6	1098.51
2 (104)	Ga	60501.523	1540.5	448.2	1092.83
3 (105)	Ca	60520.078	1563.8	987.9	1089.00
1 (103)	Ga	60487.070	1188.3	63.9	1098.51
2 (104)	Ga	60501.264	1422.4	153.7	1086.88
3 (105)	Ca	60519.112	1182.1	140.5	1077.51
4 (106)	Ca	60543.264	1162.8	290.4	1073.83
1 (103)	Ga	60486.673	1480.3	1980.8	1098.51
2 (104)	Ga	60500.987	1480.3	50.0	1097.43
3 (105)	Ca	60519.173	1309.7	50.0	1088.94
4 (106)	Ca	60544.183	1294.4	50.1	1088.54
5 (107)	Ga	60562.247	1479.5	326.7	1087.96
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60495.750	423.2	782.8	1098.51
2 (104)	Ga	60510.064	423.2	1513.7	1097.41
1 (103)	Ga	60486.515	1419.7	161.2	1098.51
2 (104)	Ga	60500.828	1423.5	53.4	1096.67
3 (105)	Ca	60518.450	1252.3	1177.3	1094.85
1 (103)	Ga	60486.784	1460.1	923.2	1098.51
2 (104)	Ga	60501.096	1460.7	183.7	1096.74
3 (105)	Ca	60518.486	1246.0	73.9	1094.46
4 (106)	Ca	60543.478	1232.7	196.2	1094.42
1 (103)	Ga	60486.604	1461.4	834.9	1098.51
2 (104)	Ga	60500.914	1453.2	51.3	1096.98
3 (105)	Ca	60519.040	1257.4	50.0	1090.28
4 (106)	Ca	60544.046	1258.0	67.5	1087.05
5 (107)	Ga	60562.345	1449.1	1960.0	1083.65
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60517.338	322.5	939.2	1098.51
2 (104)	Ga	60528.065	322.7	567.8	1097.45
1 (103)	Ga	60527.984	1580.7	1987.1	1098.51
2 (104)	Ga	60538.712	1546.3	51.9	1092.45
3 (105)	Ca	60555.817	1303.1	970.8	1088.62
1 (103)	Ga	60528.040	1303.0	1848.0	1098.51
2 (104)	Ga	60538.448	1328.7	50.0	1093.51
3 (105)	Ca	60555.878	1159.3	292.1	1083.68
4 (106)	Ca	60580.982	1145.8	1096.0	1082.36
1 (103)	Ga	60528.033	1504.9	349.0	1098.51
2 (104)	Ga	60538.777	1503.6	249.4	1097.50
3 (105)	Ca	60556.816	1381.1	60.9	1096.94
4 (106)	Ca	60581.931	1372.6	162.8	1096.93
5 (107)	Ga	60599.615	1703.7	926.6	1096.33

[1] Number between parenthesis is the actual flyby number in the CSU solution.

Table D.14: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Ga-Ga-Ca-Ca-Ga. [Settings: full freedom and 3 flyby window]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60495.180	14.31	1	1.0	1.0
				$\Delta V_{tot} =$	1.0	1.0
1 (103-104)	Ga-Ga	60487.262	14.26	1	72.0	72.0
2 (104-105)	Ga-Ca	60501.523	18.56	1	59.0	59.0
				$\Delta V_{tot} =$	131.0	131.0
1 (103-104)	Ga-Ga	60487.070	14.19	1	180.0	180.0
2 (104-105)	Ga-Ca	60501.264	17.85	1	161.0	161.0
3 (105-106)	Ca-Ca	60519.112	24.15	1	67.0	67.0
				$\Delta V_{tot} =$	408.0	408.0
1 (103-104)	Ga-Ga	60486.673	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60500.987	18.19	1	143.0	143.0
3 (105-106)	Ca-Ca	60519.173	25.01	1	7.0	7.0
4 (106-107)	Ca-Ga	60544.183	18.06	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	150.0	150.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60495.750	14.31	1	2.0	2.0
				$\Delta V_{tot} =$	2.0	2.0
1 (103-104)	Ga-Ga	60486.515	14.31	1	16.0	16.0
2 (104-105)	Ga-Ca	60500.828	17.62	1	23.0	23.0
				$\Delta V_{tot} =$	39.0	39.0
1 (103-104)	Ga-Ga	60486.784	14.31	1	2.0	2.0
2 (104-105)	Ga-Ca	60501.096	17.391	1	31.0	31.0
3 (105-106)	Ca-Ca	60518.486	24.99	1	1.0	1.0
				$\Delta V_{tot} =$	34.0	34.0
1 (103-104)	Ga-Ga	60486.604	14.31	1	7.0	7.0
2 (104-105)	Ga-Ca	60500.915	18.18	1	111.0	111.0
3 (105-106)	Ca-Ca	60519.092	25.02	1	70.0	70.0
4 (106-107)	Ca-Ga	60544.107	18.15	1	33.0	33.0
				$\Delta V_{tot} =$	221.0	221.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60517.338	10.73	1	1.0	1.0
				$\Delta V_{tot} =$	1.0	1.0
1 (103-104)	Ga-Ga	60527.984	10.73	1	92.0	92.0
2 (104-105)	Ga-Ca	60538.712	17.11	1	59.0	59.0
				$\Delta V_{tot} =$	151.0	151.0
1 (103-104)	Ga-Ga	60528.040	10.41	1	72.0	72.0
2 (104-105)	Ga-Ca	60538.448	17.43	1	168.0	168.0
3 (105-106)	Ca-Ca	60555.878	25.10	1	24.0	24.0
				$\Delta V_{tot} =$	264.0	264.0
1 (103-104)	Ga-Ga	60528.033	10.74	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60538.777	18.04	1	∼0.0	∼0.0
3 (105-106)	Ca-Ca	60556.816	25.12	1	∼0.0	∼0.0
4 (106-107)	Ca-Ga	60581.931	17.68	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼1.0	∼1.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

D.2.3 Small Freedom and One Flyby Window

Table D.15: *Results (3x) of subsets for the gravity-assist manoeuvres for the first sub-sequence Ga-Ga-Ca-Ca-Ga. [Settings: small freedom and 1 flyby window]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60486.255	316.7	259.4	1098.51
2 (104)	Ga	60496.998	317.6	1917.7	1097.43
1 (103)	Ga	60486.785	1492.6	504.3	1098.51
2 (104)	Ga	60501.099	1492.6	323.8	1096.83
3 (105)	Ca	60518.237	1244.5	1333.7	1096.28
1 (103)	Ga	60486.450	1433.6	1078.6	1098.51
2 (104)	Ga	60500.764	1433.6	50.0	1097.42
3 (105)	Ca	60518.305	1230.1	50.2	1092.44
4 (106)	Ca	60543.293	1217.8	390.1	1092.44
1 (103)	Ga	60486.944	1663.0	658.4	1098.51
2 (104)	Ga	60501.258	1663.0	54.8	1097.57
3 (105)	Ca	60518.930	1372.9	492.7	1097.01
4 (106)	Ca	60543.932	1356.2	50.0	1097.01
5 (107)	Ga	60562.125	1551.1	761.35	1096.45
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60510.368	335.7	1899.4	1098.51
2 (104)	Ga	60521.094	336.2	1057.2	1097.47
1 (103)	Ga	60512.221	1513.5	1087.2	1098.51
2 (104)	Ga	60526.526	1514.4	51.3	1096.60
3 (105)	Ca	60544.760	1378.7	742.3	1096.03
1 (103)	Ga	60512.032	1507.4	428.6	1098.51
2 (104)	Ga	60526.331	1514.4	420.1	1096.38
3 (105)	Ca	60543.636	1224.5	50.3	1094.90
4 (106)	Ca	60568.730	1237.0	1129.3	1094.71
1 (103)	Ga	60512.024	1508.8	129.3	1098.51
2 (104)	Ga	60526.337	1507.7	50.0	1097.48
3 (105)	Ca	60544.433	1345.4	74.1	1092.16
4 (106)	Ca	60569.506	1364.4	63.1	1092.09
5 (107)	Ga	60587.352	1531.4	1217.1	1091.51
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60497.582	356.5	649.8	1098.51
2 (104)	Ga	60508.316	354.2	1005.2	1097.42
1 (103)	Ga	60499.015	1499.4	634.3	1098.51
2 (104)	Ga	60513.327	1499.3	194.2	1097.35
3 (105)	Ca	60531.133	1229.1	174.5	1096.79
1 (103)	Ga	60499.039	1448.7	478.5	1098.51
2 (104)	Ga	60513.353	1448.6	54.1	1097.39
3 (105)	Ca	60531.073	1211.5	50.6	1092.88
4 (106)	Ca	60556.060	1223.4	1280.3	1092.87
1 (103)	Ga	60499.417	1493.1	87.4	1098.51
2 (104)	Ga	60513.731	1492.9	50.0	1097.55
3 (105)	Ca	60532.063	1355.8	50.0	1095.69
4 (106)	Ca	60557.034	1362.8	232.5	1095.68
5 (107)	Ga	60574.665	1696.2	1022.4	1095.07

[1] Number between parenthesis is the actual flyby number in the CSU solution.

Table D.16: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Ga-Ga-Ca-Ca-Ga. [Settings: small freedom and 1 flyby window]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60486.255	10.74	1	2.0	2.0
				$\Delta V_{tot} =$	2.0	2.0
1 (103-104)	Ga-Ga	60486.785	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60501.099	17.14	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
1 (103-104)	Ga-Ga	60486.450	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60500.764	17.54	1	80.0	80.0
3 (105-106)	Ca-Ca	60518.305	24.99	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	80.0	80.0
1 (103-104)	Ga-Ga	60486.944	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60501.258	17.67	1	∼0.0	∼0.0
3 (105-106)	Ca-Ca	60518.930	25.00	1	∼0.0	∼0.0
4 (106-107)	Ca-Ga	60543.932	18.19	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60510.368	10.73	1	1.0	1.0
				$\Delta V_{tot} =$	1.0	1.0
1 (103-104)	Ga-Ga	60512.221	14.31	1	4.0	4.0
2 (104-105)	Ga-Ca	60526.526	18.23	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	4.0	4.0
1 (103-104)	Ga-Ga	60512.032	14.30	1	8.0	8.0
2 (104-105)	Ga-Ca	60526.331	17.31	1	16.0	16.0
3 (105-106)	Ca-Ca	60543.636	25.09	1	3.0	3.0
				$\Delta V_{tot} =$	27.0	27.0
1 (103-104)	Ga-Ga	60512.024	14.31	1	2.0	2.0
2 (104-105)	Ga-Ca	60526.337	18.10	1	85.0	85.0
3 (105-106)	Ca-Ca	60544.433	25.07	1	1.0	1.0
4 (106-107)	Ca-Ga	60569.506	17.85	1	1.0	1.0
				$\Delta V_{tot} =$	89.0	89.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60497.582	10.73	1	2.0	2.0
				$\Delta V_{tot} =$	2.0	2.0
1 (103-104)	Ga-Ga	60499.015	14.31	1	1.0	1.0
2 (104-105)	Ga-Ca	60513.327	17.81	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	1.0	1.0
1 (103-104)	Ga-Ga	60499.039	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60513.353	17.72	1	71.0	71.0
3 (105-106)	Ca-Ca	60531.073	24.99	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	71.0	71.0
1 (103-104)	Ga-Ga	60499.417	14.31	1	1.0	1.0
2 (104-105)	Ga-Ca	60513.731	18.33	1	23.0	23.0
3 (105-106)	Ca-Ca	60532.063	24.97	1	∼0.0	∼0.0
4 (106-107)	Ca-Ga	60557.034	17.63	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	24.0	24.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

D.2.4 Small Freedom and Two Flyby Window

Table D.17: *Results (3x) of subsets for the gravity-assist manoeuvres for the first sub-sequence Ga-Ga-Ca-Ca-Ga. [Settings: small freedom and 2 flyby window]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60498.370	303.2	237.2	1098.51
2 (104)	Ga	60512.682	303.8	510.2	1097.50
1 (103)	Ga	60499.284	1494.8	748.7	1098.51
2 (104)	Ga	60513.598	1494.8	55.2	1097.08
3 (105)	Ca	60531.551	1278.1	911.3	1096.54
1 (103)	Ga	60498.974	1397.6	213.0	1098.51
2 (104)	Ga	60513.269	1398.3	50.0	1096.09
3 (105)	Ca	60530.872	1173.1	85.3	1088.95
4 (106)	Ca	60555.862	1185.1	1234.0	1088.62
1 (103)	Ga	60499.156	1506.7	349.9	1098.51
2 (104)	Ga	60513.470	1506.7	112.7	1097.39
3 (105)	Ca	60531.914	1329.2	50	1096.83
4 (106)	Ca	60556.863	1328.9	612.4	1089.90
5 (107)	Ga	60575.076	1453.1	1139.2	1089.35
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60519.820	328.3	345.8	1098.51
2 (104)	Ga	60530.559	327.5	1338.7	1097.45
1 (103)	Ga	60516.440	1515.5	1214.1	1098.51
2 (104)	Ga	60526.832	1516.0	178.0	1097.47
3 (105)	Ca	60546.109	1602.7	1381.9	1096.88
1 (103)	Ga	60515.116	1283.5	79.0	1098.51
2 (104)	Ga	60525.857	1287.7	50.0	1097.52
3 (105)	Ca	60543.183	1127.7	50.0	1082.47
4 (106)	Ca	60568.287	1138.6	944.3	1082.47
1 (103)	Ga	60511.814	1432.7	1820.2	1098.51
2 (104)	Ga	60526.070	1446.9	50.0	1095.34
3 (105)	Ca	60544.276	1219.3	145.4	1089.84
4 (106)	Ca	60569.357	1239.0	55.6	1089.14
5 (107)	Ga	60587.331	1434.2	150.4	1082.48
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60521.695	332.2	1177.9	1098.51
2 (104)	Ga	60532.439	332.7	1877.9	1097.46
1 (103)	Ga	60524.295	1450.7	331.8	1098.51
2 (104)	Ga	60538.609	1450.7	173.0	1097.08
3 (105)	Ca	60555.598	1262.5	1301.0	1096.54
1 (103)	Ga	60524.030	1468.8	274.0	1098.51
2 (104)	Ga	60538.344	1468.8	162.8	1097.39
3 (105)	Ca	60556.067	1265.2	66.0	1096.84
4 (106)	Ca	60556.067	1253.0	585.3	1096.84
1 (103)	Ga	60524.644	1671.8	126.2	1098.51
2 (104)	Ga	60538.958	1671.8	392.0	1096.78
3 (105)	Ca	60556.513	1384.7	224.2	1096.20
4 (106)	Ca	60581.624	1374.1	98.26	1096.20
5 (107)	Ga	60599.812	1501.8	938.4	1095.65

[1] Number between parenthesis is the actual flyby number in the CSU solution.

Table D.18: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Ga-Ga-Ca-Ca-Ga. [Settings: small freedom and 2 flyby window]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60498.370	14.31	1	1.0	1.0
				$\Delta V_{tot} =$	1.0	1.0
1 (103-104)	Ga-Ga	60499.284	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60513.598	17.95	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
1 (103-104)	Ga-Ga	60498.974	14.29	1	25.0	25.0
2 (104-105)	Ga-Ca	60513.269	17.60	1	119.0	119.0
3 (105-106)	Ca-Ca	60530.872	24.99	1	6.0	6.0
				$\Delta V_{tot} =$	150.0	150.0
1 (103-104)	Ga-Ga	60499.156	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60513.470	18.44	1	∼0.0	∼0.0
3 (105-106)	Ca-Ca	60531.914	24.95	1	124.0	124.0
4 (106-107)	Ca-Ga	60556.863	18.21	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	124.0	124.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60519.820	10.74	1	1.0	1.0
				$\Delta V_{tot} =$	1.0	1.0
1 (103-104)	Ga-Ga	60516.440	10.39	1	1.0	1.0
2 (104-105)	Ga-Ca	60526.832	19.28	1	1.0	1.0
				$\Delta V_{tot} =$	2.0	2.0
1 (103-104)	Ga-Ga	60515.116	10.74	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60525.857	17.33	1	262.0	262.0
3 (105-106)	Ca-Ca	60543.183	25.10	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	262.0	262.0
1 (103-104)	Ga-Ga	60511.814	14.26	1	37.0	37.0
2 (104-105)	Ga-Ca	60526.070	18.21	1	89.0	89.0
3 (105-106)	Ca-Ca	60544.276	25.08	1	13.0	13.0
4 (106-107)	Ca-Ga	60569.357	17.97	1	116.0	116.0
				$\Delta V_{tot} =$	255.0	255.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60521.695	10.74	1	1.0	1.0
				$\Delta V_{tot} =$	1.0	1.0
1 (103-104)	Ga-Ga	60524.295	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60538.609	16.99	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
1 (103-104)	Ga-Ga	60524.030	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60538.344	17.72	1	∼0.0	∼0.0
3 (105-106)	Ca-Ca	60556.067	25.10	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
1 (103-104)	Ga-Ga	60524.644	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60538.958	17.56	1	∼0.0	∼0.0
3 (105-106)	Ca-Ca	60556.513	25.11	1	∼0.0	∼0.0
4 (106-107)	Ca-Ga	60581.624	18.19	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

D.2.5 Small Freedom and Three Flyby Window

Table D.19: *Results (3x) of subsets for the gravity-assist manoeuvres for the first sub-sequence Ga-Ga-Ca-Ca-Ga. [Settings: small freedom and 3 flyby window]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60507.549	315.6	127.9	1098.51
2 (104)	Ga	60521.863	313.5	1201.6	1096.91
1 (103)	Ga	60511.945	1484.8	684.7	1098.51
2 (104)	Ga	60526.259	1484.8	258.4	1096.85
3 (105)	Ca	60543.382	1224.1	1264.6	1096.29
1 (103)	Ga	60511.622	1390.9	433.7	1098.51
2 (104)	Ga	60525.904	1415.5	50.0	1095.88
3 (105)	Ca	60543.625	1214.5	57.9	1091.57
4 (106)	Ca	60568.721	1225.1	417.7	1090.96
1 (103)	Ga	60511.750	1491.9	1002.9	1098.51
2 (104)	Ga	60526.064	1491.9	102.7	1097.39
3 (105)	Ca	60544.486	1336.4	50.0	1096.83
4 (106)	Ca	60569.539	1335.4	439.2	1089.29
5 (107)	Ga	60587.399	1461.6	717.3	1088.75
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60502.093	307.3	185.3	1098.51
2 (104)	Ga	60512.824	308.3	974.3	1097.41
1 (103)	Ga	60503.260	1460.6	250.0	1098.51
2 (104)	Ga	60513.639	1460.6	285.6	1097.51
3 (105)	Ca	60530.695	1266.7	636.3	1096.96
1 (103)	Ga	60502.933	1500.7	1213.0	1098.51
2 (104)	Ga	60513.660	1502.4	60.4	1097.52
3 (105)	Ca	60530.954	1250.5	745.1	1096.97
4 (106)	Ca	60555.943	1263.3	990.5	1096.97
1 (103)	Ga	60503.465	1530.6	1356.1	1098.51
2 (104)	Ga	60513.860	1530.7	50	1097.50
3 (105)	Ca	60532.157	1379.8	689.7	1096.49
4 (106)	Ca	60556.556	1379.7	787.0	1096.49
5 (107)	Ga	60574.957	1447.8	417.9	1095.94
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	$m_{post\,flyby}$ [kg]
1 (103)	Ga	60493.449	312.8	237.6	1098.51
2 (104)	Ga	60507.762	312.3	1401.6	1097.44
1 (103)	Ga	60490.717	1493.5	1342.9	1098.51
2 (104)	Ga	60501.103	1493.5	1132.2	1097.51
3 (105)	Ca	60518.035	1254.3	1548.2	1096.95
1 (103)	Ga	60490.108	1326.9	488.9	1098.51
2 (104)	Ga	60500.838	1322.5	50.0	1097.52
3 (105)	Ca	60518.159	1148.5	50.0	1085.31
4 (106)	Ca	60543.144	1137.7	983.3	1085.31
1 (103)	Ga	60490.135	1286.4	669.0	1098.51
2 (104)	Ga	60500.868	1295.6	50.0	1095.48
3 (105)	Ca	60518.552	1235.7	50.0	1084.70
4 (106)	Ca	60543.658	1242.4	50.0	1080.29
5 (107)	Ga	60561.957	1514.2	939.2	1072.45

[1] Number between parenthesis is the actual flyby number in the CSU solution.

Table D.20: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Ga-Ga-Ca-Ca-Ga. [Settings: small freedom and 3 flyby window]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60507.549	14.31	1	3.0	3.0
				$\Delta V_{tot} =$	3.0	3.0
1 (103-104)	Ga-Ga	60511.945	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60526.259	17.12	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
1 (103-104)	Ga-Ga	60511.622	14.28	1	28.0	28.0
2 (104-105)	Ga-Ca	60525.904	17.72	1	67.0	67.0
3 (105-106)	Ca-Ca	60543.625	25.10	1	11.0	11.0
				$\Delta V_{tot} =$	106.0	106.0
1 (103-104)	Ga-Ga	60511.750	14.31	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60526.064	18.42	1	∼0.0	∼0.0
3 (105-106)	Ca-Ca	60544.486	25.05	1	135.0	135.0
4 (106-107)	Ca-Ga	60569.539	17.86	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	135.0	135.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60502.093	10.73	1	2.0	2.0
				$\Delta V_{tot} =$	2.0	2.0
1 (103-104)	Ga-Ga	60503.260	10.38	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60513.639	17.06	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
1 (103-104)	Ga-Ga	60502.933	10.73	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60513.660	17.29	1	∼0.0	∼0.0
3 (105-106)	Ca-Ca	60530.954	24.99	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
1 (103-104)	Ga-Ga	60503.465	10.40	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60513.860	18.30	1	8.0	8.0
3 (105-106)	Ca-Ca	60532.157	24.40	1	∼0.0	∼0.0
4 (106-107)	Ca-Ga	60556.556	18.40	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	8.0	8.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (103-104)	Ga-Ga	60493.449	14.31	1	2.0	2.0
				$\Delta V_{tot} =$	2.0	2.0
1 (103-104)	Ga-Ga	60490.717	10.39	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60501.103	16.93	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	∼0.0	∼0.0
1 (103-104)	Ga-Ga	60490.108	10.73	1	∼0.0	∼0.0
2 (104-105)	Ga-Ca	60500.838	17.32	1	211.0	211.0
3 (105-106)	Ca-Ca	60518.159	24.99	1	∼0.0	∼0.0
				$\Delta V_{tot} =$	211.0	211.0
1 (103-104)	Ga-Ga	60490.135	10.73	1	37.0	37.0
2 (104-105)	Ga-Ca	60500.868	17.68	1	185.0	185.0
3 (105-106)	Ca-Ca	60518.552	25.11	1	80.0	80.0
4 (106-107)	Ca-Ga	60543.658	18.30	1	134.0	134.0
				$\Delta V_{tot} =$	436.0	436.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

D.3 Optimisation of Sequence Eu-Eu-Eu-Io-Io for ΔV and Moon Face Points

D.3.1 $W_{\Delta V} = 10.0$ and $W_M = 1.0$

Table D.21: *Results (3x) of subsets for the gravity-assist manoeuvres for the sub-sequence Eu-Eu-Eu-Io-Io. [Settings: small freedom and 2 flyby window and $W_{\Delta V} = 10.0$ and $W_M = 1.0$]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	# Face	Face value	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59873.789	307.7	1479.3	19	6	1889.67
2 (16)	Eu	59880.894	306.4	656.5	25	6	1885.20
1 (15)	Eu	59873.631	311.7	403.2	24	6	1889.67
2 (16)	Eu	59880.725	313.3	1171.4	18	6	1886.14
3 (17)	Eu	59887.830	313.6	1345.1	25	6	1881.22
1 (15)	Eu	59874.082	522.6	418.8	25	6	1889.67
2 (16)	Eu	59879.598	578.1	1242.9	23	6	1861.21
3 (17)	Eu	59886.584	1544.8	626.1	16	6	1777.71
4 (18)	Io	59890.912	1866.0	685.8	16	3	1701.00
1 (15)	Eu	59874.922	727.0	931.8	15	6	1889.67
2 (16)	Eu	59880.127	648.2	181.6	18	6	1869.55
3 (17)	Eu	59887.149	946.4	207.6	23	6	1819.31
4 (18)	Io	59891.165	1455.8	298.9	21	3	1707.90
5 (19)	Io	59893.705	1294.5	1873.0	15	3	1684.28
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	# Face	Face value	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59874.523	305.4	1089.4	18	6	1889.67
2 (16)	Eu	59878.076	306.7	1900.2	24	6	1888.01
1 (15)	Eu	59873.528	300.0	1058.1	24	6	1889.67
2 (16)	Eu	59880.633	300.0	74.7	19	6	1885.14
3 (17)	Eu	59887.720	300.0	703.3	25	6	1880.74
1 (15)	Eu	59875.082	502.4	567.9	15	6	1889.67
2 (16)	Eu	59881.637	687.4	1639.5	18	6	1857.05
3 (17)	Eu	59887.224	1464.5	62.3	20	6	1763.40
4 (18)	Io	59891.062	1787.8	1577.1	24	3	1698.09
1 (15)	Eu	59876.226	564.9	1352.7	18	6	1889.67
2 (16)	Eu	59883.209	613.0	1999.7	24	6	1861.18
3 (17)	Eu	59887.296	1242.7	124.9	21	6	1812.22
4 (18)	Io	59891.156	1848.8	1354.3	21	3	1716.14
5 (19)	Io	59893.702	1851.2	1354.1	26	3	1708.11
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	# Face	Face value	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59869.810	310.4	1287.2	24	6	1889.67
2 (16)	Eu	59875.162	308.9	818.7	25	6	1886.27
1 (15)	Eu	59871.225	300.4	863.7	19	6	1889.67
2 (16)	Eu	59878.320	300.0	110.1	25	6	1884.51
3 (17)	Eu	59885.425	300.0	1286.0	18	6	1881.43
1 (15)	Eu	59871.851	463.1	310.3	19	6	1889.67
2 (16)	Eu	59878.957	626.4	115.9	23	6	1864.23
3 (17)	Eu	59883.798	1372.5	50.9	21	6	1806.76
4 (18)	Io	59887.864	1909.9	366.6	23	3	1732.94
1 (15)	Eu	59870.529	503.0	614.3	25	6	1889.67
2 (16)	Eu	59877.607	306.2	1920.7	19	6	1858.85
3 (17)	Eu	59883.637	913.5	965.2	22	6	1799.33
4 (18)	Io	59887.610	1355.9	50.0	20	3	1677.35
5 (19)	Io	59890.269	1297.5	1137.4	17	3	1653.84

[1] Number between parenthesis is the actual flyby number in the CSU solution.

Table D.22: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: full freedom and 2 flyby window and $W_{\Delta V} = 10.0$ and $W_M = 1.0$]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59873.789	7.11	1	2.0	2.0
$f =$	-4.09647			$\Delta V_{tot} =$	2.0	2.0
1 (15-16)	Eu-Eu	59873.631	7.09	1	7.0	7.0
2 (16-17)	Eu-Eu	59880.725	7.11	1	7.0	7.0
$f =$	-2.20927			$\Delta V_{tot} =$	14.0	14.0
1 (15-16)	Eu-Eu	59874.082	5.52	1	40538.0	269.0
2 (16-17)	Eu-Eu	59879.598	6.99	1	368523.0	863.0
3 (17-18)	Eu-Io	59886.584	4.33	1	458558.0	839.0
$f =$	-0.8750346			$\Delta V_{tot} =$	867619.0	1971.0
1 (15-16)	Eu-Eu	59874.922	5.21	1	21001.0	184.0
2 (16-17)	Eu-Eu	59880.127	7.02	1	130504.0	505.0
3 (17-18)	Eu-Io	59887.149	4.02	1	971209.0	1226.0
3 (18-19)	Io-Io	59891.165	2.54	1	66773.0	240.0
$f =$	-0.8000336			$\Delta V_{tot} =$	1189487.0	2155.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59874.523	3.55	1	3.0	3.0
$f =$	-3.78519			$\Delta V_{tot} =$	3.0	3.0
1 (15-16)	Eu-Eu	59873.528	7.11	1	3.0	3.0
2 (16-17)	Eu-Eu	59880.633	7.09	1	15.0	15.0
$f =$	-2.01681			$\Delta V_{tot} =$	18.0	18.0
1 (15-16)	Eu-Eu	59875.082	6.56	1	46607.0	302.0
2 (16-17)	Eu-Eu	59881.637	5.59	1	440756.0	986.0
3 (17-18)	Eu-Io	59887.224	3.84	1	365928.0	727.0
$f =$	-0.8750352			$\Delta V_{tot} =$	853291.0	2015.0
1 (15-16)	Eu-Eu	59876.226	6.99	1	254.0	254.0
2 (16-17)	Eu-Eu	59883.209	4.09	1	225083.0	499.0
3 (17-18)	Eu-Io	59887.296	3.86	1	653446.0	1042.0
3 (18-19)	Io-Io	59891.156	2.55	1	61.0	61.0
$f =$	-0.8000455			$\Delta V_{tot} =$	878844.0	1856.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59869.810	5.35	1	6.0	6.0
$f =$	-2.38975			$\Delta V_{tot} =$	6.0	6.0
1 (15-16)	Eu-Eu	59871.225	7.10	1	10.0	10.0
2 (16-17)	Eu-Eu	59878.320	7.10	1	2.0	2.0
$f =$	-2.41664			$\Delta V_{tot} =$	12.0	12.0
1 (15-16)	Eu-Eu	59871.851	7.11	1	222.0	222.0
2 (16-17)	Eu-Eu	59878.957	4.84	1	285861.0	591.0
3 (17-18)	Eu-Io	59883.798	4.07	1	417254.0	779.0
$f =$	-0.8750427			$\Delta V_{tot} =$	703337.0	1592.0
1 (15-16)	Eu-Eu	59870.529	7.08	1	280.0	280.0
2 (16-17)	Eu-Eu	59877.607	6.03	1	263257.0	611.0
3 (17-18)	Eu-Io	59883.637	3.97	1	1187148.0	1363.0
3 (18-19)	Io-Io	59887.610	2.66	1	71584.0	237.0
$f =$	-0.8000263			$\Delta V_{tot} =$	1522269.0	2491.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

D.3.2 $W_{\Delta V} = 1.0$ and $W_M = 1.0$

Table D.23: *Results (3x) of subsets for the gravity-assist manoeuvres for the sub-sequence Eu-Eu-Eu-Io-Io. [Settings: small freedom and 2 flyby window and $W_{\Delta V} = 1.0$ and $W_M = 1.0$]*

$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	# Face	Face value	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59889.420	321.8	182.0	25	6	1889.67
2 (16)	Eu	59894.724	324.6	261.1	24	6	1886.15
1 (15)	Eu	59887.855	300.0	1349.3	19	6	1889.67
2 (16)	Eu	59894.957	301.0	50.5	25	6	1884.97
3 (17)	Eu	59902.057	300.0	1417.0	16	6	1879.63
1 (15)	Eu	59886.158	346.7	510.0	18	6	1889.67
2 (16)	Eu	59892.964	577.7	52.2	26	6	1851.61
3 (17)	Eu	59897.844	1477.9	108.5	22	6	1793.21
4 (18)	Io	59901.751	1781.4	1466.5	24	3	1741.08
1 (15)	Eu	59887.513	598.4	54.0	21	6	1889.67
2 (16)	Eu	59892.795	988.1	50.0	24	6	1854.25
3 (17)	Eu	59898.055	1373.9	51.1	23	6	1798.32
4 (18)	Io	59901.839	1426.6	333.6	21	3	1703.93
5 (19)	Io	59904.448	1472.1	1403.3	23	3	1675.63
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	# Face	Face value	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59902.729	323.5	317.8	24	6	1889.67
2 (16)	Eu	59906.280	320.6	343.6	17	6	1886.50
1 (15)	Eu	59901.551	308.8	1755.4	18	6	1889.67
2 (16)	Eu	59908.654	310.9	172.3	24	6	1885.23
3 (17)	Eu	59912.215	307.6	1068.6	19	6	1880.97
1 (15)	Eu	59901.410	340.5	81.1	25	6	1889.67
2 (16)	Eu	59908.286	301.1	1367.1	18	6	1864.93
3 (17)	Eu	59915.305	1235.9	58.0	21	6	1797.24
4 (18)	Io	59919.405	1946.1	1625.0	24	3	1716.71
1 (15)	Eu	59900.307	301.9	56.1	25	6	1889.67
2 (16)	Eu	59905.984	300.6	2000.0	24	6	1857.46
3 (17)	Eu	59912.046	1079.7	51.0	22	6	1791.47
4 (18)	Io	59915.857	1847.5	50.0	19	3	1684.30
5 (19)	Io	59918.474	1837.7	400.6	16	3	1645.03
$\#^1_{flyby}$	Moon	Epoch [MJD]	$V_{-\infty}$ [m/s]	h_{flyby} [km]	# Face	Face value	$m_{post\,flyby}$ [kg]
1 (15)	Eu	59877.968	314.7	824.1	25	6	1889.67
2 (16)	Eu	59885.071	312.4	550.3	19	6	1886.45
1 (15)	Eu	59878.033	300.1	1111.6	18	6	1889.67
2 (16)	Eu	59885.131	301.7	188.0	24	6	1886.13
3 (17)	Eu	59890.460	307.5	92.1	25	6	1882.20
1 (15)	Eu	59878.621	685.1	63.2	15	6	1889.67
2 (16)	Eu	59885.185	950.7	230.8	18	6	1850.87
3 (17)	Eu	59890.512	1268.0	55.1	20	6	1810.61
4 (18)	Io	59894.679	2105.5	409.0	24	3	1724.58
1 (15)	Eu	59877.694	300.8	1999.9	15	6	1889.67
2 (16)	Eu	59883.759	478.3	50.1	18	6	1844.65
3 (17)	Eu	59890.828	1107.1	50.1	23	6	1775.43
4 (18)	Io	59894.710	1652.6	50.0	20	3	1659.16
5 (19)	Io	59898.249	1581.2	177.7	25	3	1647.24

[1] Number between parenthesis is the actual flyby number in the CSU solution.

Table D.24: *Results (3x) of subsets for the low-thrust arcs for the first sub-sequence Eu-Eu-Eu-Io-Io. [Settings: full freedom and 2 flyby window and $W_{\Delta V} = 1.0$ and $W_M = 1.0$]*

$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59889.420	5.31	1	7.0	7.0
$f =$	-1.11821			$\Delta V_{tot} =$	7.0	7.0
1 (15-16)	Eu-Eu	59887.855	7.10	1	6.0	6.0
2 (16-17)	Eu-Eu	59894.957	7.10	1	13.0	13.0
$f =$	-1.09296			$\Delta V_{tot} =$	19.0	19.0
1 (15-16)	Eu-Eu	59886.158	6.81	1	67089.0	371.0
2 (16-17)	Eu-Eu	59892.964	4.88	1	295768.0	602.0
3 (17-18)	Eu-Io	59897.844	3.91	1	219065.0	566.0
$f =$	-0.8750052			$\Delta V_{tot} =$	581922.0	1539.0
1 (15-16)	Eu-Eu	59887.513	5.28	1	99241.0	346.0
2 (16-17)	Eu-Eu	59892.795	5.26	1	206079.0	571.0
3 (17-18)	Eu-Io	59898.055	3.78	1	638602.0	1044.0
3 (18-19)	Io-Io	59901.839	2.61	1	92325.0	294.0
$f =$	-0.8000039			$\Delta V_{tot} =$	1036247.0	2255.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59902.729	3.55	1	5.0	5.0
$f =$	-1.16356			$\Delta V_{tot} =$	5.0	5.0
1 (15-16)	Eu-Eu	59901.551	7.10	1	4.0	4.0
2 (16-17)	Eu-Eu	59908.654	3.56	1	16.0	16.0
$f =$	-1.09319			$\Delta V_{tot} =$	20.0	20.0
1 (15-16)	Eu-Eu	59901.410	6.88	1	215.0	215.0
2 (16-17)	Eu-Eu	59908.286	7.02	1	242796.0	698.0
3 (17-18)	Eu-Io	59915.305	4.10	1	507287.0	873.0
$f =$	-0.8750040			$\Delta V_{tot} =$	750298.0	1785.0
1 (15-16)	Eu-Eu	59900.307	5.68	1	46978.0	291.0
2 (16-17)	Eu-Eu	59905.984	6.06	1	325292.0	682.0
3 (17-18)	Eu-Io	59912.046	3.81	1	868467.0	1195.0
3 (18-19)	Io-Io	59915.857	2.62	1	199762.0	429.0
$f =$	-0.8000028			$\Delta V_{tot} =$	1440499.0	2597.0
$\#^1_{leg}$	Moons	Epoch [MJD]	TOF [days]	N_{rev} [-]	ΔV [m/s]	ΔV [m/s] [2]
1 (15-16)	Eu-Eu	59877.968	7.10	1	5.0	5.0
$f =$	-1.17469			$\Delta V_{tot} =$	5.0	5.0
1 (15-16)	Eu-Eu	59878.033	7.10	1	8.0	8.0
2 (16-17)	Eu-Eu	59885.131	5.33	1	12.0	12.0
$f =$	-1.09145			$\Delta V_{tot} =$	20.0	20.0
1 (15-16)	Eu-Eu	59878.621	6.56	1	87677.0	369.0
2 (16-17)	Eu-Eu	59885.185	5.33	1	106762.0	403.0
3 (17-18)	Eu-Io	59890.512	4.17	1	540295.0	929.0
$f =$	-0.8750041			$\Delta V_{tot} =$	734734.0	1701.0
1 (15-16)	Eu-Eu	59877.694	6.07	1	92990.0	445.0
2 (16-17)	Eu-Eu	59883.759	7.07	1	177309.0	709.0
3 (17-18)	Eu-Io	59890.828	3.88	1	948102.0	1301.0
3 (18-19)	Io-Io	59894.710	3.54	1	14585.0	106.0
$f =$	-0.8000032			$\Delta V_{tot} =$	1232986.0	2561.0

[1] Numbers between parenthesis are the actual flyby numbers in the CSU solution and define the departure and arrival point of the leg.
[2] ΔV without the penalties for thrust constraint violation.

Bibliography

Biesbroek, R. (2006). A comparison of the differential evolution method with genetic algorithms for orbit optimisation. In *57th International Astronautical Congress*. American Institute of Aeronautics and Astronautics.

Brest, J., Bošković, B., Greiner, S., Žumer, V., and Maučec, M. S. (2007). Performance comparison of self-adaptive and adaptive differential evolution algorithms. *Soft Computing*, 11(7):617–629.

Brest, J., Greiner, S., Bošković, B., Mernik, M., and Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10:646–657.

Brest, J., Zamuda, A., Bošković, B., Maucec, M. S., and Zumer, V. (2009). Dynamic optimization using self-adaptive differential evolution. In *IEEE congress on evolutionary computation*, volume 2009, pages 415–422.

Casalino, L. (2012). Team 5 gtoc6 report. `http://www.polito.it/gtoc/gtoc6_presentation.pptx`. Accessed: May 31, 2014.

Casalino, L. (2014). 6^{th} global trajectory optimisation competition. `http://areeweb.polito.it/gtoc/gtoc6.html`. Accessed: May 31, 2014.

Colasurdo, G., Zavoli, A., Longo, A., Casalino, L., and Simeoni, F. (2014). Tour of jupiter galilean moons: Winning solution of gtoc6. *Acta Astronautica*, 102:190–199.

Cornelisse, J., Schöyer, H., and Wakker, K. (1979). *Rocket propulsion and spaceflight dynamics*. Number pt. 1 in Aerospace Engineering Series. Pitman.

Das, S. and Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15:4–31.

Elsayed, S. M., Sarker, R. A., and Essam, D. L. (2011). Differential evolution with multiple strategies for solving CEC2011 real-world numerical optimization problems. In *2011 IEEE Congress of Evolutionary Computation, CEC 2011*, pages 1041–1048.

Gämperle, R., Müller, S. D., and Koumoutsakos, P. (2002). A parameter study for differential evolution. *Advances in intelligent systems, fuzzy systems, evolutionary computation*, 10:293–298.

Gijsen, H.-P. (2014). The Tisserand Graph with Low-Thrust - A Graphical Method to Design Multiple Gravity-Assist Trajectories including Low-Thrust Legs. Master's thesis, Delft University of Technology, Delft.

Haines, E. (1994). Point in polygon strategies. *Graphics gems IV*, 994:24–26.

He, S. and Gao, Y. (2014). A new solution scoring 320/324 to the gtoc6 problem. `https://docs.google.com/viewer?a=v&pid=forums&srcid=MDUyNzAyODEzOTU2MTgwNTIyMzQBMDcwOTYyNzUwNTUyODQ5NTAwNjQBVWV5N2U5SzU1MzRKATAuNAEBdjI&authuser=0`. Accessed: Februari 9, 2015.

Hoving, L. (2014). Tour of the Galilean Moons - A Literature Survey of Low-Thrust Multiple Gravity-Assist Trajectory Design for GTOC6. Master's thesis, Delft University of Technology, Delft.

Izzo, D., Ruciński, M., and Ampatzis, C. (2009). Parallel global optimisation meta-heuristics using an asynchronous island-model. In *2009 IEEE Congress on Evolutionary Computation, CEC 2009*, pages 2301–2308.

Izzo, D., Simões, L. F., Märtens, M., de Croon, G. C. H. E., Heritier, A., and Yam, C. H. (2013). Search for a Grand Tour of the Jupiter Galilean Moons. In *Genetic and Evolutionary Computation Conference (GECCO 2013)*, pages 1301–1308.

Mambrini, A. and Izzo, D. (2014). Pade: A parallel algorithm based on the moea/d framework and the island model. In *Parallel Problem Solving from Nature–PPSN XIII*, pages 711–720. Springer.

McFadden, L.-A., Weissman, P., and Johnson, T. (2006). *Encyclopedia of the solar system*. Academic press.

Morris, R. (2007). Intersection between a line and plane. `https://en.wikipedia.org/wiki/File:Line_plane.svg`. Accessed: October 20, 2015.

Musegaas, P. (2012). Optimsation of space trajectories including multiple gravity assist and deep space maneuvers. Master's thesis, Delft University of Technology, Delft.

Novak, D. and Vasile, M. (2011). Improved shaping approach to the preliminary design of low-thrust trajectories. *Journal of Guidance, Control and Dynamics*, 34(1):128–147.

Novak, D. M. (2012). *Methods and tools for preliminary low thrust mission analysis*. PhD thesis, University of Glasgow.

Park, R. S. and Chamberlin, A. B. (2015). Jpl small-body database browser - nasa. `http://ssd.jpl.nasa.gov/sbdb.cgi`. Accessed: Februari 20, 2015.

Petropoulos, A. E. (2012). Problem description for the 6^{th} global trajectory optimisation competition. `http://sophia.estec.esa.int/gtoc_portal/wp-content/uploads/2012/11/gtoc6_problem_stmt-2.pdf`. Accessed: December 9, 2013.

Price, K. V. (1996). Differential evolution: a fast and simple numerical optimizer. *Fuzzy Information Processing Society, 1996. NAFIPS. 1996 Biennial Conference of the North American*, pages 524–527.

Roegiers, T. (2014). Application of the spherical shaping method to a multiple asteroid rendezvous mission (GTOC2 problem). Master's thesis, Delft University of Technology, Delft.

Storn, R. and Price, K. (1997). Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359.

Vasile, M. and Pascale, P. D. (2006). Preliminary design of multiple gravity-assist trajectories. *Journal of Spacecraft and Rockets*, 43(4):794–805.

Vroom, J. (2015). On the out-of-plane component of shaping methods. Master's thesis, Delft University of Technology, Delft.

Walker, R. J. and Snoeyink, J. (1999). Practical point-in-polygon tests using csg representations of polygons. In *Algorithm Engineering and Experimentation*, pages 114–128. Springer.

Ward, G. and Heckbert, P. (1994). Graphics gems iv. *Academic Press Professional, Inc., San Diego, CA, USA, ch. A contrast-based scalefactor for luminance display*, 415:421.

Yam, C. (2012). Global trajectory optimisation competition esa-hkust. `http://www.ust.hk/eng/news/photos/20121210-1004-gtoc6_act_hkust_hippo_v3.pdf`. Accessed: December 11, 2013.