Global Optimisation of Interplanetary Trajectories

by Hattie Hall

A Thesis in submission for the award of Master of Philosophy

University of Southampton

School of Mathematical Sciences

August 7, 2022

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

SCHOOL OF MATHEMATICAL SCIENCES

Mathematics

Master of Philosophy

GLOBAL OPTIMISATION OF INTERPLANETARY TRAJECTORIES

by Hattie Hall

This thesis introduces and explores the full global interplanetary trajectory optimisation problem.

The biggest challenges in this field are expensive objective function evaluations, the size and multimodality of the search space, a requirement for good initial solutions to initialise search algorithms, the need for manual input and separate solutions to solve the combinatorial and continuous elements of the problem and finally solution robustness.

The literature is summarised, analysing current solution methods, global algorithms, software and toolboxes with respect to the challenges identified.

It is concluded that Monte Carlo Tree Search and hybrid evolutionary algorithms are perhaps the most effective algorithms currently in use. Though techniques used for search space reduction and approximation (that are algorithm agnostic) can have just as large an impact.

Opportunities for further work into algorithm parameter optimisation, machine learning for search space reduction and extended objective function approximation are outlined.

Contents

List of Figures

Research Thesis: Declaration of Authorship

Name: Hattie Hall

Title: Global optimisation of Interplanetary Trajectories

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research. I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. None of this work has been published before submission;

Signature:

Chapter 1

Introduction and Objectives

The purpose of this thesis is to summarise and analyse the existing literature related to global optimisation problems for interplanetary trajectories. This thesis will identify challenges, summarise progress and make recommendations for future work to further progress the field.

There are many types of global optimisation problem, typically characterised by the variable types (integer, continuous, binary) and features of the objective function(s)/search space (linear, non-linear, convex, non-differentiable, for example). Global optimisation problems are a more challenging problem than local optimisation; due to the need to distinguish the global optimum from among the possibly many local optima.

Interplanetary trajectory optimisation is widely regarded to be one of the most complex optimisation problems there are. The search space is highly non-linear and non-differentiable; the objective function is a complicated function containing elements from a Lambert-solver and ephemeris calculations, that cannot be written down into one elegant and easily solvable equation.

The complexity of this problem is adequate motivation for this thesis; Any progress in understanding or methodology for one of the hardest definable optimisation problems today, is sure to yield benefit for future spacecraft missions and across a much wider range of optimisation problems.

Chapter 2

Background and Context

2.1 Interplanetary Travel

Interplanetary travel is the relocation of a spacecraft or probe (that may or may not contain living beings) between two different planets. This could include planets in other solar systems, however this is impossible with current engineering, technologies and timescales. In the context of today, interplanetary travel refers to travel between two planets within our solar system- usually travelling from Earth on a one-way trip, though return missions are desirable and beginning to emerge.

Space and our solar system has fascinated people for hundreds of years; Many planets, moons, comets and asteroids were observed from the ground using simple telescopes. Astonishingly accurate calculations were made of their orbits and trajectories without ever sending a probe beyond our atmosphere. Humans have always been curious of our environment and origins, and this naturally extends to a desire to explore the solar system in a search for answers.

The first successful Earth orbiter was launched in 1957 by the Soviet Union of the name Sputnik 1. It was in orbit for only 3 months, but observing its orbital path and radio wave propagation enabled scientists to better understand the upper atmosphere and ionosphere of the Earth.

It wasn't long before a spacecraft was launched to the moon. The first successful flyby, launched in 1959 by the Soviet Union, was completed by Luna 1. There have been many spacecraft both landing on and in orbit around the Moon, including the famous Apollo missions which saw man travelling safely to and from the moon. As our nearest celestial object, the moon is a great place to start our exploration of the solar system and to test technologies that may be useful for interplanetary missions.

On August 27th 1962, NASA launched Mariner 2, the first spacecraft to successfully flyby Venus. Launcher capabilities meant that the probe had only a few scientific instruments on board but the mission was successful in measuring the extremely hot surface temperature of Venus and much cooler clouds higher in its atmosphere. Other observations were made during its journey to Venus including Solar wind, solar flares and interplanetary dust. Later missions revealed high wind speeds and surface pressure on the planet (NASA's Mariner 5), atmospheric composition (Venera 4) and even transmitted photographs of the Venusian surface back to Earth (Venera 9/10).

Until 1964, all Spacecraft had been using chemical (high thrust) propulsion systems as standard but, in 1964 the first low-thrust ion engine was tested on board SERT-1 (Space Electric Rocket Test). The engine only operated successfully for around 30 minutes, but this proof of concept was enough to continue research and testing and eventually launch craft relying solely on low thrust propulsion.

NASA's Mariner 4 craft was launched successfully to the vicinity of Mars in November 1964 after numerous failed attempts from NASA and the Soviet Union. Many subsequent missions have taken place (successes and failures), including many landers and rovers that transmit information from the surface back to the Earth. One of the primary motivations for visiting Mars was, and still is, the search for life. It was suggested that the climate and geography of the planet may have supported life at some time in the past, though no concrete evidence has been found of this yet. Exploration here has also found the highest mountain in the solar system, the Olympus Moons and evidence of past surface water.

Early missions to the Moon, Venus and Mars offered opportunity to test technologies and gain understanding of the dynamics of Space and how to control a craft, probe or lander in these sometimes hostile environments. For example, approximately 2 in 3 of all the missions to Mars failed, but something was gained from each attempt even if it was small and not necessarily relevant to the original mission aim; Luna 1 sailed past the moon, contrary to the original plan, and was able to return readings of solar wind and measurements from the Earth's radiation belt.

Confidence grew in interplanetary navigation and spacecraft engineering and in 1972 and 1973, respectively, spacecraft Pioneer 10 and 11 reached the vicinities of Jupiter and Saturn. During their missions they identified new Saturnian moons, transmitted important photographs of the planet's rings and measured the strong electromagnetic field of Jupiter. Pioneer 11 also passed through the rings of Jupiter and determined Titan to be too cold to support life.

1973 also saw NASA launch Mariner 10 to Mercury, making use of a gravity assist at Venus in order to bend the craft's trajectory into an orbit that frequently passed Mercury in close proximity. This was the first use of a gravity assist (which will be described in more detail later on) in an interplanetary mission and it's success led the way for many more missions to the outer planets of the solar system, that were previously thought unreachable with the available launch speeds. Most notably, NASA's Voyager 1 and 2 craft used a series of gravity assists at Jupiter and Saturn to enable fly-by's at the previously unvisited Neptune and Uranus. Voyager 1 is currently in interstellar space, more than 135AU from the sun and still communicating with the deep space network to transmit basic data.

The Soviet craft Vega 1 and American Galileo achieved the first successful fly-by's of a comet and an asteroid, in 1984 and 1989 respectively. Vega used a gravity assist at Venus to intercept Halley's comet and return images, dimensions and composition data that would help scientists to understand more about these objects and their origins.

The last planet of the solar system to be visited by an Earth-born spacecraft was Pluto, which was visited by NASA's New Horizons Spacecraft launched in 2006. Making use of a flyby at Jupiter to get there and having completed its flyby of Pluto, it is now on course to visit Kulper belt object 2014MU69 in 2019.

As technologies have advanced, space travel has become easier. Improved reliability means fewer failed missions and lost craft. With this new confidence in our abilities to navigate interplanetary space and complete missions successfully, ambitions have grown. Missions began as simple fly-by's of our nearest neighbours Mars and Venus but now incorporate complex trajectories with multiple fly-bys in order to observe moon systems far more complex than our own- Such as the Cassini-Huygens mission launched in 1997 to explore the Saturnian moon system and the planned JUICE mission (to launch in 2022) to explore the Jovian Icy Moons.

Plans are also underway for an asteroid sample return mission, namely OSIRIS-REx led by NASA to Bennu. Bennu is a carbon rich asteroid that could provide the key to answering questions of the origins of human life and the early solar system. The chemical composition of the asteroid may also be useful information for future asteroid deflection missions that put the Earth at risk. The OSIRIS-REx craft will use chemically propelled deep space manoeuvres, and a gravity assist from Earth in order to reach the asteroid.

Further to the human involvement in Space missions seen in the Apollo missions and astronauts aboard the International Space Station, are manned missions to the moon, an asteroid and Mars- which are all currently being planned. The European Space Agency are actively testing the use of 3D printing to construct a lunar base, whilst NASA are planning a manned mission to Mars in 2037 with a manned training mission to an asteroid placed in Lunar orbit in the 2020's.

Another interesting advancement in the Space Industry has been the addition of independent, or private, space agencies. Previously, Space has only been accessible by publicly funded or government controlled organisations such as NASA, ESA, JAXA etc, with the main aims and objectives of the missions being Scientific discovery and exploration. However companies such as Virgin Galactic and SpaceX have emerged; their motives are less the scientific advancements gained from planetary observations, and more the commercial opportunities found within Space travel. Virgin Galactic in particular, aim to make space travel more accessible to the general public, and Space X work specifically on the design of spacecraft and rockets with the ultimate goal of enabling humans to live on other planets.

Space exploration satisfies a very human need to explore our surroundings and understand our origins. It offers the opportunity for collaboration across the globe and many challenges presented by the hostile environments and conditions the spacecraft must endure. These challenges encourage technological advancements in many fields and often result in spin-off technologies such as:

- Material development- material composites are developed for very specific properties, and these often have an interesting use elsewhere. For example, translucent polycrystalline alumina- TPA has been used both as infrared protection for antennae and invisible braces.
- Anti-Icing systems- for use on spacecraft or aircraft.
- Water Purification systems- to aid astronauts away from Earth or people unable to obtain fresh water.
- Solar cells-to power spacecraft instruments and for use on Earth as a sustainable alternative to burning fossil fuels.
- Magnetic Resonance Imaging- For planetary analysis or medical scanning.
- Mathematical modelling and optimisation methods- for optimising the efficiency of all aspects of a mission, with many methods, techniques and resulting algorithms able to be applied in many other situations.

There is some opposition to the investments in space exploration and particularly interplanetary exploration. It is said that the amount of money spent on space exploration could be put to much better use solving the existing problems on Earth. How can humans be expected to travel to other planets, and make use of the resources sustainably, if they cannot already do this on Earth? Those against investment in Space research claim the technological advancements, knowledge and understanding of the wider environment gained, and the possibilities for the future of the human race are not enough to justify the spending when problems such as climate change and deforestation are so imminent here.

It is clear that interplanetary travel will continue to take place regardless of motive, and will ultimately bring benefit to humanity and additional scientific advancement in many fields. As interplanetary missions continue, their aims and requirements become more complex and technologies continue to advance, optimisation and mathematical modelling will (and already does) play a vital role in assessing mission feasibility, keeping costs down, keeping mission times minimal, and maximising the return on investment.

2.2 Components of Interplanetary Trajectory Design

In order to explain the process of interplanetary trajectory design, it is necessary to introduce the constituent parts of an interplanetary trajectory, the astrodynamics supporting them and some of the mathematics used to aid the design process.

2.2.1 Astrodynamics

Astrodynamics is the science of dynamic motion and gravitation of natural and man-made objects in Space.

Before interplanetary trajectory optimisation can be explored in its full form, an understanding of some definitions and basic principles of astrodynamics is required

An *orbit* is the regularly repeated path of a celestial object or spacecraft about some massive body (such as a star, planet or moon). This kind of orbit is sometimes referred to as a *closed* orbit.

A *Trajectory* is the path followed by a projectile or an object moving under the action of given forces. For a spacecraft, these forces are those due to the gravitational fields of the stars, planets and moons nearby, and the forces created by the on-board propulsion systems. A trajectory will not necessarily repeat itself, and may have no easily definable or describable shape.

2.2.1.1 Kepler's and Newton's Laws

At the beginning of the 17th century, Kepler published three laws of planetary motion [6] that ultimately define the orbits of any object experiencing gravitational effects from just one body:

Kepler's First Law:

"The orbit of a planet is an ellipse with the Sun at one focus."

This leads to the following equations for the aphelion and perihelion radii (R_a, R_p) respectively. Where a is the semi major axis and e is the eccentricity:

$$R_a = a(1 + e)$$

$$R_p = a(1 - e) \tag{2.1}$$

Kepler's Second Law:

"A line segment joining a planet and the Sun sweeps out equal areas during equal intervals of time."

This law is derived from the conservation of angular momentum (L) within orbit.

$$L = mv_1r_1 = mv_2r_2$$

Where m is the mass of the orbiting object, r_1, r_2 are the respective orbital radii at two points on an orbit and v_1, v_2 are the velocities of the spacecraft at those same points.

Kepler's Third Law:

"The square of the orbital period (T) of a planet is proportional to the cube of the semi-major axis of its orbit (a)."

$$T^2 = \frac{4/pi^2}{GM}a^3$$

These laws were discovered from hundreds of planetary observations from the Earth in a time where putting manmade objects into Space seemed impossible. It later turned out that the laws could be generalised to describe anything in orbit in a similar environment; that is to say that Kepler's laws can be used to describe the orbit/motion of a spacecraft provided its mass is negligible in comparison to the body it is orbiting, and it does not experience the gravitational effects of any other celestial body.

Whilst Kepler's laws explain the kinematics of orbital space motion well, they do not offer any explanation of the dynamics of the system that lead to the elliptic motion. This explanation was offered by Isaac Newton [7] later in the 17th century :

Newton's First Law:

Every body continues in its state of rest, or of uniform motion in a straight line, unless it is compelled to change that state by forces applied to it.

Newton's Second Law:

The change of motion is proportional to the motive force applied and is made in the direction of the straight line in which that force is applied.

Newton's Third Law:

To every action there is always an equal and opposite reaction; or, the mutual actions of two bodies upon each other are always equal and opposite.

Newton's Universal Law of Gravitation:

A particle attracts every other particle in the universe using a force F that is directly proportional to the product of their masses (m_1, m_2) and inversely proportional to the square of the distance between them r. Where G is the gravitational constant, the law forms the following equation:

$$F = \frac{Gm_1m_2}{r^2}$$

2.2.1.2 Orbits and Conic Sections

Kepler's first law describes orbital motion to be elliptic in shape, however this is only one particular case of the motions allowed in Space. In fact, all conic sections satisfy Newton's laws of gravitation and hence the circle, ellipse, parabola and hyperbola all describe valid orbits and trajectories for objects in space.

The four conic sections can be defined as the intersections of a plane with a right cone and are demonstrated in figure 2.1.

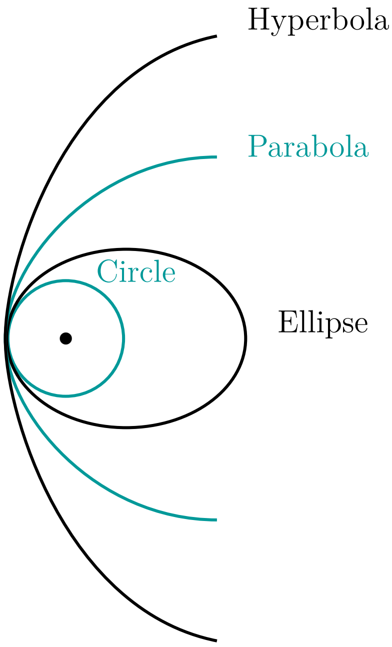


Figure 2.1: Demonstrating the four conic sections that describe motion in space.

Circular and Elliptic Orbits:

The most common type of orbit is an ellipse, shown in figure 2.2 this is a closed orbit that repeats itself with some period. An ellipse is described geometrically using the semi-major axis a (half of the width of the ellipse), and the semi-minor axis b (half of the height of the ellipse). It has two focus points(F_1 and F_2). A defining feature of the ellipse is that for any point P on its perimeter, the sum of its distance from the two focus points $(r_1 + r_2)$ is constant.

A special type of ellipse is a circle, where $a = b$ and consequently the two focus points are both located at the centre of the circle.

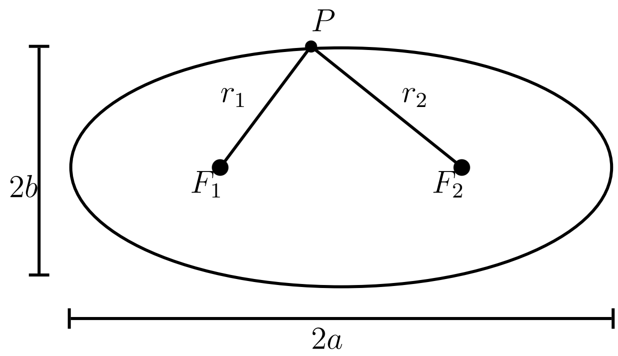


Figure 2.2: An ellipse with marked parameters.

A spacecraft in an elliptical orbit around some massive celestial body orbits with the body at one focus. This is known as the primary focus of the ellipse.

Hyperbolic and Parabolic Orbits:

A parabola is a conic section with eccentricity $e = 1$. It represents a non-periodic, open orbit around a celestial body at one focus. An open orbit occurs when the energy of the spacecraft is sufficiently large to escape the gravity well of the orbited body. A parabolic trajectory is the minimum escape trajectory.

Similarly, hyperbolic trajectories are defined by an eccentricity $e > 1$. These trajectories are higher energy. Parabolic and hyperbolic trajectories describe flyby and escape trajectories.

2.2.1.3 Orbital Elements

In general, it takes at least six parameters to uniquely define an orbit and the location of a spacecraft on the orbit. These elements are as follows:

- Semi-Major Axis a- This defines the size of the orbit and it's value also describes the nature of the orbit; $a < 0$ is an hyperbolic trajectory, $a > 0$ is an elliptical orbit.

- Eccentricity e- Eccentricity defines the shape of the orbit. A circle has eccentricity 0, whereas a long, thin ellipse with a $a >> b$ would have an eccentricity closer to 1. An orbit with $e = 1$ is parabolic, and $e > 1$ describes an hyperbolic trajectory.

- Inclination i- Inclination defines the orientation, as an angle, of the orbit with respect to a plane of reference of the orbited body (usually the equator). For example, an orbit in the same plane as the equator would have an inclination $i = 0$, and an orbit in the same plane as both the North and South poles would have an inclination of $i = \frac{\pi}{2}$.

- Right Ascension of the Ascending Node Ω- This gives the location of the ascending node. Given a reference direction, within the reference frame, Ω is the angle from this direction to the ascending intersection of the orbit with the reference plane (Otherwise known as the ascending node).

- Argument of Periapsis ω- Defines the location of periapsis of the orbit, with respect to the surface of the planetary body being orbited. Periapsis is the point of the orbit at which the spacecraft is closest to the central body. ω is the angle between the ascending node and the the point of periapsis.

- True Anomaly v- Measured from periapsis, this defines the location of the satellite on the orbit as an angle.

If the location of the spacecraft, or the orientation of the orbit is not required only 2 variables are needed to specify the size and shape of the ellipse. The choice for these 2 variables is not limited to the 6 outlined above.

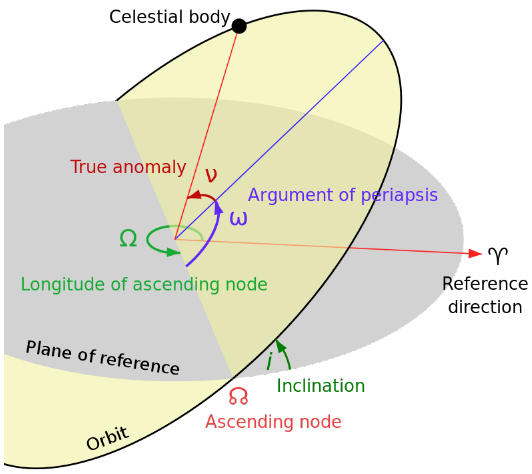


Figure 2.3: Demonstrating the orbital parameters: Figure taken from [1].

An additional element that will be used later in the thesis is the Eccentric Anomaly E. This is the angle obtained by drawing the auxiliary circle of an ellipse with centre O and focus F, and drawing a line perpendicular to the semi-major axis and intersecting it at A. This is represented graphically below.

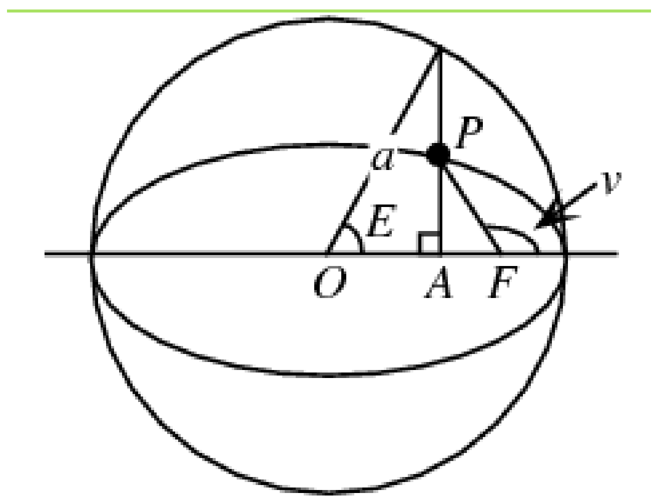


Figure 2.4: Demonstrating the calculation of the Eccentric Anomaly [2].

2.2.1.4 Orbital Equations

The following equations explain the motion of a spacecraft in orbit. Where:

$\mu = GM$

G is the gravitational Constant,

M is the mass of the orbited planet,

r is the radius of orbit above the centre of the orbited planet.

The Vis-Viva Equation:

$$v_e = \sqrt{\mu(\frac{2}{r} - \frac{1}{a})} \tag{2.2}$$

2.2 defines the orbital speed of a body on an *elliptic* orbit, where μ is the gravitational parameter. The equation is often referred to as the orbital energy conservation equation as it describes the conservation of total energy in the system.

$$v_h = \sqrt{\mu(\frac{2}{r} + \frac{1}{a})} \tag{2.3}$$

2.3 defines the orbital speed of a body on an *hyperbolic* trajectory.

Specific Orbital Energy

$$\frac{v^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a} = \epsilon < 0 \tag{2.4}$$

Rearranging 2.3 and 2.2 produces the specific orbital energy ϵ in 2.4. Specific orbital energy is used to calculate the required escape velocity v_e. This is achieved by setting $\epsilon = 0$.

Escape Velocity:

$$v_e = \sqrt{\frac{2\mu}{r}} \tag{2.5}$$

The escape velocity in equation 2.5 is the velocity required in order to break free from an elliptic orbit.

2.2.1.5 Sphere of Influence

The *Sphere of Influence* (SOI) of a celestial body is the spherical region surrounding it in which the motion of an object inside it is affected by that body's gravitational field. This behaviour is best defined using Newton's Law of Gravitation:

$$F = \frac{Gm_1m_2}{r^2}$$

Where F is the gravitational force acting between objects 1 and 2, with associated masses m_1 and m_2, and a distance of r between them. G is the gravitational constant.

The sphere of influence can be considered as any region in which $F > 0$. For $m_1, m_2 > 0$ this means the sphere of influence is of infinite radius as $F > 0 \; \forall r > 0$. It is important to note that as $r \to \infty$, $F \to 0$ which delivers the assumption that for some large enough radius r, F will be negligibly small, thus creating a finite sphere of influence.

For the purposes of ease of modelling, the following definition will be used for sphere of influence:
The *Sphere of Influence* (SOI) of a celestial body is the spherical region surrounding it in which the motion of an object inside it is affected by that body's gravitational field more than any other.

A planet's SOI is an important concept in interplanetary trajectory design as it allows the models used for the motion of a spacecraft to be simplified. Considering only the orbiting and orbited body is often referred to as the *2 body problem*. It is the task of determining the motion of two body's interacting only with each other. For example, the Earth's sphere of influence is around 924,000 km in radius. Within this region, the Earth's gravity is dominant and movement of an object in this region will largely be described by the Earth's gravity. However, outside this region, the Sun's gravity is dominant and so the motion of an object will be governed predominantly by the Sun's gravitations forces.

2.2.2 Spacecraft Propulsion

2.2.2.1 High thrust

Chemical Propulsion systems use heat energy produced by a chemical reaction to generate thrust by expelling gas at great temperature and pressure through a small nozzle; working in accordance with Newton's third law of motion. They can produce large changes in velocity in a relatively small time period. For example, a deep space manoeuvre (DSM) performed by the Juno spacecraft on October 9th 2013 lasted 30 minutes, burned 376kg of fuel and increased the velocity of the spacecraft by roughly 388 meters per second[8].

Chemical Propulsion systems rely on the availability of the necessary chemicals to produce the thrust, which is constrained by the amount that can be feasibly stored on board the spacecraft. The main problem with carrying fuel into space, is that it requires fuel to do so; fuel requirement increases exponentially with the mass of the craft, so minimising the need for fuel in a mission helps to keep costs down.

2.2.2.2 Modelling High Thrust propulsion

High thrust, chemical propulsion is typically modelled as an instantaneous change in velocity at some position on a trajectory (often referred to as a deep space manoeuvre (DSM) in the literature). The position is usually fixed, often halfway between two planets or in the vicinity of a GAM.

Although allowing DSM's improves the quality of solutions to the problem, the position is usually fixed as it also increases the complexity of the problem; allowing DSM's at any point on a trajectory would likely enable some good trajectories to be found, however it would also require an additional variable and significantly increased computation time.

The size of the thrust itself is not typically modelled, rather inferred as the difference in spacecraft energy just before and just after the burn. They are constrained to feasibility within the optimisation problem and provide a means to smooth (or patch) discontinuities when connecting elliptic and hyperbolic orbits.

2.2.2.3 Low Thrust

Electric, or low thrust, propulsion works similarly to chemical propulsion, but instead of a chemical reaction, uses electrical energy (produced from some external power source such as solar radiation or electromagnetism) to eject matter at high velocity through a small nozzle to produce thrust.

Over the same time period, electric propulsion systems produce much smaller changes in velocity than chemical propulsion. But they can thrust for much longer. This capability enables spiral trajectories, and other transfer orbits to be achieved.

Electric propulsion systems depend on access to some energy source, such as solar radiation, which may be affected by the spacecrafts distance from the sun. Low thrust propulsion offers a massive benefit where fuel is concerned. The lack of fuel on-board the craft reduces the weight of the craft and allows for a faster launch speed, or larger scientific payload.

2.2.2.4 Modelling Low Thrust Propulsion

Low thrust propulsion is typically more difficult to model than chemical propulsion.

Due to the constant force applied using low thrust propulsion, its affect can not be modelled using just one instantaneous energy change. The Sims-Flanagan transcription method [9] used a series of instantaneous energy changes, connected by conic arcs, to model the propulsion. The method was widely used, but it fails to accurately represent the true dynamics of low thrust propulsion. Instead, a shape-based approach was adopted. Low thrust propulsion systems enable spiralling trajectories and exponential sinusoids were identified as the best fit to model this behaviour[10, 11].

In polar coordinates (r, θ), an exponential sinusoid is given as follows:

$$r = k_0 e^{k_1 \sin(k_2\theta + \phi)}$$

Where k_0, k_1, k_2 and ϕ are constants that control the shape of the function.

k_1 defines the ratio between the apoapsis and periapsis distances. k_2 defines the winding behaviour. k_0 is a scaling factor and ϕ defines the orientation angle for the function in the plane.

Assuming that thrust is tangential (either along or against the velocity vector), the two body equations of motion are:

$$\ddot{r} - r\dot{\theta}^2 + \frac{\mu}{r^2} = F\sin\alpha$$
$$\frac{1}{r}\frac{d}{dt}(r^2\dot{\theta}) = F\cos\alpha$$

Where μ denotes the gravitational parameter of the central body being orbited, $F = [F\cos\alpha, F\sin\alpha]$ is the magnitude of the thrust and α is the thrust angle measured clockwise from the axis perpendicular to r. Where the flight path angle γ is equivalent to the thrust steering angle α, the control acceleration is given by:

$$F = \frac{\mu}{r^2}\frac{tan\gamma}{2\cos\gamma}\left(\frac{1}{\tan^2\gamma + k_1k_2^2s + 1} - \frac{k_2^2(1-2k_1s)}{(tan^2\gamma + k_1k_2^2s + 1)^2}\right)$$

Where $s = \sin(k_2\theta + \phi)$. The time variation of the true anomaly is:

$$\dot{\theta}^2 = \frac{\mu}{r^3}\frac{1}{\tan^2\gamma + k_1k_2^2s + 1}$$

The flight path angle γ is:

$$\tan\gamma = k_1k_2\cos(k_2\theta + \phi)$$

Hence the final time of flight is the solution to the following integral:

$$\Delta t = \int \frac{d\theta}{\sqrt{\left(\frac{\mu}{r^3}\right)\frac{1}{\tan^2\gamma + k_1k_2^2s + 1}}}$$

Values for k_0, k_1 and k_2 can be shown to be dependent on one another (see [11], meaning only one variable k_2 is free and used to target the next planet for flyby or to be captured into orbit.

The start and length of the low thrust burn may be free or fixed, but as with the chemical propulsion models, this will increase the complexity of the model.

2.2.2.5 Other Propulsion Methods

There are a few other Spacecraft propulsion methods currently being tested, that do not depend on an internal reaction mass. The most notable is perhaps the solar sail, which relies on the solar radiation pressure to accelerate the craft, much like wind on a sail boat. However a large sail (in excess of 0.5km wide) would be required for any useful propulsive effects to be noted.

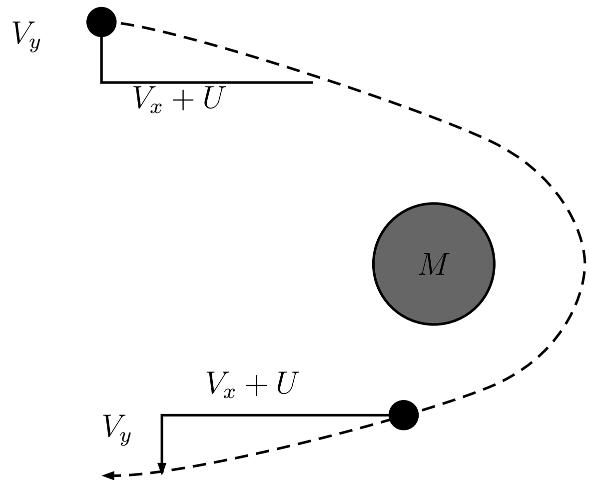
The concept has been proven in a number of trial missions, notably Japan's interplanetary mission IKAROS [12] which launched to Venus in May 2010. It completed the flyby successfully and provided many useful readings and observations. Since this proof of concept, more missions have been planned to test the technology in the hope that it may one day provide a cheap, lightweight alternative to the chemical and electric propulsion systems used today.

2.2.3 Gravity Assist Manoeuvres

A gravity assist manoeuvre (GAM or flyby) is the change in path of a spacecraft using the relative orbital motion and gravity of a more massive celestial body. They are particularly useful for accelerating a spacecraft, curving a trajectory, reducing the overall length of a mission and the amount of propellant required. Using GAM's it is possible to reach parts of the solar system initially thought impossible due to limits on the capabilities of the launcher, fuel capacity and mission duration. The technique was notably used as part of the Voyager missions in the 1970's where flyby's of Jupiter, Saturn, Uranus and Neptune accelerated the spacecraft to the outer reaches of our solar system [13].

The manoeuvre is somewhat counter-intuitive due to the conservative nature of momentum and gravity. Considering the manoeuvre in the planetocentric frame of reference, the initial velocity of the spacecraft is conserved providing a departing velocity from the planet of equal magnitude. The relative speed of the spacecraft upon approach and departure from the planet are equal due to the conservation of momentum and the equal gravitational forces acting at arrival and departure. This can be seen in figure 2.5 simplified to only two dimensions as viewed in the orbital plane.

Figure 2.5: Demonstrating a Gravity Assist Manoeuvre in the Planetocentric frame of reference in the orbital plane.

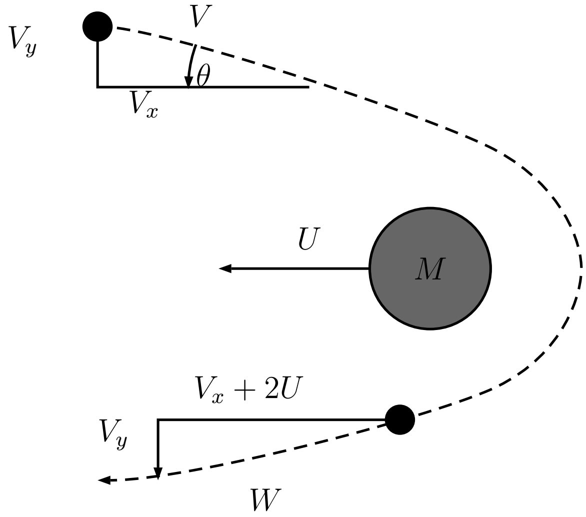


As the spacecraft approaches the planet it begins to accelerate due to the Planet's gravitational field. However, the spacecraft is not captured into an elliptic orbit as its initial velocity V is too large and specifies an hyperbolic trajectory past it. As the spacecraft departs the planet, it decelerates with respect to the planet due to the gravitational field once again.

The true effect of the gravity assist is best seen in the heliocentric reference frame. Because the planet is of much greater mass than the spacecraft, and is moving (unlike in the planetocentric frame, where it is considered stationary), conservation of momentum dictates a change in velocity of the spacecraft (and theoretically, in the planet though the ratio of spacecraft/planet mass is so small that this change is insignificant and negligible).

The spacecraft gains a change in velocity of approximately twice the velocity of the planet in this frame due to the relative speed of their approach, $V_x + U$ (in the x axis), and the additional acceleration provided by the gravitational forces inside the sphere of influence of the moving planet at velocity U. This provides a new x-component of velocity of $V_x + 2U$. This is demonstrated in figure 2.6.

Figure 2.6: Demonstrating a Gravity Assist Manoeuvre in the Heliocentric frame of reference in the orbital plane.



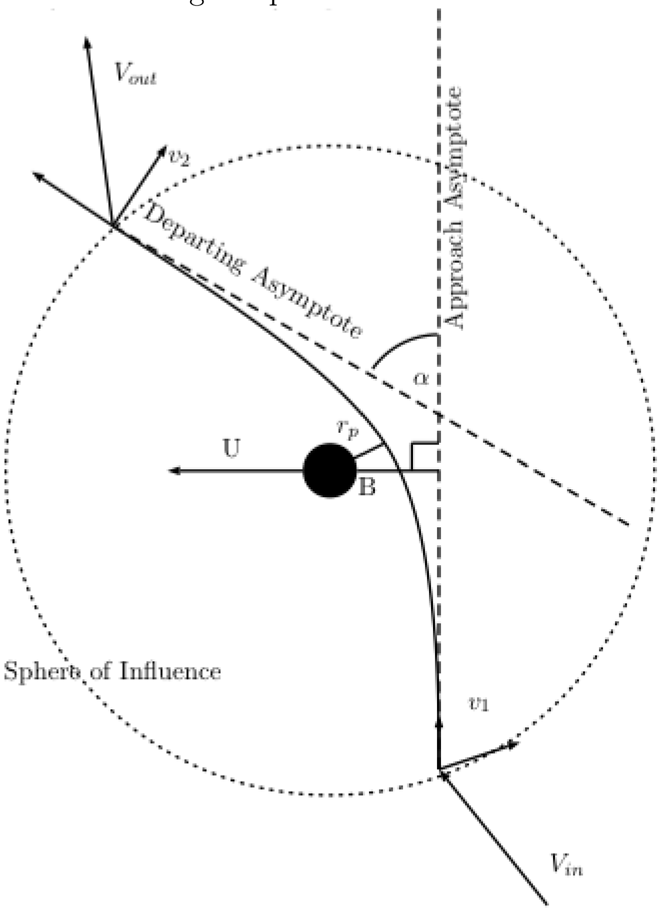
The path change, or angle of deflection from the initial path is the result of the hyperbolic trajectory of the spacecraft.

It is also possible to use a GAM to decelerate a spacecraft, by approaching the planet from behind and intersecting its orbit just *before* it gets there itself (the opposite of the GAM for acceleration technique just described). This technique could be particularly useful when travelling to a distant, small moon or planet, that will require a much slower speed for a successful orbit insertion than the fast speed required to travel there quickly.

2.2.3.1 Modelling Gravity Assist Manoeuvres

A concise mathematical model for a GAM is given in [14] and provides some explanation of the geometry of the event. Figure 2.7 shows the manoeuvre.

Figure 2.7: Demonstrating the parameters used to describe a Gravity Assist



A spacecraft approaches the sphere of influence of a planet with velocity V_{in}. An impulsive manoeuvre on the boundary of the sphere of influence accelerating the craft in the direction of v_1 aligns the craft for a hyperbolic trajectory past the planet with pericentre radius r_p. The rotational velocity change α is calculated with the help of the approach and departing asymptotes and B (the perpendicular distance between the planet and the approaching asymptote).

$$\sin\frac{\alpha}{2} = \frac{\mu}{\mu + r_p v_\infty^2}$$

$$\Delta V = 2v_\infty \sin\frac{\alpha}{2}$$

2.2.3.2 Modelling Powered Fly-bys

A powered flyby is almost identical to a ballistic/unpowered flyby. A powered flyby still takes advantage of the gain in kinetic energy, but additionally, takes advantage of a well timed DSM to experience an even larger gain in kinetic energy.

The spacecraft is at its largest velocity at periapsis, making it the most efficient point to thrust and gain kinetic energy. Further, if the spacecraft where in a parabolic orbit (as opposed to the usual hyperbolic), then the velocity at periapsis would be equivalent to the escape velocity for the planet, and this would be the optimal point to thrust. In some cases, it is even optimal to use an additional thrust in order to lower an hyperbolic orbit into a parabolic one, so that this effect may be fully taken advantage of [15]. This is known as the Oberth effect [16].

Mathematically modelling a powered flyby is a combination of DSM thrust points and a GAM. One thrust point is positioned at periapsis, allowing for an instantaneous change in velocity. Another, optional thrust point is placed upon entrance to the SOI (to position into the desired orbit around the planet). The spacecraft arrives and departs the planet's SOI on two different trajectories.

For a parabolic orbit of the planet, the departing velocity V of the spacecraft in the planetocentric frame of reference is calculated as:

$$V = \Delta v \sqrt{1 + \frac{2V_p}{\Delta v}}$$

Where Δv is the impulse provided by thrust and V_p is the velocity at periapsis (the escape velocity) before the thrusters initiated the change in velocity.

Similarly to the ballistic GAM, the departing velocity in the heliocentric frame of reference would be further enhanced by a component of the planet's velocity.

Chapter 3

Interplanetary Trajectories and the Two-Body Problem

3.1 Two-body Problem

In classical celestial mechanics, the two-body problem is to describe the motion of two massive, celestial objects. The bodies are assumed to be only within each others sphere of influence, so no other external forces act on the bodies.

Perhaps the most relevant application for this work is known as Kepler's Problem. This is a specific case of the two-body problem whereby the central force F between the two bodies varies in strength as the inverse square of the distance between them. This was noted earlier in section 2.2.1.5. Other examples include the determination of the orbit of a satellite moving about a planet or sun, a planet around its sun or the seemingly more complex interaction of binary stars orbiting around one another.

The two-body problem is solved using Kepler's laws, identifying the motion of the bodies as orbits described using orbital elements.

An in depth discussion of the problem and its solution can be found in [17] in addition to many other texts on celestial mechanics.

3.2 Hohmann Transfers

A Hohmann transfer (first introduced in [18]) is one of the most fuel efficient ways to transfer from one circular orbit to another. It is an orbital manoeuvre often employed in orbit raising (moving a spacecraft or satellite from a lower orbit to one of higher altitude), but can also be used for interplanetary transfers. The transfer trajectory is half of an elliptical orbit around the Sun. Two burns are required for the transfer, the first takes place upon departure of the initial planet's sphere of influence in order to acquire the necessary energy and velocity to follow the planned elliptic orbit. The second occurs upon entry to the destination planets sphere of influence in order to aid orbital capture into Planet B's sphere of influence.

An example is shown in figure 3.1 of a Hohmann transfer from Earth to Mars. A burn at Earth, tangential to the orbit marks the periapse of the Hohmann transfer orbit. The spacecraft then follows the Keplerian orbit until it reaches Mars at the apoapsis of the orbit, and uses a burn to capture into the Mars orbit as necessary.

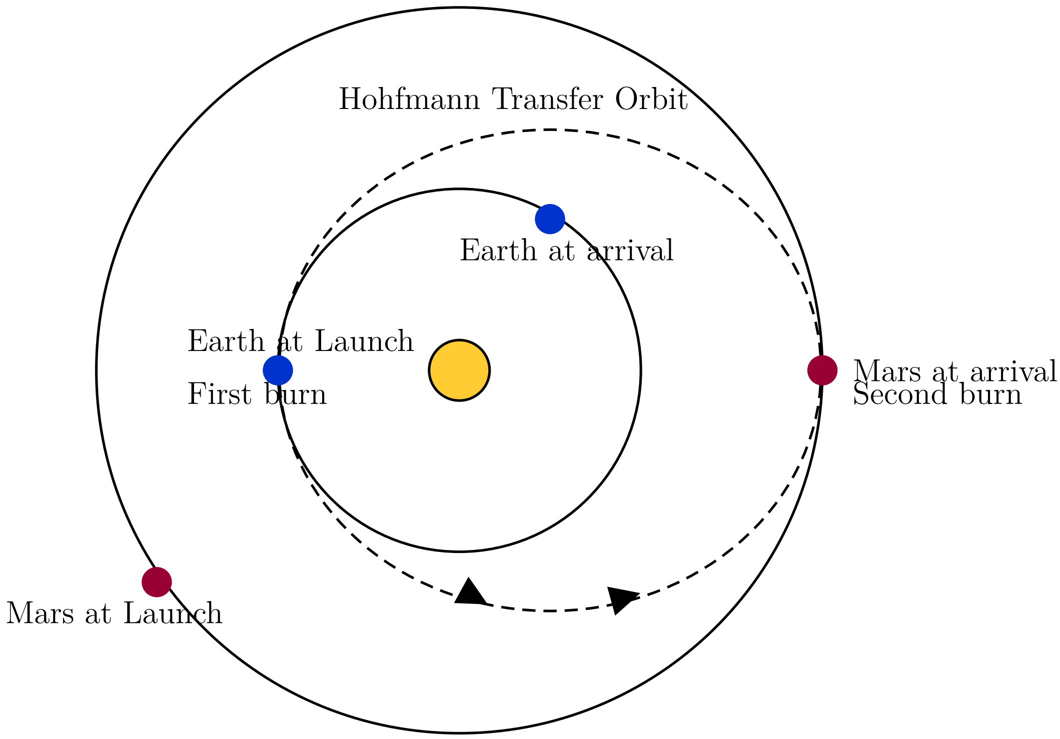


Figure 3.1: Demonstrating a Hohmann transfer from Earth to Mars.

This method uses a very small amount of fuel in comparison to other, faster, transfer orbits. From Earth, it can take over a year to reach relatively nearby planets, such as Mars or Venus. This is because the transfer trajectory takes advantage of a low energy orbit, requiring an initial velocity of only slightly more than the hyperbolic excess velocity. This means that the spacecraft requires only a minimal change in velocity due to propulsion but does require the spacecraft to travel halfway around the sun. There are often opportunities where two planets are much closer together, however a transfer trajectory based on a much higher energy orbit would be required to get there; this would require considerably more fuel at both burn points.

3.3 Lambert's Theorem and Problem

3.3.1 Lambert's Problem

Lambert's Problem is the determination of an orbit, given two position vectors (r_1 and r_2) with respect to a specified central body, an angle θ separating them, and a transfer time between them ($t_2 - t_1$) [19]. The problem was originally stated by Johann Heinrich Lambert in the 18th century, as a means to calculate the orbits of asteroids, comets, moons and planets observed from the ground. But is now fundamental in trajectory planning for spacecraft and orbit targeting for satellites. The problem was first solved with mathematical proof by Joseph Louis Lagrange. The problem is nicely introduced in [20] as follows:

Suppose a particle in a gravitational inverse-square central force field has distances r_1 and r_2 from the centre of attraction at times t_1 and t_2. Let c be the distance, and θ be the central angle between the positions of the particle at the two times, where $0 \leq \theta \leq 2\pi$.

Lambert's problem is that of finding the semi-major axis a or some related quantity for the particle, given $t_1, r_1, t_2, r_2,$ and θ.

It can also be written formally as a boundary value problem [21, 19]:

$$\ddot{\mathbf{r}} - \frac{\mathbf{r}}{r^3} = 0$$

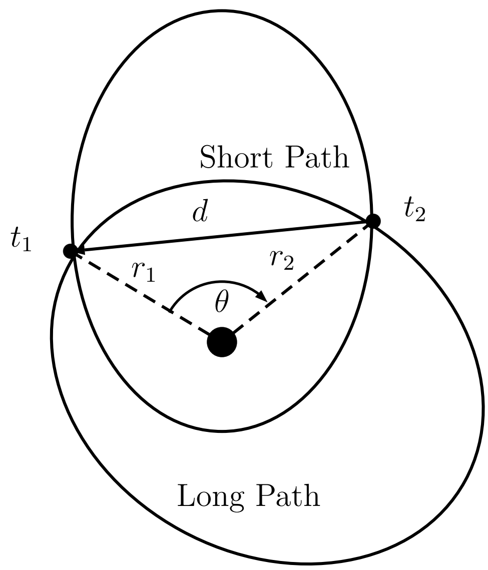
$$\mathbf{r}(t_1) = \mathbf{r}_1, \quad \mathbf{r}(t_2) = \mathbf{r}_2$$

$$r = ||\mathbf{r}||$$

Where r_1 and r_2 are the two specified position vectors that are the boundary conditions for this boundary value problem.

Lambert's Problem will always have at least one feasible solution for $\Delta t > 0$ [22]. A feasible solution is any trajectory that satisfies the boundary conditions and this can be part of an hyperbolic, parabolic or elliptic orbit. The simplest (and most common) case is the elliptic orbit solution. Shown in figure 3.2, when the elliptic problem is feasible there are two orbits (high and low energy) that satisfy the boundary conditions. Problem context and any additional information can help to determine which trajectory is the solution to the problem in question.

Figure 3.2: Demonstrating the four possible solutions to Lambert's boundary value problem.



3.3.2 Lambert's Problem for Interplanetary Trajectories

Lambert's problem was initially a problem of orbit determination, However it is now a fundamental part of interplanetary design. The problem can be introduced and presented in an identical format- the main difference lies simply in the timing of the calculation. Historically, two or three observations of a celestial object were required in order to calculate the orbit. Today, the spacecraft hasn't left the ground and its position at future times is planned and calculated.

The problem can be constructed for interplanetary trajectories as follows:

For elliptic problems, the focus is usually the Sun. Elliptic orbits provide the lowest energy transfers between planets. For hyperbolic problems the focus is often a planet or moon used for a GAM.

As time and location cannot be observed for use in this problem it must be known a priori. For example, if the trajectory required is from Earth to Mars, r_1 and t_1 refer to the location and time of Earth launch with respect to the Sun. r_2, θ and t_2 refer to the location and time of Mars at arrival, relative to the sun and launch location. Choosing a launch date and desired journey time provides easy calculation of the required inputs.

Feasibility wasn't a concern for problems of orbit determination as the time and locations were observed. In this modern use for the problem however, feasible trajectories are of obvious importance. Feasibility is defined by the spacecrafts ability to traverse the trajectory. Hence, there are unavoidable limitations on the energy of the orbit governed both by the spacecrafts thrusting capabilities, and by the initial energy of the spacecraft.

3.3.3 Lambert's Theorem

The orbital transfer time [$t_2 - t_1$] depends only upon the semi-major axis [a], the sum of the distances of the initial and final points of the arc from the centre of force [$r_1 + r_2$], and the length of the chord [c] joining these points [19].

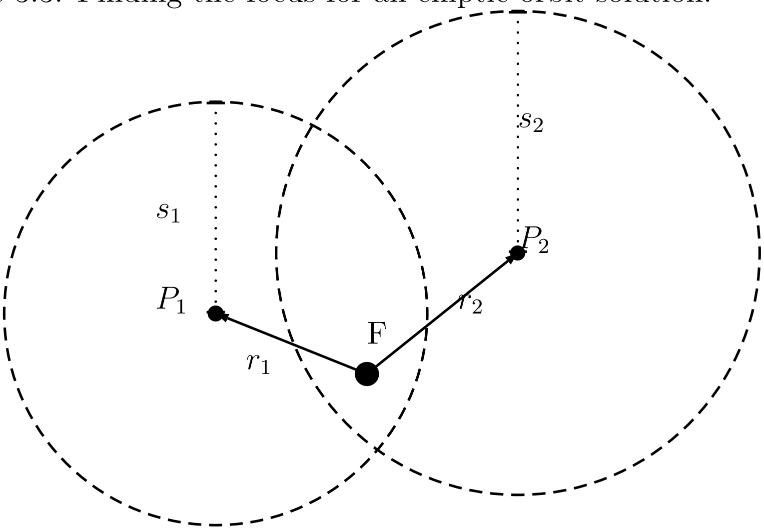
This is represented as follows, by some function f and the gravitational constant μ :

$$\sqrt{\mu}(t_2 - t_1) = f(a, r_1 + r_2, c)$$

r_1 and r_2 are given in Lambert's BVP and c is easily computable using the given angle θ and basic trigonometry. Hence, the only unknown here is the semi-major axis, a.

Assuming that an appropriate value for a (that corresponds to the required time of flight) has been calculated, the position of the second focus must now be located as this will characterise a trajectory satisfying the boundary conditions. The basic geometry of an ellipse implies that for any i, $2a = r_i + s_i$, where r_i and s_i are the distances from a point i on the perimeter to the first and second foci, respectively.

Figure 3.3: Finding the focus for an elliptic orbit solution.



In this problem r_1 is known, therefore the second focus must lie on the circumference of a circle of radius $s_1 = 2a - r_1$ centred at P_1. Since the trajectory must also pass through the point P_2, another circle can be constructed around this centre of radius $s_2 = 2a - r_2$. The intersections of these circles provide a maximum of two feasible locations for the second focus, describing trajectories that satisfy the boundary conditions.

By varying a, the possible locations of the second focus point can be seen to trace a hyperbola, but the specific value of a that corresponds to the required time of flight is required. This is calculated using Kepler's equation (among others) which links the mean anomaly (M) with the eccentric anomaly (E) and eccentricity (e):

$$M = E - eSin(E) \tag{3.1}$$

The mean angular motion per unit time is:

$$n = \sqrt{\frac{\mu}{a^3}} \tag{3.2}$$

The change in mean anomaly can be written in terms of n:

$$\Delta M = n(\Delta t) = n(t_2 - t_1) = E(t_2) - E(t_1) - e(Sin(E(t_2)) - Sin(E(t_1))$$

Rearranging this gives an equation for the transfer time T:

$$T = t_2 - t_1 = \sqrt{\frac{a^3}{\mu}}(E_2 - e\sin(E_2) - E_1 + e\sin(E_1))$$

The eccentric anomaly (E) is also not known at this point. To prove Lambert's theorem, E must be shown as a function of the a, $r_1 + r_2$ and c.

The following relationship is true and can be seen from the geometry of the eccentric anomaly and elliptical orbit:

$$r_1 + r_2 = a(1 - e\cos(E(t_1)) + a(1 - e\cos(E_2))$$

The following results from the cosine rule:

$$c^2 = (r_1 + r_2)^2 - 4r_1 r_2 cos^2(\frac{\theta}{2})$$

Using these two relationships, it is possible to rewrite the transfer time equation in terms of $a, r_1 + r_2$, and c, which proves Lambert's theorem.

Full proofs and further explanations for all trajectory types are discussed in much more detail in [19, 22], and in [20] for the multi-revolution case.

3.3.4 Solving Lambert's Problem for Interplanetary Trajectories

Finding a for the required $T = t_2 - t_1$ cannot be done analytically. An iterative approach is required. Although a was first used as stated in the original theorem, the eccentric anomaly E is a better choice as a offers non-unique solutions and a singular derivative of the transfer time equation [20].

There are three methods commonly used to find the appropriate value for the iteration parameter. Algorithms based on Newton's methods, series expansions or bisection.

Newton's methods can be complicated due to the requirement of a derivative, which is possible to calculate, but costly and not continuous due to the trigonometric function tan giving an infinite value. Series expansions and bisection methods are incredibly simple to understand and implement, however they are often slow to converge.

Once a (or E) is calculated, it is simple to calculate the δV required to execute the manoeuvre using the vis- viva equation, provided the initial velocity vector is known. From this, the cost of the trajectory can be evaluated and the calculated velocities input to assist in the next phase of the trajectory design.

To summarise, there are four main components of a Lambert Solver:

- The iteration variable,
- Calculating an initial guess for the free iteration variable,
- The iteration scheme/process/algorithm (typically a root solver, commonly the Newton-Raphson method),
- A method to derive the velocities required to instruct spacecraft.

3.3.5 Orbit Propagation

A propagator is a model that determines the position of an orbiting body at any instance in time, given an initial state in the orbit (initial velocity and acceleration).

A propagator is important in the construction of interplanetary trajectories as it provides the relationship between time and location.

Similarly to Lambert's problem, there is no analytical method for this, but the process to calculate the polar coordinates (r, θ) of the orbiting body is as follows:

1. Compute the mean anomaly $M = nt$ where n is the mean angular motion as in equation 3.2 and t is the given time for propagation.

   $$n = \sqrt{\frac{\mu}{a^3}}$$

2. Compute the eccentric anomaly E by solving Kepler's equation, given in 3.1.

   $$M = E - \epsilon \sin(E)$$

   This equation must be solved using an iterative technique such as Newton's method.

3. Compute the true anomaly *theta* by solving the equation:

   $$(1 - \epsilon) \tan^2(\frac{\theta}{2}) = (1 + \epsilon) \tan^2(\frac{E}{2})$$

4. Compute the radius from the centre of the orbited planet using Kepler's first law in equation 2.1:

   $$r = a(1 - \epsilon \cos(E))$$

Propagation is computationally intensive due to the requirement of an iterative solution to Kepler's equation. This is an identical issue to that seen in Lambert's problem.

3.4 Constructing Trajectories

3.4.1 The Flyby Sequence

One of the most challenging aspects of interplanetary trajectory optimisation is the determination of the sequence of planetary encounters, or flyby sequence as it is also known.

The difficulty is largely caused by planetary motion and spacecraft capability. Although a flyby sequence is a combinatorial problem with a finite number of possible combinations, a good flyby sequence in one particular time frame may not be in another. There is a dependency between the combinatorial and continuous elements of the problem that demand the two problems are solved in parallel in order to find a truly optimal solution. Combining both elements of the problem leads to a very high dimensional search space however, which is difficult to search.

Some efforts have been made to automate the search for an optimal flyby sequence (these are discussed later in 4.4). But typically a lot of problem knowledge and human intervention is required to at least limit the number of possible flyby sequences to be explored. The next section outlines an energy-based graphical method to determine feasible flyby sequences.

3.4.1.1 Tisserand Graphs

in 1889, during comet observation around Jupiter, Tisserand noted a quantity that remained constant for each comet despite orbital perturbations caused by Jupiter.

Formally, Tisserand remarked that two observed orbiting bodies are possibly the same if they satisfy Tisserand's Criterion:

$$\frac{1}{2a_1} + \sqrt{a_1(1-e_1^2)}\cos i_1 = [\frac{1}{2a_2} + \sqrt{a_2(1-e_2^2)}\cos i_2$$

Where a, e and i are as defined in section 2.2.1.3 and their index refers to the first observation, before perturbation, or the second to the observation after perturbation.

Whilst this criterion wasn't strictly related to GAM's, it has been used to check the assumptions made in trajectory designs for some time (specifically when using the patched conic approach (later discussed in section 3.4.2). The invariant quantity T is:

$$T = \frac{r_{\text{planet}}}{a} + 2\sqrt{\frac{a(1-e^2)}{r_{\text{planet}}}}\cos i$$

Tisserand Graphs are plots of orbital energy (E) or period (P) against the perihelion radius (r_p). Each contour represents heliocentric orbits (they can also represent planetocentric orbits when working in moon system, rather than our planetary solar system). The plots were not drawn or used by Tisserand, but are named after him in tribute to his work as a great astronomer that lead to their use. An example of a Tisserand graph for the inner planets can be seen in 3.4.

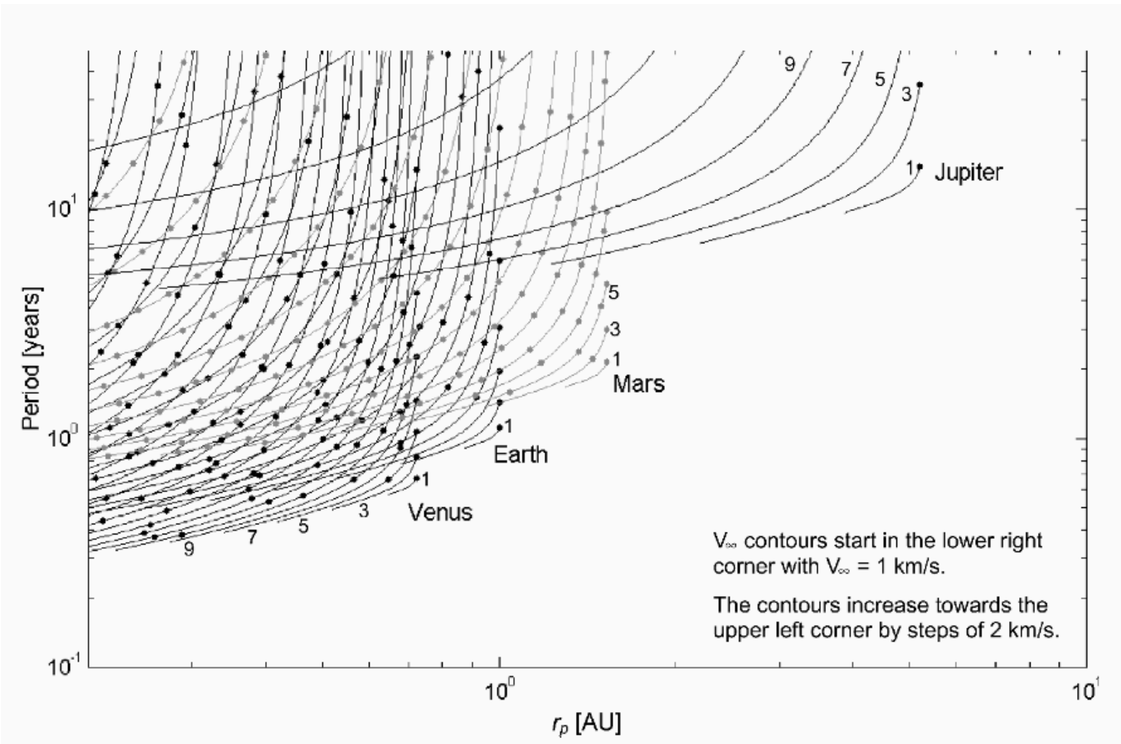


Figure 3.4: Example Tisserand Graph showing possible transfer trajectories between Venus, Earth, Mars and Jupiter [3].

Tisserand graphs are plotted using some assumptions. It is assumed that the planetary orbits are circular and co-planar (which is not true, though most planetary orbits have low eccentricities and inclinations which make this assumption reasonable). It does however render this method infeasible for use in designing trajectories to asteroids, comets or moons with highly inclined or eccentric orbits [23]. In addition, it is also assumed that the spacecraft that would traverse the represented orbits travels in the same plane as the planets.

The Tisserand graphs can be interpreted as follows: Each contour describes a set of orbits with the same energy (here the total energy is considered as the hyperbolic excess velocity V_∞). The contours form into natural clusters as only those that may provide a planetary rendezvous are included. Tick marks (or dots) are placed at intervals relating to a safe flyby altitude above the planet in question. The first tick mark at the bottom left hand side of a contour refers to the lowest heliocentric energy orbit and a negative alignment (spacecraft velocity at an angle π to the planet's velocity). The last tick mark, in the top right hand corner of a contour refers to the opposite, the largest heliocentric energy with good velocity alignment between the planet and spacecraft. The intersections of contours represent possible transfer trajectories between the two planets.

The purpose of a Tisserand graph is to enable a feasible flyby sequence to be determined using the purely ballistic trajectories represented by the contours. The graph contains no data regarding time or phasing however, so deduced flyby sequences are only feasible with respect to energy, and the assumptions made.

A feasible flyby sequence can be determined from a Tisserand graph by following the contours and using the intersections to navigate from one planet to another. It is important to use the tick marks here, one must traverse a length of the contour approximately equal to the interval between the tick marks in order to maintain a safe flyby altitude at each planet. An example can be seen in figure 3.5 for a Venus Earth Earth Gravity assist trajectory to Jupiter.

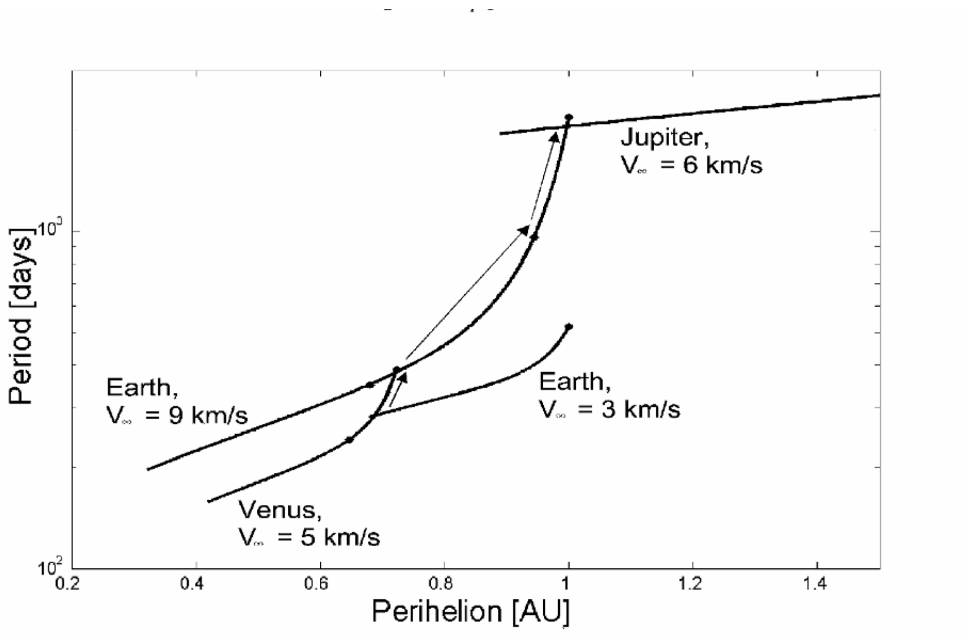


Figure 3.5: A reduced Tisserand graph showing a VEEGA transfer [3].

3.4.2 Patched Conics

Orbital mechanics is well understood for systems of 2 bodies, but this system doesn't provide much help when considering interplanetary travel of the spacecraft. The very nature of the problem means that at least two planets, the sun and the spacecraft must be considered-that's a minimum of 4 bodies.

The patched conic approximation is a method of splitting the n-body problem into a series of 2 body problems. By considering only the gravitational effects of one celestial body on the spacecraft at a time, the problem can be reconfigured to a series of simpler 2-body problems for which the solution method is better understood. The method was first introduced by Breakwell in 1961 [24], where the method was used to find alternatives to the Hohmann transfer between Earth and Mars; he noted considerable improvements in computation time and ease of modelling. The method quickly gained popularity and is commonly found in the literature.

The boundary for splitting the problem (often referred to as the patch point) is determined by a change in the dynamical laws governing the spacecraft. This happens when entering/departing a planet's Sphere of Influence (SoI). This allows the assumption that the effects of gravity from other celestial bodies are negligible within this sphere.

A patch point can also be defined as a burn point for a DSM, or the start/end of a low thrust burn. Although the dynamical laws are not changing, the spacecraft's trajectory changes at this point and the conics describing the spacecraft trajectory before and after the burn would be different.

Figure 3.4.2 demonstrates a simple interplanetary trajectory (from planet A to planet C) using a gravity assist (at planet B) and one DSM using the patched conics approximation. With the help of the approximation, the interplanetary trajectory problem can be split into the following subproblems:

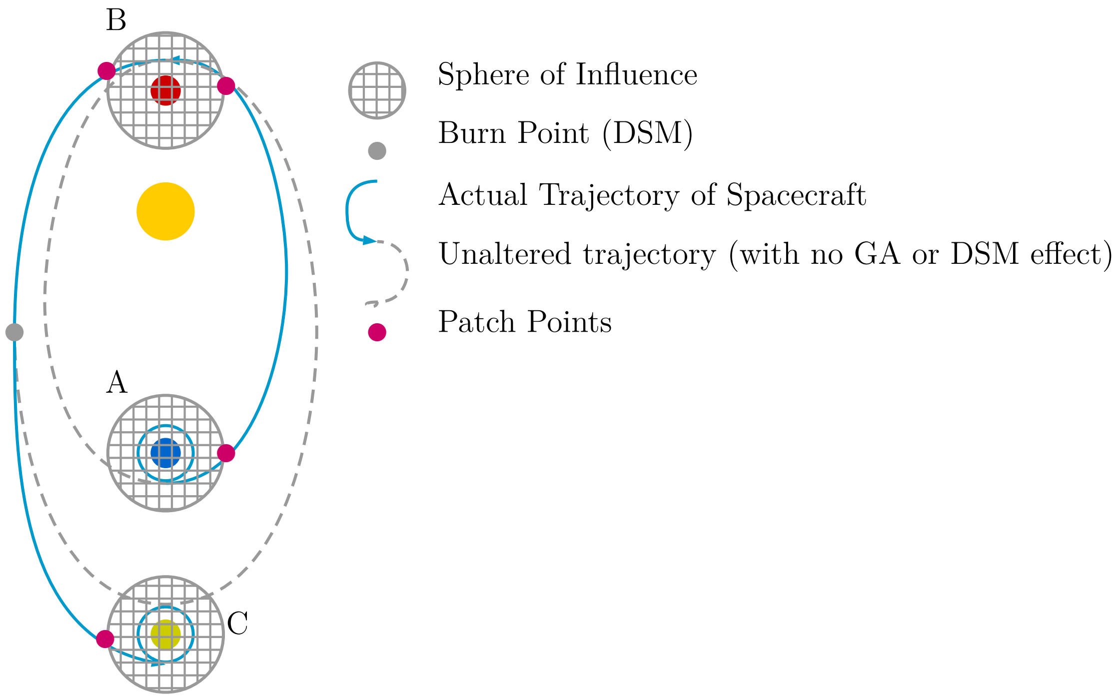


Figure 3.6: Demonstrating a patched conic model for a simple trajectory with one GAM and one DSM.

1. Spacecraft leaves planet A's SoI from parking orbit with an hyperbolic trajectory.

2. Spacecraft follows an elliptical orbit around the Sun (Possibly a Hohmann transfer) to reach planet B's SoI, a DSM may occur on this leg, though does not in the example demonstrated in figure 3.4.2.

3. A gravity assist manoeuvre is performed at planet B.

4. The spacecraft follows an elliptical orbit around the Sun to reach planet C's SoI, a DSM takes place on this leg.

5. The spacecraft enters the sphere of influence of Planet B, with an hyperbolic trajectory. Adjustments may be needed (via propulsion) in order to be "captured" into an elliptical orbit around the planet.

Using the patched conic model described above, a feasible interplanetary trajectory is made up trajectories that satisfy each phase of the mission and stitched together at the patch point. A small corrective propulsive manoeuvre is sometimes required to remove the velocity discontinuity at the patch points, though this can be controlled by providing good constraints for the trajectories.

The patched conics method has been used for real interplanetary missions by ESA and NASA, amongst others, it was used to plan The Cassini trajectories to Saturn, using MIDAS and PLATO, software developed at NASA/JPL [25] and other possible future missions to outer planets using STOUR [26] (See 5 for further information about available software).

The patched conic approximation can be formulated with respect to position or velocity.

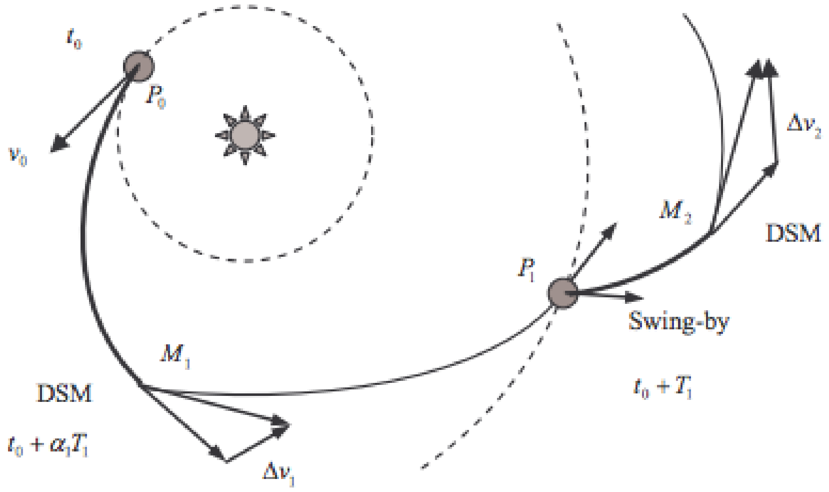
3.4.2.1 Velocity Formulation

The velocity formulation of the patched conic approximation requires the following input information:

- Initial position (P_0),
- Initial velocity (v_0,
- Transfer time for each arc (T_i $\forall 0 < i \leq n$) Where n is the number of legs in the trajectory.

Each leg of the journey consists of two conic arcs. The first is propagated analytically forward in time from the starting planet (P_0) at time t_0, the second is the solution to a lambert's problem connecting the end of the first trajectory to the location of the next planet (P_1) in the flyby sequence with a total time of T_1. The two arcs are connected by a discontinuity in the heliocentric velocity that characterises a DSM (M_1). The DSM is calculated as the vector difference between the two trajectories at the M_1. This can be seen in 3.7.

Figure 3.7: Schematic Representation of a multiple gravity assist trajectory taken from [4].



The location of the DSM is also variable, by describing the lengths of the two adjoining trajectories as complimentary fractions (defined by $\alpha < 1$) of the transfer time of the whole leg. Hence the transfer time for the first trajectory would be αT_1, whilst the second trajectory would be calculated as $(1 - \alpha)T_1$.

The time, velocity and position vectors must be calculated at the end of each trajectory as they inform the calculations for the next leg of the journey.

This formulation is commonly used and examples can be found in [27, 28]. It is possible to transform this formulation into a constrained non-linear program as described in 4.1.3 and solve using existing global search methods that will be outlined in future sections.

3.4.2.2 Position Formulation

The position formulation of the patched conic approximation requires the following input:

- Initial position for each arc (P_0),
- Final position for each arc (P_1),
- Transfer Time for each arc ($T_i \; \forall 0 < i \leq n$) Where n is the number of legs in the trajectory.

As in the velocity formulation, each leg consists of two conic arcs connected by a discontinuity characterising a DSM. Unlike the velocity formulation, each arc in this model can be calculated independently of all others. This is because the times and positions of each flyby are fixed. The location of DSM's can be varied using an angle θ between the position angle of the initial and target position ($\theta_0 < \theta < \theta_1$) and a time of flight t from the initial position to the location of the DSM.

Solutions can be constructed using an elliptic Lambert solver to connect flyby's and DSM's, and an hyperbolic Lambert solver to calculate the fly-by arcs. It is likely a powered fly-by will be required (using on-board propulsion in addition to the GA) in order to create feasible trajectories [4].

Initial analysis or an outer optimisation loop would be required in order to find good fixed points and resulting solutions.

3.4.2.3 Comparing the Two Formulations

The velocity and position formulations are logical methods for the calculation of interplanetary trajectories with the same number of free parameters, however there are some notable differences:

The velocity formulation requires a number of initial value problem to be solved. Each one dependent on all those preceding it. The consequence of these dependencies is exponential growth in the size of the search space as the number of initial value problems to solve increases.

The position formulation requires the solution to a number of independent boundary value problems. The lack of dependency between the problems leads to polynomial growth in the search space as the number of boundary value problems to solve increases.

In contrast, the velocity formulation offers a simplification in the solution process by making use of analytical propagation. This leaves a requirement for the solution of only one, computationally expensive, lambert problem per leg of the interplanetary trajectory. The position formulation however, requires 2.

Using the velocity formulation, ballistic GAM's are favoured as they are an analytical propagation forwards. This is computationally inexpensive and perfectly adequate to satisfy the problem solution and conditions. It may not return the most efficient solution however. A powered GAM, making use of the Oberth effect (as discussed in 2.2.3.2), may provide a better overall interplanetary trajectory. Often a powered GAM is required within the position formulation in order to satisfy the boundary conditions imposed on that leg.

Chapter 4

Global Interplanetary Trajectory Optimisation

Interplanetary Trajectory Optimisation is the task of finding an optimal spacecraft trajectory from Planet A to Planet B. Feasible trajectories are heavily dependent on:

- Spacecraft characteristics (such as mass and propulsion capabilities),
- Flyby sequence (the ordered list of gravity assist manoeuvres),
- Launch date (planetary alignments can play a significant role in the amount of time or fuel required to complete the transfer),
- Mission objectives (This often dictates planet B, but can also specify a required flyby of another Planet or space object for example).

There exists a strong relationship between mission design and trajectory design. Non-negotiable elements of a mission (such as destination planet) must be incorporated within the trajectory design. However, the mission design, and indeed mission feasibility, are also influenced heavily by the features of an optimal trajectory; Launch windows, mission costs, necessary propulsion capabilities, mass limit for spacecraft and payload, and flyby sequence, to name a few, can offer both restriction and opportunity for the mission design.

This problem remains of interest to space agencies as robust time, fuel and cost efficient interplanetary trajectories provide opportunity to optimise our exploration and understanding of Space.

Forward thinking tech-companies such as SpaceX and Virgin will also be interested in advancements in this area as their ambitions to make interplanetary travel more accessible will be hindered without a Spacecraft SatNav they can trust.

The remainder of this chapter will formally introduce the problem of global interplanetary trajectory optimisation; breaking down the solution process and reviewing the literature for each constituent part.

4.1 Problem formulation

In general, the problem of interplanetary trajectory optimisation is too large to be solved in its entirety. The literature tends to deal with specific problem instances, defined by the types of GAM (powered or ballistic), propulsion type (high thrust or low thrust) and sometimes characteristics of the trajectories, such as the number of revolutions allowed for a low thrust trajectory. For example, an ITO-MGA problem is an Interplanetary Trajectory Optimisation problem with Multiple Gravity Assists. ITO-MGA-DSM is as above, with the option for high thrust deep space manoeuvres whilst ITO-MGA-LT provides the option for low thrust propulsion. Ideally, the problem would be solved without having to specify too many features of the trajectory a-priori, however, capabilities are not yet that developed.

4.1.1 Solution Processes

Due to their complexity, ITO problems are typically solved in a series of iterative stages. The large search space and multi-objective nature of the problem splits itself somewhat naturally into phases:

1. The target planet is identified with an initial launch window defined. The

earliest date is constrained by the mission preparation requirements.

2. Possible fly-by sequences are found. These are often identified using existing problem knowledge, simple analysis, or graphically using Tisserand graphs. There can be many possible fly-by sequences, but a feasible fly-by sequence is not always efficient, and may not be feasible when launch windows and the planetary phases are considered at the next stage.

3. Each flyby sequence is assessed for feasibility using a low-fidelity model and defined launch windows. This is completed using existing ITO software or with a custom model constructed from an astrodynamics toolbox and solved using commercial/open-source solvers or global optimisers. A review of these is given in 5. Flyby sequences are assessed for their potential to produce an optimal trajectory, and for robustness given a change in launch date or other problem.

4. A higher fidelity model (using perturbation models, exact spacecraft capability and other mission constraints) can then be used to further assess promising trajectories found in the previous model. This stage can make use of global or local optimisers.

5. Final Trajectory decided.

ITO is a process that has required considerable human input for many years[25]. More recently however, many of the phases required for ITO are being automated and merged so that, in time, finding optimal interplanetary trajectories may cease to be quite such a time consuming and manual endeavour.

4.1.2 Optimal Control Formulation

An optimal control problem consists of a system, described by equations and ”control” variables, a control strategy (values for the control variables) and some desirable (optimal) behaviour for the system to exhibit- in the form of a minimal or maximal function. The aim is to find the control strategy that achieves this optimal behaviour. Formally, the ITO problem can be introduced as an optimal control problem:

For some time t in a specified time period $t_0 \leq t \leq t_f$ consider a dynamical system (Earth's sphere of gravitational influence, for example) described by non-linear differential equations:

$$\dot{y}(t) = f[y(t), u(t), t]$$

t_0, $y(t_0)$ (the initial state of the spacecraft) and $y(t_f)$ (the final state of the spacecraft) are known for some fixed or free t_f. $u(t)$ is a utility (or control) function that is independent of the dynamical system and typically describes the use of thrusters or other propulsion methods on board the spacecraft.

Using a patched conic methodology, the dynamics effecting the spacecraft change as it enters or exits a planets sphere of influence. So a collection of n phases must be considered, characterised by the independent variable time t. For phase $k \in \{1, ..., n\}$, time is constrained to be strictly within some boundary $t_0^k \leq t \leq t_f^k$ and phases are sequential: $t_0^k = t_f^{k-1}$. Within any phase k, the dynamics of the system are described by a set of dynamic equations defined exactly as above, but are characterised with subscript k:

$$\dot{y_k}(t) = f_k[y_k(t), u_k(t), t]$$

t_0, $y_1(t_0)$ (the initial state of the spacecraft) and $y_n(t_f)$ (the final state of the spacecraft) are known for some fixed or free t_f (whether the position or velocity formulation of the problem is chosen will impact the exact boundary conditions defined here). The problem is to find the set of functions $u(t) = \{u_1(t), ..., u_n(t)\}$ that minimise some objective function $J[u] = \Phi(y(t), t)$ subject to constraints imposed on the path (avoiding proximity to Jupiter for example) and the state and control variables.

Condensed, a generalised ITO problem is characterised as follows:

Min/Max

$$J[u] = \Phi(y(t), t)$$

Subject to:

$$\dot{y_k}(t) = f_k[y_k(t), u_k(t), t], \quad t_0^k \leq t \leq t_f^k \quad \forall k \in \{1, ..., n\} \tag{4.1}$$

$$s_k(y_k(t), t) \leq 0 \quad \forall k \in \{1, ..., n\} \tag{4.2}$$

$$c_k(u_k(t), t) \leq 0 \quad \forall k \in \{1, ..., n\} \tag{4.3}$$

Where (4.1) explains the dynamics of the system in each phase, (4.2) constrains the state variables, and (4.3) restricts the control variables.

State variables may be position, velocity and mass of the craft, whilst control variables are magnitude and direction of thrust.

Optimal Control problems are a particularly complex group of optimisation problems as decision variables are functions. This leads to an infinite-dimension optimisation problem. In order to simplify the problem, it is transcribed (often referred to as transcription) into a constrained non-linear program.

Once formulated and transcribed, optimal control problems are generally solved using direct methods, indirect methods, or a hybrid of the two; a good introduction can be found in [29].

Indirect methods work through analytical construction of the necessary and sufficient conditions of optimality. It is then possible to solve the problem numerically. Direct methods aim to find an optimal solution through the construction of improving approximations of the optimal solution.

Although indirect methods are preferable for accuracy, ITO problems are typically so complex that the construction of optimality conditions is too difficult. Thus, a direct method is typically adopted to provide a good approximation.

4.1.3 Non-Linear Program Formulation

This section introduces an example of a non-linear program velocity formulation of the ITO-MGA-DSM problem. It is not a transcription of the optimal control

formulation described in 4.1.2. This is a simplified formulation using a fixed flyby sequence and non-powered fly-bys.
Consider an ordered sequence of n planets where the first planet is (usually) Earth, and the last, n, is the target destination planet. The $n-2$ planets between make up the flyby sequence. Using the patched conics model, allowing one DSM on each arc between planets in the flyby sequence for $\Delta V_{DSM}^{i}, i \in (1, n-1)$.

$$\text{Minimise: } J(x) = \Delta V_0 + \Delta V_{n-1} + \Sigma_{i=1}^{n-1} \Delta V_{DSM}^{i}$$

Where x is the decision vector:

$$x = [t_0, v_0, \alpha_0, \beta_0, ..., t_i, \phi_i, r_i, t_{DSM}^{i}, ..., t_{n-2}, \phi_{n-2}, r_{n-2}, t_{DSM}^{n-2}, t_{DSM}^{n-1}]^T$$

and subject to some constraints:

$$x < X_u$$

$$x > X_l$$

ΔV_i is the ith impulsive movement that controls the spacecraft.
t_0 defines the time of launch.
$[v_0, \alpha_0, \beta_0]$ describe the hyperbolic velocity of the spacecraft at launch.
ϕ_i and r_i define the gravity assist manoeuvre at planet i at time t_i.
t_{DSM}^{i} defines the time of the DSM on arc i (connecting planets $i-1$ and i). This variable was represented as a function of α in 3.4.2.1.
X_u, X_l describes the upper and lower bounds for each variable in x, respectively. These constraints are typically informed by the mission requirements, and capabilities of the spacecraft.

The objective function $J(x)$ is the sum of all of the impulsive movements in the trajectory. This is the equivalent of minimising propellant required or maximising the mass of the spacecraft upon arrival at the target planet (allowing for a maximal payload).

Evaluating the objective function is no easy task. Where v_∞ is the velocity of the spacecraft upon exit of a GAM.

Initialise:

$i = 0$.

$v_\infty = v_0$.

for $i = 1$ to $n - 1$:

1. Propagate to time t^i_{DSM},
2. Solve a Lambert Problem between the spacecraft location at t^i_{DSM} and the planet location at time t^i_{GAM},
3. Calculate ΔV^i_{DSM} that makes this feasible
4. Calculate v_∞ as the result of a GAM at planet i,
5. $i = i + 1$.

As stated at the beginning of the section, this is purely an example to illustrate the problem of ITO in general terms. Different ITO problems will require slightly different systems of equations or variables to represent them accurately. This is due to the many problem characteristics, including: the choice of powered or unpowered fly-by's, fixed or free flyby sequences, and the choice of propulsion.

4.2 Challenges in Global Interplanetary Trajectory Optimisation

Interplanetary Trajectory Optimisation is a large and complex problem consisting of both continuous and combinatorial elements. Combinatorial elements, such as the flyby sequence and spacecraft propulsion choice are often calculated separately and before the optimiser begins to tackle the continuous problem. These combinatorial choices are frequently made by an expert and input as constraints or hard-coded into the optimisation model for the continuous problem. This saves time and decreases complexity by reducing the number of variables and the size of the search space, but could be overly-constraining the optimisation so that a true optimal trajectory will never be found.

The continuous problem is one of time-dependent optimal control of the spacecraft. When should the spacecraft be launched? When should propulsion be used? Despite having relatively few variables this optimisation problem alone is very difficult due to the complex interactions of planetary movements and spacecraft dynamics; it results in a vast, multi-modal and multi-dimensional search space with many clusters of local optima.

The objective function is non-differentiable, so derivative based methods cannot be used and local search methods consistently find themselves attracted to local optima near to their start point; The quality of these solutions is heavily dependent on the quality of the start point provided to the optimiser. This is a problem as there is no way of defining how *good* a solution is.

The vastness of the search space coupled with the computational cost of objective function calculations (comprising of a lambert solver, propagator and ephemeris calculations) means anything close to an exhaustive search is infeasible.

Also of great importance is the format for the optimal solution. Whilst mathematically there will be only one global optimum, a solution must also be considered practically. A good robust solution is superior to an excellent singularity. In other words, it is important to consider that human error and weather may cause a delay to launch, so a larger window for launch is preferable to a short one.

Interplanetary Trajectory Optimisation is truly a global optimisation problem pushing existing global optimisation techniques to the limits. To encourage continued progress in the area, ESA have published several ITO problems and benchmarks to motivate the global optimisation and aerospace community to use the ITO problem as a test-case for their own algorithms and methodologies [30]. They also hold an annual competition GTOC (Global Trajectory Optimisation Competition)[31] that encourages teams from across the world to get involved.
It is clear that the problem of Interplanetary Trajectory Optimisation is a difficult one. The main challenges are:

- The computational cost of objective function evaluations,
- The size and multimodality of the search space,
- The requirement for good initial solutions,
- The manual/separate selection of combinatorial elements of the problem,
- Solution Robustness.

The remainder of this report will introduce the progress that has been made in over-coming these challenges through the literature.

4.3 Lambert Solvers

An efficient lambert solver is key to a fast and less computationally intensive objective function valuation within an optimisation procedure. The following is a summary of the progress made in improving the efficiency of Lambert's solvers.

Lambert's problem was defined centuries ago, however progress in Lambert's solvers accelerated in the 1960's coinciding with the realisation of interplanetary spaceflight. Scientists realised the applicability of the problem and began work to decrease the computational effort required to find a solution.

The first progress in the solution was made by Gauss in 1857 [32]. He presented a classical iterative approach with a successive substitution algorithm (such as the Newton-Raphson method) still used as the foundation of many of the modern procedures. His algorithm worked well for small transfer angles, $\theta << \frac{\pi}{2}$, which suited his purposes, as the celestial observations would often be close together. It is not impossible to use his methodology for larger transfer angles, however convergence is much slower.

More recent developments (specifically with ITO problems in mind) can be traced to the work of Lancaster and Blanchard [20]. In the second of two papers (the first being a brief outline [33]) they provide full derivations for Lambert's equations valid for all cases: elliptic, parabolic and hyperbolic orbits. This was a significant piece of research; the main development here was to describe the transfer time T as a function of parameters q (based on known values) and $x(\alpha)$ instead of a or E which gives a double-valued function that is more difficult to solve. This reduced the complexity of the problem, enabling shorter computation time.

A more efficient formulation of the problem is presented in [34]. The transfer time equation, for all single-revolution cases, is further simplified to require only one hypergeometric function evaluation which can be completed using new recursive identities. Iteratively finding the solution is sped up using Battin's improved formulation due to the reduced effort for each calculation of the transfer time equation.

Battin and Vaughan furthered their study in [35] with notable improvement on the convergence of their Lambert algorithm. Using Gauss' initial work as a basis [36], a geometric transformation helped to simplify the equations involved, and a new free parameter added to Kepler's equations allows for much faster convergence. The geometric transformation fixes $r_1 + r_2, c$ and a but changes the shape of the orbit so that the orbital tangent is parallel with c at periapsis. This simplifies the equations hugely and allows the time of flight to be written simply as Kepler's equation. A good initial guess is now less responsible for the speed of convergence of the algorithm and the singularity at $\theta = \pi$ is removed. The algorithm explained in this paper is referred to as the 'Direct Transfer Algorithm'.

Four years later, the principles of the work outlined in [35] were extended for the multiple revolution case in [37]; developing two algorithms which identify either the high or low energy solutions individually. The algorithms use the same geometric transformation and successive substitution iterative method seen in the direct transfer algorithm. Both algorithms converge in between 5 and 10 iterations and reduce the issues found in Gauss's original solutions.

Gooding interjected next with some improvements building directly on the work of Lancaster and Blanchard that are outlined extensively in [38] and a more condensed version in [22]. The main contributions of the paper are in the discussion of the iteration process and chosen parameter x (as originally introduced in [20], initial estimates, solution accuracy and subroutine testing. Gooding rejects the use of the Newton-Raphson method in favour of Halley's cubic iteration procedure (with 3rd order convergence) due to it's dependency on a good initial estimate. Gooding also provides some depth into the kinematic geometry of the problem, discussing the idea of Lambert-equivalence and Lambert-similarity derived from equivalence classes and congruency within the triangles that define the problem. This leads to a non-dimensionalised iterative process for the solution of the transfer-time equations derived in [20]. The whole procedure is extremely efficient, requiring only 2 or 3 iterations for a solution of good accuracy.

In 1992, Nelson et al. introduced a new approach in [39]. The authors claim that the existing methods, based on conic geometry, are difficult to physically interpret, so have reformulated Lambert's problem based on the free-flight ballistics problem. Using the flyout angle (angle between the initial velocity vector and the local horizontal at the initial position) and initial velocity vector, a time function and velocity function can be used to iteratively find a trajectory connecting two desired locations with the desired transfer time. The process iterates on the flyout angle, which is arguably an easier variable to consider than a function $x(\alpha)$ where α is a function of other orbital elements that has a complicated geometrical and physical interpretation. The developed formulation offers no improvement in terms of computation time, but the papers merit lies in its intuitive formulation and ease of understanding.

2005 saw Lambert's multi-revolution problem begin to be considered in conjunction with low-thrust propulsions systems where chemical propulsion had only previously. Dario Izzo of the ESA Advanced Concepts Team offered some insights in [40]. The paper offers a solution to Lambert's problem using exponential sinusoids that is somewhat simpler than the ballistic arc counterpart. The solution requires the evaluation of the flight path angle in its calculations, and has significantly fewer issues with continuity and multi-valued functions than the traditional problem. Izzo also derives an extension of Lambert's theorem for the low-thrust problem, giving the following equation:

$$\Delta t = F(\tan\gamma, r_1, \frac{r_1}{r_2}, \bar{\theta})$$

Where γ is the initial flight path angle, r_1 and r_2 are the position vectors as previously defined, and $\bar{\theta} = \theta + 2m\pi$ represents the cumulative angle of rotation of the spacecraft as m is the number of complete revolutions of the craft.

In 2008, Avanzini proposed a re-parametrisation of the problem using the eccentricity vector e [41]. The new parametrisation once again enabled a non-dimensionalised version of the problem to be solved, iterating with the Newton-Raphson procedure on the component of transverse eccentricity. Similarly to the approach of Nelson in [39], it offered little gain in terms of accuracy, convergence or computation but they claim benefitted from a simplicity that made it a favourable alternative to Battin and Gooding's approaches. To a mathematician at least, the derivation seems of similar complexity and the use of the the Newton-Raphson method a questionable decision due to its poor convergence given a bad initial estimate.

The classical approaches don't work for perturbed gravitational environments due to their reliance on Keplerian motion. Though perturbation is not such a huge concern for interplanetary transfers, it may be a consideration when transferring between moons in planetary systems, as the gravitational effects of the multi-body systems are far more complex to model. In [42], Lambert's problem with perturbation is formulated as an optimal control problem and solved with the help of the Hamilton-Jacobi-Bellman equation. The HJB equation is a partial differential equation whose solution over the whole state space provides necessary and sufficient conditions for an optimal solution to this formulation.

Izzo builds on the work of Lancaster and Blanchard [20, 33] in [43]. He uses the previously defined iteration variable x, a householder scheme to iterate aswell as calculating an initial guess using a new analytical procedure. The new algorithm converges quickly in mostly 2 or 3 iterations (for the multi-revolution case). Whilst it offers no improvement on accuracy over Gooding's algorithm, there is a reduction in computational complexity that is of note.

[44] contains a review and comparison of many of the main lambert solvers to date. Offering a comparison with respect to speed, robustness and accuracy in addition to a direct comparison of the differences in the four main components of the Lambert Solver. Other reviews of Lambert Solvers have been completed in previous years [45, 46], usually finding Gooding and Izzo's algorithms to be most favourable.

4.3.1 Conclusion

Lambert's Problem still continues to be a topic of interest for researchers due to its complexity. The impossibility of a purely analytical method to solve the problem means there will likely always be research to improve the iterative process with respect to speed, accuracy or robustness.

Many of the main challenges have been overcome. Lambert's problem has been formulated and functioning solvers have been developed for the elliptic, parabolic, hyperbolic [20] and multi-revolution low thrust instances of the problem [40]. Research has even begun to consider the highly perturbed environments of multi-moon planetary systems [42].

Lambert solvers are evolving in line with Space mission design. As long as Lambert solvers continue to be such an integral part of the trajectory calculations, they will continue to develop and improve in formulation applicability, speed, accuracy and robustness.

4.4 Global Optimisation Methods in ITO

A significant challenge in all global optimisation problems is found in choosing the best optimisation algorithm for the job. The wrong algorithm may converge prematurely to a local optimum, take too long to converge or never converge at all. The right algorithm will find the global optimum in a reasonable amount of time. It is important to fully understand the structure of a problem as it can often rule-out, or suggest candidate algorithms.

The following sections introduce many different methods that exist for solving global optimisation problems, and in particular ITO problems.

4.4.1 Stochastic and Heuristic Methods

A Stochastic method for solving global optimisation problems is an algorithm involving an element of randomness. Searching the solution space randomly can help to avoid premature convergence.

A heuristic method for solving global optimisation problems is an algorithm that attempts to find the global optimal solution, but cannot provide any mathematical proof or guarantee of optimality or performance.

Heuristics are used to solve many different types of problem, though they are particularly useful for global optimisation problems as they are much faster than classical methods to find a *good* solution. Particularly for such large global optimisation problems as ITO, classical methods fail to produce optimal solutions and their ability to converge to at least a *good* local optimum is largely determined by the quality of the start point provided to the algorithm. Heuristics, in general, are not dependent on their initial start point and are more systematic in their approach to search a solution space, which often leads to preferable performance and solution quality.

The following sections will introduce the most common stochastic and heuristic methods used for ITO.

4.4.1.1 Simulated Annealing

Simulated annealing (SA) is a stochastic method for global optimisation. The algorithm is inspired by the process of annealing in metallurgy, where a material is cooled slowly and in a controlled way. SA incorporates a similar temperature function, in order to explore a search space by moving to worse solutions with a decreasing probability at each iteration. This limits the possibility of premature convergence.

Sample pseudocode for the algorithm is as follows:

1. Let S be the initial trajectory to initiate the algorithm, and $f(S)$ the corresponding objective function value.

2. Let i_{max} be the maximum number of iterations allowable.

3. $i = 1$, $S^{Best} = S$.

4. let Temp(i) be a temperature function of iteration i.

5. $\forall\ i < i_{max}$:

   (a) Generate S^*, any solution in the neighbourhood of S.

   (b) If $f(S^*) < f(S)$ then $S^{Best} = S^*$.
   Else generate random number r=rand(0,1).
   $\Delta = f(S^*) - f(S)$.
   If $r < e^{\frac{\Delta}{Temp(i)}}$ then $S = S^*$.

   (c) i=i+1.

6. return S^{Best}.

SA is particularly effective for combinatorial optimisation problems, though it is possible to adapt it for continuous and integer optimisation, too.

Simulated annealing with an adaptive neighbourhood paradigm (the neighbourhood of each solution is adaptively changed depending on the acceptance rate of new solutions [47]) SA-AN, was benchmarked against five other global optimisation algorithms (using default parameters) in [30]. It found the best trajectory by far in one of six tests, failing to compete in the other five. Parameters were not tuned for each of the algorithms or problem types. Which, arguably, invalidates this benchmarking exercise as parameter choice could play a crucial roll in an algorithms ability to find a good solution.

4.4.1.2 Monotonic Basin Hopping

Monotonic Basin Hopping (MBH) was first introduced in [48] as a method to solve global optimisation problems found in physical chemistry.

MBH is a robust, stochastic global optimisation algorithm. It works by repeatedly perturbing the returned optimal value of a local search algorithm with a random (uniform) distribution. The perturbed solution forms the starting point for another local search, and the process iterates until some stopping criteria is met. This is outlined in the following pseudocode:

1. X_0 = initial solution.
2. $X_{BEST} = X_0$.
3. Y_0 is the best solution found using a local search method starting with X_0.
4. $i = 1$, i_{max}=maximum number of iterations allowed.
5. $\forall\ i < i_{max}$:
   (a) $X_i = perturbation(Y_{i-1})$.
   (b) Find Y_i.
   (c) If $f(Y_i) < f(Y_{i-1})$ then $i = i + 1$ and $X_{BEST} = X_i$.
6. return X_{BEST}.

MBH minimises issues found when using local search algorithms for global optimisation problems; iterative local search reduces the need for a good initial guess and hopping/perturbation minimises the risk of premature convergence.

MBH is easily parallelised for an even faster algorithm. Parallelised MBH is used in [49] for solving low thrust trajectory optimisation problems. It is shown that the algorithm is fast and even capable of finding good solutions when given an infeasible start point.

An improvement to the random perturbation is recommended in [50]. It is demonstrated that drawing the random variable for perturbation from a long

-tailed distribution such as a Cauchy or bipolar-pareto distribution is superior to drawing it from a uniform distribution as was previously commonplace. The distribution change improved the efficiency and robustness of that algorithm and is considered to be an improvement independent of the problem at hand, though this paper again deals with low thrust trajectory optimisation problems.

Other, similar, examples of the use of MBH can be found in [51] and [52].

[53] compares MBH and SA-AN on benchmark low thrust problems, showing MBH to out-perform SA-AN by a statistically significant margin in all.

4.4.1.3 Evolutionary Algorithms

Evolutionary algorithms take inspiration from the natural evolutionary process to produce optimal solutions through generations of natural selection.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be the non-differentiable objective function to minimise. $x \in \mathbb{R}^n$ is an individual, or candidate solution to the optimisation problem. The goal is to find a solution y such that $f(y) \leq f(z)$, for all z in the search space. this defines y to be the global minimum.

Evolutionary algorithms consist of 4 stages:

- *Initialisation:* A set $S \subset \mathbb{R}^n$ of initial solutions forms a population of k individuals. Individuals $x \in S$ are often chosen at random.

- *Selection:* Not dissimilar to the process of natural selection, the best k individuals are kept, based on their fitness (evaluated as the objective function $f(x)$) and the worst are removed from the population. Different EA's implement different selection strategies; some keep only the best k individuals, whilst others choose to keep some of the worst solutions in the population for diversity.

- *Crossover/re-generation:* Combine features of good individuals (parents) to produce new, hopefully fitter (more optimal), individuals. There are many different methods for crossover, usually chosen based on the characteristics of the individuals vectors. For example, where $\bigoplus$ is the crossover operator, an example crossover between vectors $x, y \in \mathbb{R}^n$:

$$\begin{bmatrix} x_1 \\ x_2 \\ y_3 \\ \dots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} \bigoplus \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \dots \\ y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 \\ y_2 \\ x_3 \\ \dots \\ y_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix}$$

- *Mutation:* To keep some diversity in the population, random mutations are introduced with each generation of a new population. Mutation methods vary between algorithms and are usually chosen based on the characteristics of the individual vectors. For example, a mutation may be a perturbation to the jth vector element by an amount δ. The mutation may be added to all vectors in the population, or a small number of them.

Evolutionary algorithms have many benefits for global optimisation and have been used successfully to solve a number of different problems.

Each iteration is built on a new, mutated, generation of the previous. This makes it unlikely that the algorithm will get stuck in a locally optimal solution. It also means that convergence of the algorithm to a global optimum is not dependent on a good initial solution. This different approach to search makes the algorithm applicable to a wide range of problem types; the objective function need not be differentiable or convex. Another benefit of evolutionary algorithms is how easy they are to parallelise. This can be done simply by splitting the search space and exploring them simultaneously with some communication between the individuals of each search space.

There are a few limitations of these algorithms. One is the lack of guaranteed convergence in a finite amount of time. Convergence is asymptotic. As more of the search space is explored the closer the algorithm will be to convergence, but the rate of progress per iteration will decrease as the proportion of unexplored search space decreases. This means that a termination criterion for the algorithm must be specified a priori. This is either a condition on running time, number of generations or objective function improvement since the initial population. Another limitation of these algorithms is the need to control the algorithm to calibrate to perform optimally for different problems. The algorithms are typically controlled by a series of parameters or methods defining the population size at each generation, the maximum number of generations, the mutation stage and the re-generation stage. This means the algorithm is fully controllable and customisable to get the optimal performance for the problem at hand. The key is to find an optimal balance between exploration of the search space and exploitation of good solutions. Parameters and selection/re-generation methods should be chosen to balance these appropriately for the search space at hand. These are often chosen using a combination of intuition and trial and error. Though the values of these parameters can have a huge impact on the success of the algorithm. Self-adapting parameters have been used in some case in an attempt to optimise the parameters for algorithm performance.

An overview of the use of evolutionary algorithms within interplanetary trajectory optimisation and related problems is presented in [54].

4.4.1.4 Differential Evolution

Perhaps one of the most successful evolutionary algorithms applied to ITO problems is *Differential Evolution*. The algorithm was first developed in 1997 in [55]. The algorithm is different in the approach of most evolutionary algorithms as it considers *agents* exploring a search space, where the agents exchange location information between one another, as opposed to a reproductive/re-generation process. The algorithm can be summarised for a population ≥ 4 agents as follows:

1. Initialise population of agents at random locations in the search space $\mathbb{R}^n$.

2. Until a specified termination criteria is met repeat for each agent $x \in \mathbb{R}^n$:

- Choose 3 distinct agents, $m, n, p \neq x$.
- $\forall i \in \{1......n\}$ generate $r_i \sim U(0,1)$.
- if $r_i < M$, (where $M \in [0,1]$ is a specified crossover probability parameter) then the new location y of the agent x is calculated as $y_i = m_i + F \times (n_i - p_i)$ (F is the differential weight).
- if $f(y) < f(x)$ then move agent x to location y.

3. return the agent(s) with the best fitness function value.

Once again, differential evolution has a few parameters to set in order to tune performance for the particular problem at hand.

Differential Evolution was first introduced to the global interplanetary trajectory optimisation problem in [56] in 2004. It showed a marked improvement over genetic algorithms on benchmark problems and inspired further exploration into the use of the algorithm. A 2007 study [57], analysing the performance of Differential evolution on ITO problems found their performance to be heavily influenced by chosen parameters for the algorithm. Since then, researchers have focussed their efforts on finding the optimal parameters for each problem. This could be done by testing parameters on a subset of the problem, however this is potentially time consuming and there is little known to define an optimal parameter for these problems. It is therefore better to continually improve the parameters in parallel to the search for an optimal interplanetary trajectory. This has resulted in the development of *self-adaptive* algorithms such that the optimal parameters for the problem may be tested, learned and applied during the search for the optimal solution [58, 59]. This approach does increase the complexity of the algorithm, however testing shows it to yield better solutions than DE algorithms without the adaptations [60].

4.4.1.5 Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) is another optimisation algorithm inspired by nature. It aims to emulate flocking/swarming behaviours of animal groups such as birds, ants and fish. Similarly to differential evolution and other evolutionary algorithms, the search space is explored via a group of agents, known as particles in PSO, that share information with one another. In this instance, the information shared is the best achieved group position. Agent i is located at position x_i^t at time t. It is instructed, at time t to search the space via a calculated trajectory v_i^t. The trajectory and resulting position is calculated as follows:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (xBest_i^t - x_i^t) + c_2 r_2 (gBest_i^t - x_i^t)$$

$$x_i^{t+1} = x_i^t + t v_i^t$$

Where $xBest$ and $gBest$ are the best particle and group position respectively. ω is inertia weight, c_1, c_2 are two positive constants and $r_1, r_2 \in [0, 1]$ are two random parameters. In this way, an agent is instructed to search the space taking information from its best known solution, the global best known solution, and some stochastic perturbation.

In [61], a multi-objective formulation of the ITO-1GA problem is considered, and MOPSO (multiple objective particle swarm optimisation) shows to perform well.

Benefits of the PSO algorithm are the easy implementation, with relatively few parameters and fast convergence [62, 54]. The main issue found with PSO algorithms however, is its lack of safeguarding around convergence to local minima. Variations on the algorithm could be developed to minimise this risk.

4.4.2 Deterministic Methods

A deterministic method for solving global optimisation problems is an exact algorithm that will always produce the same output for a given input. There are no random elements. Deterministic algorithms are theoretically more robust and reliable, but testing generally confirms stochastic and heuristic methods to converge faster [63, 56]. The following outlines some of the most successful deterministic approaches used in ITO.

4.4.2.1 Tabu Search Methods

Tabu search was first introduced in 1986 in [64] as a method for combinatorial optimisation.

The Tabu Search algorithm maintains a short-term memory detailing the impact of recent moves within the search space. The memory is accessed at each iteration and prevents returning to recently visited areas of the search space, referred to as cycling.

Medium memory structures can be used to encourage moves toward promising areas of the search space and Longer-term memory structures can ensure diversity in the search across the search space.

For ITO, an adaptation of the initial algorithm is used. This is known as the Enhanced Continuous Tabu Search (ECTS) and is based on the original algorithm. The algorithm is commonly hybridised with a local search performed at each iteration, not dissimilarly to SA and MBH.

Tabu search is rarely used alone in the literature. Given its merits in combinatorial optimisation, the flyby sequence might be a natural application for the algorithm. The dependency on the continuous elements of the optimisation prevent it from being such a simple task, however.

Perhaps the algorithms most extensive use is in [65] as a search methodology for spacecraft trajectory tours. The algorithm shows favourable performance when coupled with a search space pruning technique, identifying previously unknown best solutions for the GTOC 4 competition.

[66, 67] make use of tabu search in conjunction with other global search methodologies- this is discussed in greater detail in 4.4.3.

4.4.2.2 Monte Carlo Tree Search

Monte Carlo tree search was first introduced in 2006 in the context of two person zero-sum games [68]. The algorithm exists to answer the following question: Given the current state of gameplay, what is the most promising next move?

A combination of tree search and Monte-Carlo simulations, a node of a tree is evaluated by averaging the final outcome of several random simulations. The process consists of four steps, performed iteratively until a stopping criterion is met.

- Selection: A selection policy is deployed, starting at the root of the tree (an initial, partial trajectory). This policy must balance the need for exploration of new parts of the search space, with exploitation of promising paths.

- Expansion: When a leaf node is reached, the next action is chosen at random, expanding the tree by an additional node. This random action is a planetary encounter or time of flight.

- Simulation: A Monte-Carlo simulation is run, selecting actions either randomly, or informed by a weighting of the potential of an action to produce a good trajectory.

- Back-Propagation: Upon successful construction of a trajectory to the target planet, the objective function value is propagated back through the tree and each node defined in step 1 is updated with this new information.

MCTS is particularly applicable to ITO problems due to the sequential nature of the trajectories and its scalability to very large problems.

MCTS is successfully implemented for ITO problems in [69]. A fifth contraction stage is implemented in order to prune the tree of trajectories dependent on too many planetary encounters, or too large an objective function value. Not dissimilar to branch and bound pruning methodology, this step maintains a broad and shallow tree, most likely to contain an optimal solution.

Benefits of this method are its ability to incorporate the flyby sequence into the optimisation process. There is no need to specify it a priori, which most algorithms discussed in this chapter are incapable of coping with. The structured nature of the search also results in the need to solve orders of magnitude fewer Lambert problems, providing a significant improvement in computational cost and efficiency. The algorithm show a lot of potential for a completely automated approach to the full ITO problem.

4.4.3 Hybrid Algorithms

Hybrid algorithms merge two or more algorithms together in an effort to mitigate the weaknesses of one algorithm with the strengths of another. There is no strict method by which this can be achieved; one hybrid algorithm may run two search algorithms concurrently and share information between them. Whilst another may embed an algorithm inside another, creating an inner and outer loop, for example.

Studies show the alliance of multiple algorithms can be beneficial when solving ITO problems:

In [70], a genetic algorithm is used in an outer loop solving for categorical variables (including flyby sequence) whilst DE and PSO are used co-operatively in an inner loop to solve the remainder of the problem. The hybrid algorithm provides the better performance when compared with a non-hybrid optimiser.

In [71], a GA, DE and PSO are run in parallel, capable of sharing best known solutions between them at regular intervals. This is commonly referred to as the island model in the literature. The algorithm leads consistently to global convergence for simple transfers. Though performance is similar to the DE algorithm alone for simpler problems, it does offer improvement in more complex scenarios.

[66] goes ones step further than [71]. This paper outlines the use of GA, DE and PSO used in parallel with two additions. Firstly, a mass mutation operator is introduced, which diversifies the population in the event of stagnated progress, preventing premature convergence. Secondly, The method is initialised

with a module based on an enhanced continuous tabu search algorithm, which provides a good starting population for the optimisation algorithm. The results show that the addition of tabu search is largely responsible for improvements in performance, efficiency and computational cost. A similar approach is presented in [67] also with good results.

[72] describes a competitive algorithm for MINLP based on Ant Colony optimisation and the Oracle penalty method for constraint handling. The algorithm performs well on all benchmarks (both ITO and other aerospace problem sets), finding new best known solutions on some problems.

In [73], the global search strengths of a genetic algorithm are combined with the local search benefits of recursive quadratic programming. The authors achieve a significant reduction (two orders of magnitude) in the number of objective function evaluations required when compared to a grid search methodology. The authors do not comment on the performance with respect to the genetic algorithm alone

Inevitably however, combining so many algorithms can lead to an increase in complexity and run-time. This isn't so much of an issue for ITO problems specifically, as run-time for an algorithm is not heavily constrained, though speed would enable an iterative mission design process.

[74] presents a hybrid deterministic-heuristic method. The heuristic element of this model is an evolutionary based algorithm. The deterministic element of this model is the division of the search domain into successively smaller subdomains until convergence is reached. Domains spawn from divisions made at the best and worst current known solutions. Many subdomains are produced in this way, a fitness function is calculated for each subdomain and then they are ranked, and only those subdomains deemed most likely to contain optimal solutions are searched. The fitness function is based on the quality of the evolutionary algorithms exploration within the subdomain and the proportion of the search space left unexplored.

This is an intelligent approach to search space exploitation and shone light on the potential of search space reduction principles that is explored in more detail in the next chapter.

4.4.4 Search Space Exploitation

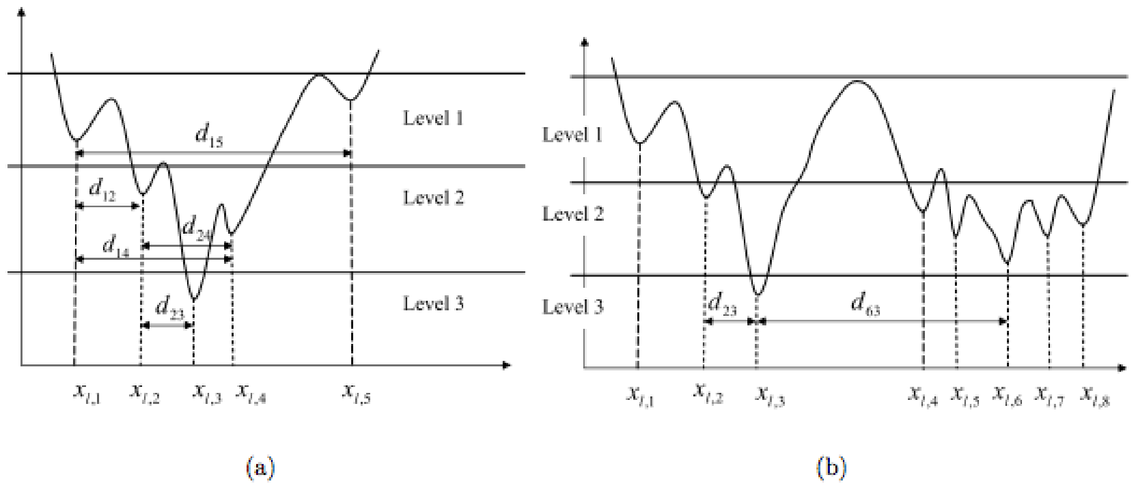
Understanding a search space can be key to developing a fast and efficient algorithm for global optimisation problems; knowledge of its patterns and structure can help to inform a logical optimisation strategy- indicating a good choice of algorithm and tuning parameters.

The global ITO problem has a vast search space. Too vast to search in its entirety. Instead, the problem is constrained and discretised to reduce the size and increase the likelihood of finding a good solution. This paper will refer to these as subproblems of the Global ITO problem.

It is well known that, in general, the search spaces of these ITO subproblems are large, multi-dimensional spaces with many clusters of local optima. The space is continuous when unconstrained, with some periodic patterns due to predictable planetary orbits.

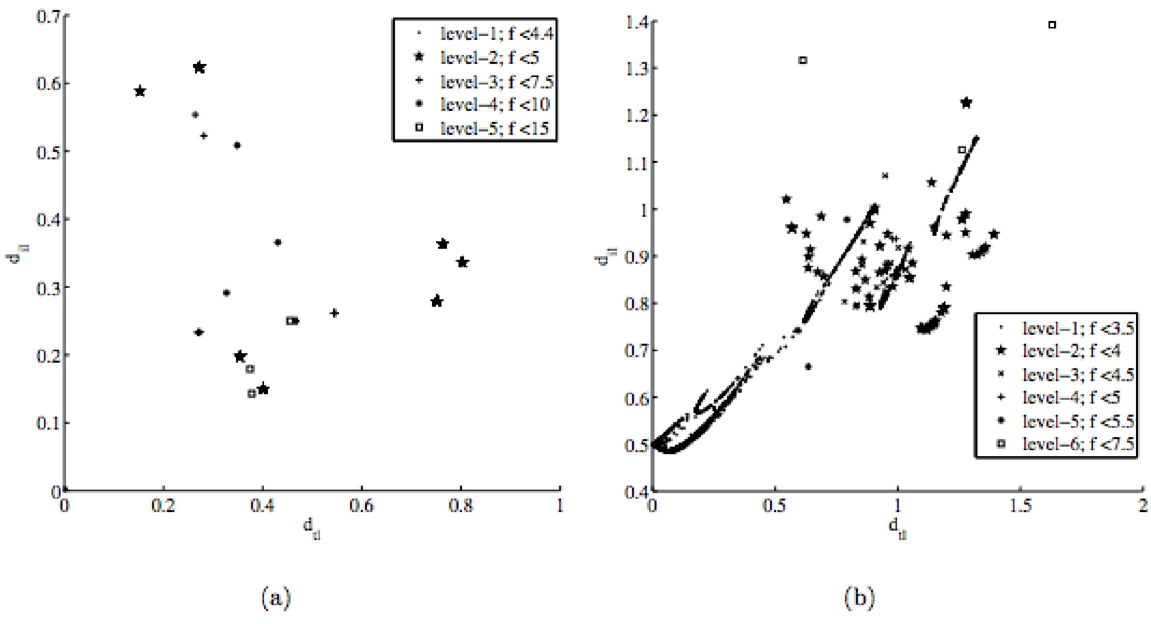
A deeper understanding of the subproblem search spaces is given in [5]. Four different subproblems were considered each with varying numbers of gravity assists and deep space manoeuvres, formulated in two slightly different ways. No low thrust problems were analysed. The analysis looked at the distances between local optima within and between different levels of depth of funnelling local optima. This can be seen in Figure 4.1 where d_{il} is the relative distance of each local minimum within the same level (intra-level), and d_{tl} is the relative distance between local minima in different levels(trans-level). These two metrics can provide a lot of information about the distribution of local minima in the search space. Trans-level distances can provide information about the funnel structure of the space, combined with the intra-level distance it is possible to consider the probability of transfer to more optimal solutions. Understanding of the funnel structure of a search space could motivate algorithm design or developments that

Figure 4.1: One dimensional example of a) single funnel structure and b) bi-funnel structure from [5].



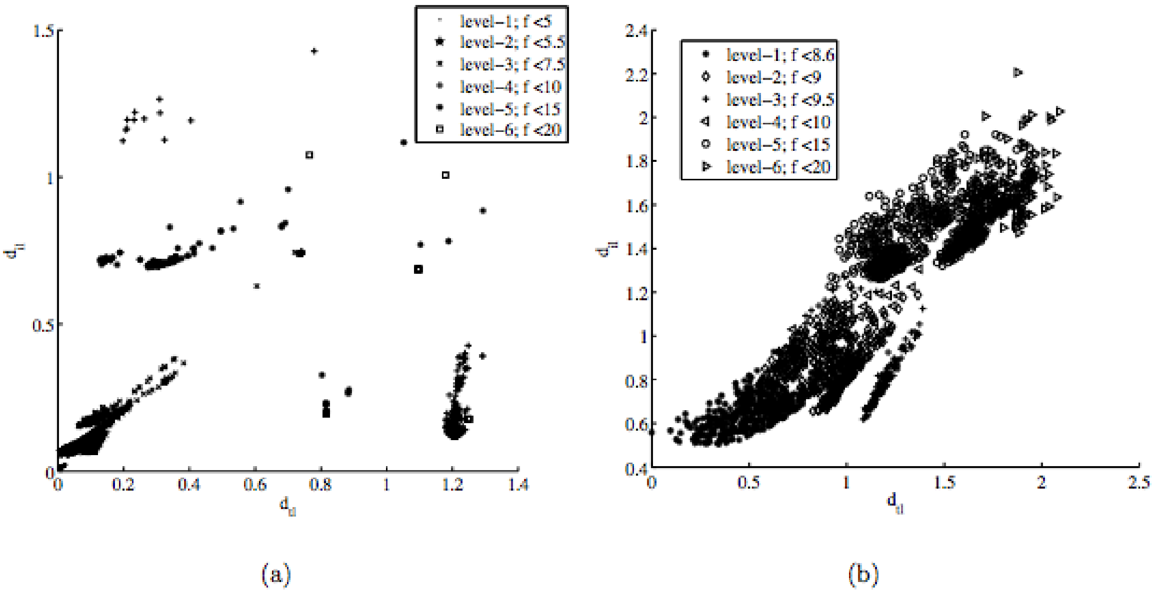
help to avoid premature convergence at local optima. Additionally, Having information about the distances between optima, both within and between defined levels, could aid in parameter tuning for algorithms such that they are less likely to converge prematurely.

Figure 4.2: Relative distance of local minima for a) an Earth-Asteroid transfer with two impulses and b)an Earth-Venus-Mars transfer with gravity assist and DSM. Taken from [5]



Figures 4.2 and 4.3 show all three metrics where the levels are defined by the objective function value f. Basic analysis indicates an increased number of local minima as the complexity of the subproblems increases. The most simple problem with only two impulses has an extremely sparse distribution, with few optima in figure 4.2a. Similarly, a GA only problem has a seemingly random, but more clustered distribution in figure 4.3a. The combination of DSM's and GA's greatly increases the multi-modality of the search space; forcing the requirement of more complex search techniques, but also increasing the likelihood of a lower global optimum than just GA's or DSM's alone.

Figure 4.3: Relative distance of local minima for an EVVEJS flyby sequence with gravity assists and a) no DSM's or b)DSM's. Taken from [5].



Graph b of figures 4.2 and 4.3 suggest that ITO-MGA-DSM problems may have similar distributions of local optima. The EVM transfer has considerably less local optima than the EVVEJS problem, as is to be expected. However both show some clustering within levels and a near continuous distribution of optima.

Clusters with similarly small values for d_{il} and d_{tl} show that it is likely easy to transition from one level to another, suggesting a single funnel structure (figure 4.1a). This is evident in figure 4.3b. Figure 4.3a however, suggests a more complex funnel structure. More similar to that in figure 4.1b. This is evident in the distinct clusters found in level 5. Level 3 and below, however, has a more uniform distribution that could be attributed to a single funnel structure.

The work of [5] confirms that the more complex the subproblem, the more complex the solution space. The addition of MGA's and DSM's dramatically increased the number and distribution of local minima.

To the author's best knowledge, this analysis has not been carried out for low thrust search spaces. This presents difficulties when drawing conclusions about the full ITO search space. Though, naturally, it will be of much higher dimension.

4.4.4.1 Search Space Reduction Algorithms

One of the main issues with ITO problems is the size and complexity of the search space. It contains many local minima due to nonlinearities and periodic planetary motion. These local minima can cause problems when using local search algorithms due to their tendency to converge prematurely. Global search algorithms can be used, though depending on the complexity and size of the problem, it may be unrealistic to expect the algorithm to converge to a global optimum in any reasonable period of time.

A solution to this problem is to use knowledge of the problem and search space to reduce the search space by removing areas unlikely to contain optimal or even feasible solutions.

The Gravity Assist Space Pruning (GASP) Algorithm was first introduced in [75], solving the ITO-MGA problem. This model does not allow for DSM's but uses a powered swing-by model. The GASP algorithm was designed to efficiently detect and prune unfeasible parts of the search space, leaving several sets of box bounds with vastly smaller contents. These reduced box bounds may then be optimised efficiently using a standard optimisation method. The paper demonstrates that the search space can be reduced using forward and backward constraining on the following bounds:

- Hyperbolic excess velocity C^3 (launch speed),

- Maximum gravity assist thrust,
- Minimum gravity assist flyby radius/angle,
- Maximum breaking manoeuvre.

GASP is a successful search space reduction algorithm offering search space reductions in the region of 6 orders of magnitude. It is also polynomial in time and space complexity (using a simple grid search optimisation technique- other optimisation algorithms may produce a different complexity).

GASP was later built on to consider one DSM on each leg of the journey [76]. The addition increases the complexity of the problem greatly due to the additional Lambert's problems within the objective function evaluations,and increased dimensionality. These problems were combatted somewhat in [77] using differential algebra to generate a Taylor expansion to approximate the objective function. The search space is split into boxes, and the approximation is evaluated over these- as opposed to a point-wise evaluation using grid sampling in the earlier version of the algorithm. Evaluation of the approximation is considerably less expensive than the objective function and makes GASP applicable to the ITO-MGA-DSM problem.

GASP was also modified for use on Low thrust missions with fixed revolution trajectories in [10].

In all phases of the GASP evolution, it effectively reduces the search space, removing many infeasible solutions that create noise in the optimisation. This results in faster convergence to good solutions and a higher proportion of good solutions identified.

4.4.5 Machine Learning techniques

4.4.5.1 An Introduction to Machine Learning

Machine learning tools are a collection of algorithms and techniques that learn from observed data by constructing mathematical models that explain those observations.

The fundamental goal of machine learning algorithms is to apply the learnings from the training data to successfully interpret new, unseen data, much like a human.

These algorithms have many possible applications but are commonly used to aid in classification, forecasting, predictive analytics and decision making.

Issues with machine learning algorithms commonly stem from the training of the algorithm. It is important the algorithm is trained on as wide and varied a data set as possible, but is also representative of true events. A data set that is too small may be biased and unrepresentative of the distributions of true events. Biased training data can lead to over-fitting. For example, the learning algorithm may perceive a pattern that is not present in a larger data set. Highly complex algorithms may require large, multi-dimensional data sets to be adequately trained, however these can be difficult to acquire in the initial instance.

4.4.5.2 Machine Learning in Interplanetary Trajectory Optimisation

In 2004, Bernd Dachwald presented the first combination of evolutionary algorithms and neural networks to solve low thrust interplanetary trajectory optimisation problems[78]. These are referred to as Evolutionary Neuro-control algorithms. The spacecraft is steered by the Neuro-controller based on a set of parameters chosen for their suitability to produce an optimal solution. The Neuro-controller employs the evolutionary algorithm to learn the optimal control strategy for the spacecraft.

This algorithm performs exceptionally well, providing trajectories far better than the benchmarks for the tested problems. Another benefit of this method is that there is no need for a good starting point, so an expert is not required to be on hand. In 2009 this work was extended to include multiple gravity assist manoeuvres [79].

In 2009, the advanced concepts team at ESA suggested another hybrid solution of an evolutionary algorithm and artificial neural network to solve ITO problems [80]. In this implementation, the artificial neural network provides an approximation for the computationally expensive fitness function, significantly reducing the computational cost of the algorithm. The results of the preliminary tests were positive, Using the approximated fitness function not only reduced computation time, but also consistently improved the quality of solutions found. This is likely because the approximated fitness function is smoother than the original, preventing the evolutionary algorithm from getting stuck in local optima; clusters of local optima are smoothed, whilst robust global optima remain attractive to the algorithm.

This work was later extended ([81], [82]) and adapted for use with low thrust propulsion, allowing for multiple-revolution solutions. Multiple machine learning techniques were tested to replace the lambert solver approximation for low thrust arcs. The solution to a simplified optimal control problem was still required as an input to the regressors, however the increase in the approximation accuracy as a result of this was astounding. All methods show a considerable improvement over the lambert approximation- over an order of magnitude better in terms of error. Neural networks were proven a good choice, though gradient boosting was the best performing method in this instance.

In 2012, LeGO (Learning for Global Optimisation)[83] was introduced. LeGO is a generic framework that is not limited to ITO problems, though when tested on the problems showed considerable potential. The idea behind LeGO is to train a machine learning tool (In this instance it is a support vector machine, however there is no reason a different, appropriate, method could not be applied.) to classify start points for a global optimisation algorithm to be good or bad (likely/unlikely to lead to a globally optimal solution). The tool is trained on data from initial runs of the global optimisation algorithm used, with start points obtained from grid sampling. The main benefit from this framework is the ability to discard bad starting points, spending the time exploring more promising ones instead. Results show the tool was not only producing better solutions (closer to the global optimum) than the unaided global optimisation algorithms, but also a higher proportion of better quality solutions. This is a great benefit for mission planners in particular, who like to have multiple trajectory/launch date options to choose from.

More recently, in 2018 [84], deep neural networks were considered to represent the optimal guidance profile for an interplanetary mission, on-board the spacecraft.

The last decade has seen a significant increase in the application of machine learning and artificial intelligence in almost all areas of research. Specifically, in the field of interplanetary trajectory optimisation it has been used to overcome many of the challenges outlined earlier in the paper.

Hybrid Evolutionary/Machine Learning algorithms improve the quality of solutions by an incredible margin; [78] uses machine learning to control and direct the evolutionary algorithm reducing the need for expert input, whilst the works of [80, 82] use machine learning as an efficient approximation for the computationally expensive elements of the optimisation, whilst simultaneously smoothing the search space, allowing the algorithms to cover more ground with less chance of getting stuck in local optima. LeGO [83] offers the beginnings of an artificially intelligent approach to the assessment of good starting points. However more broadly, it is possible it could be the start of an artificially intelligent approach to search space pruning and constraining. Current assessments of the search space are limited to the application of feasibility constraints, that although offer a significantly reduced search space, offer no opportunities to identify and discard feasible, but high cost areas of the search space.

Chapter 5

Software

Many toolboxes, solvers and software are available to assist in the problem of interplanetary trajectory optimisation.This chapter outlines many open-source, commercial and academic software available.

5.1 Astrodynamic Toolboxes:

The following toolboxes are libraries of particularly useful astrodynamics functions that can be used within applications and when parsing the ITO problem to a solver or global optimiser. ITO software and toolboxes typically house a Lambert solver, propagators and local/global optimisation algorithms.

The search for astrodynamics toolboxes that are of use in the ITO problem revealed a small number available.This is largely due to the fact that the ITO problem is not yet commercially interesting. ITO missions are completed by non-profit organisations so the majority of toolboxes focus around orbital mechanics and orbit raising/transfers of Earth orbiting satellites which has some commercial value. The two most useful toolboxes, PyKep and TUDAT have been produced by ESA and TU Delft, Scientific and Academic institutions, respectively, who are less interested/dependent on the commercial profitability of their toolboxes.

5.1.1 PyKep

PyKep [85] was developed by the ESA Advanced Concepts Team. It is a scientific C++/Python library of efficient astrodynamics tools and solvers separated into different modules. Most notably, and usefully for ITO problems, is the core module, which contains efficient multi-revolution lambert solvers, Keplerian propagators and a hyperbolic fly-by propagator.

PyKep also contains the Trajopt module which uses PyGMO to solve many different instances of the ITO problem:

- MGA-1DSM (Multiple gravity assist with 1 deep space manoeuvre),
- pl2pl-nDSM (Planet to planet with n deep space manoeuvre),
- LTMGA (Low-thrust multiple gravity assist),
- LTMR (Low-thrust multiple rendezvous).

Low Thrust problems are solved with help from the SimsFlanagan module, which offers a low thrust ITO model modified to include high fidelity propagation and easier mesh refinement.

Finally, PyKep also offers orbit plotting capabilities for some visualisation of final trajectories.

5.1.2 TUDAT

TU Delft Astrodynamics Toolbox (TUDAT) [86] is a C++ library designed to provide users with the functionality to simulate various astrodynamics applications such as mission analysis, orbits, space propulsion, ascent and re-entry trajectories and planetary exploration. The toolbox provides a number of features such as numerical integrators, mission segments, gravitational force environment models and unit converters that are all somewhat useful when modelling an ITO problem.

The library is built and maintained by staff and students at TU Delft, mainly as part of MSc theses (see [87] and [88]). The library is consistently maintained.

TUDAT is a larger library than PyKEP, and offers more in the way of environmental models and tools for general astrodynamics applications. This library is not specifically geared towards ITO problems, but has many useful functions (Lambert solvers and propagators for example) that enable it to be used for such problems, if paired with good solvers/optimisers.

5.1.3 Poliastro

Poliastro [89] began life as a University project but has since expanded as an attempt to combine Python best practices with orbital mechanics. The toolbox contains SPICE/PYKEP capabilities (Izzo's Lambert solvers and orbit and manoeuvre classes). Although not specifically compiled to aid with ITO problems, the functions and classes contained within this toolbox could be useful for a more hands-on approach.

The toolbox currently undergoes regular updates and the developer aims to keep up to date with the literature in both Python efficiency and astrodynamics.

5.1.4 Orekit

Orekit [90] is a low-fidelity space dynamics library written in java. It is built and maintained by CS systèmes d'information and offers basic elements and algorithms for use in space applications. Whilst again, not specifically designed for use in ITO problems it contains efficient propagators and frameworks that could be useful to describe the problem before parsing it to a solver or global optimiser.

5.2 Solvers/ Global Optimisers

As ITO problems can be formulated as non-linear programming problems, they can be solved using some large scale solvers and global optimisers. Below are a some popular options for this:

5.2.1 PyGMO/PaGMO

The Parallel Global Multi-objective optimiser [91, 92] is a general optimisation framework based on the asynchronous island model paradigm. Written in C++, it works by allowing multiple algorithms to run simultaneously on different cores (islands). The different islands can communicate in a user determined fashion in order to share best solutions between the islands.

PaGMO works with many existing algorithms, and can also incorporate the users own, new algorithms into the optimisation process. PaGMO was developed by Dario Izzo of the ESA Advanced Concepts Team

5.2.2 WORHP

WORHP (We Optimise Really Huge Problems) [93] is a large scale sparse non linear optimiser developed as the official NLP solver of the European Space Agency. The solver has many interfaces and is available in many programming languages. It makes use of sequential quadratic programming and interior point methods for global optimisation of NLP problems.

WORHP also offers outputs in addition to good solutions, in the form of plots and real time solution evolution. WORHP is maintained and updated by researchers at the University of Bremen.

5.2.3 MIDACO

MIDACO (Mixed Integer Distributed Ant Colony Optimisation)[94] is a derivative free heuristic solver that makes use of efficient parallelisation strategies to solve large, MINLP(Mixed Integer Non-Linear Program) problems. MIDACO can be extended to solve for multiple objective functions using parallelisation strategies and includes a local solver for solution refinement.

MIDACO is commonly used for optimisation in space applications ([54], [72]) and is still being developed and maintained. It offers a complete solution history file, containing all evaluated iterates and an interactive 2D & 3D plotting tool.

5.2.4 SNOPT

SNOPT [95] is another large scale NLP solver produced by Stanford University. It is based on SQP methods and interfaces with many applications and programming languages. Although it can still be purchased, and performs well in some tests it is unclear if any development or maintenance has taken place recently. Licenses are expensive, but there is a significant discount for academics.

5.3 Interplanetary Trajectory Optimisation Software

This section introduces the software that has been developed with a specific capability for solving ITO problems.

5.3.1 ASTOS

AeroSpace Trajectory Optimisation Software (ASTOS) [96], as the name suggests, this software is built almost specifically to aid in the calculation of optimal trajectories; though interplanetary trajectory optimisation is supported, much of the functionality of this software is geared around launch and re-entry trajectories.

ASTOS offers the standard patched conics framework for the problem requiring a fixed flyby sequence. Chemical and electric propulsion models are both supported.

Multiple solvers are incorporated within the software to solve trajectory optimisation problems, with additional options for warm and cold starts. Available algorithms come as part of the GESOP optimisation package:

- TROPIC and SOCS: Direct collocation methods,
- PROMIS Multiple shooting method,
- CAMTOS: Hybrid of collocation and multiple shooting methods,
- Genetic algorithms,

- Runge-Kutta (initial guess generator),
- SNOPT/WORHP/SLLSQP solvers.

There are mission analysis capabilities within the software that enable some data extraction, plotting and real-time monitoring of the optimisation process that could be useful.

ASTOS offers nothing special with regards to ITO problems, but it does provide a simple and easy package in which to begin analysis to find an optimal trajectory.

5.3.2 IGATO

Interplanetary Gravity Assist Trajectory Optimiser (IGATO) is an open source cross-platform application designed to intuitively tackle difficult trajectory optimisation problems involving MGA manoeuvres. IGATO was developed as part of an MSc thesis [97] and as such, has not been updated since 2012. However, the results of the software demonstrated in the thesis, suggest it would still be applicable to current ITO problems.

IGATO is an optimisation routine built around PaGMO that offers a graphical user interface, and additional capabilities in branching, pruning, restarting, subdomain decomposition and solution similarity testing. Surface stay return missions are also possible within this model. Although PaGMO is capable of multi-objective optimisation, this is not carried through as a feature of IGATO.

IGATO was shown to consistently perform as well or better than PaGMO alone for difficult MGA-DSM problems.

As with all methods discussed so far, the optimisation model behind this ITO problem is based on the usual patched conics models and the user must input a predetermined fly-by sequence.

IGATO also offers realtime plots to monitor the evolution of the optimal solution, 2D trajectory plots and Tisserand plots.

5.3.3 NASA in-house tools

NASA/JPL (Jet Propulsion Laboratory) have produced many tools and software packages to aid in the process of ITO. Some are open-source and free to use, but others remain in-house or only freely available for use by American institutions/those affiliated to NASA. The most applicable are presented below:

5.3.3.1 ChebyTop

The Chebyshev Trajectory Optimisation Program is named such as it uses Chebyshev polynomials to represent state variables. The program was originally constructed by Forrestor Johnson et al. at Boeing company in 1969 [98], but has subsequently been updated by Boeing, JPL and analysts at Glenn Research Center (GRC). The tool is intended to offer efficient trajectory design of one way trajectories between planets for low thrust missions. The tool is considered low-fidelity and only of use for preliminary analysis.

Chebytop uses Chebyshev optimisation method using Chebyshev polynomials for approximation. This simplifies the problem to that of ordinary calculus (as opposed to requiring a calculus of variations approach) and Newtons method can be implemented to optimise.

This tool is not without its limitations and struggles with particularly complicated interplanetary missions (multiple gravity assists and significant plane changes for example). The approximations are also poor for spacecraft with a high mass to thrust ratio.

The software and documentation can easily be downloaded from
https://spaceflightsystems.grc.nasa.gov/SSPO/ISPTProg/LTTT/

5.3.3.2 MALTO

MALTO [99] is a medium-fidelity mission analysis tool for low-thrust optimisation. The program uses a Matlab GUI and can cope with more complex interplanetary missions (using multiple gravity assists for example) than ChebyTop.

MALTO models low-thrust using many impulsive burns to simulate a continuous low thrust burn (Sims Flanagan method), this simplifies the problem and makes convergence easier. SNOPT is used for optimisation.

The tool is sold with commercial licensing, but may be freely available to academia.

5.3.3.3 Mystic

Mystic [100] was developed at JPL as a tool to use static/dynamic optimal control to perform non-linear optimisation of low-thrust trajectories. It is an n-body tool that includes gravity assists within the optimisation problem.

Mystic links to a Matlab GUI, providing ease of use for the user in both problem initiation and solution analysis.

5.3.3.4 OTIS

Optimal Trajectories by Implicit Simulation (OTIS) was developed by teams at GRC and Boeing. It was initially intended for launch vehicle trajectories but has since been expanded to include robust and accurate low thrust interplanetary mission analyses. Because of this, OTIS contains particularly sophisticated vehicle models and is considered a high fidelity optimisation and simulation tool.

OTIS uses implicit and explicit integration with analytic propagation, and SLSQP and SNOPT to solve the resulting NLP's.

OTIS is available to academia, though is subject to export control. A range of papers that make use of OTIS can be found here [101].

5.3.3.5 EMTG

The Evolutionary Mission Trajectory Generator (EMTG) was conceived as part of Jacob Englander's PhD thesis in 2013 [51] but has since been developed by NASA Goddard Space Flight Centre. It has evolved and improved and offers optimisation methods for interplanetary trajectories including multiple gravity assists, deep space manoeuvres and both low thrust and impulsive propulsion methods.

EMTG is a hybrid algorithm designed to solve multi-objective ITO problems using an integer program outer loop to calculate an optimal flyby sequence. The inner loop optimises the interplanetary trajectory given the flyby sequence. The outer loop consists of a genetic algorithm capable of variable chromosome length (as the number of flyby's is yet to be determined) and the inner loop uses SNOPT, a gradient based algorithm with monotonic basin hopping to optimise. Low thrust propulsion is modelled using the Sims Flanagan trajectory formulation.

EMTG is, at the time of writing, currently the only accessible software/tool found that optimises the flyby sequence in-line with the rest of the interplanetary trajectory. EMTG is considered to be medium fidelity and offers a good solution for preliminary analysis of a trajectory.

5.4 Analysis Tools

As interplanetary trajectory optimisation is such a complex multi-objective problem, it is unrealistic to optimise with respect to all possible objectives and constraints due to computation times. Instead, a simplified model is used to construct a preliminary trajectory with the intention of assessing its accuracy and feasibility later using analysis tools.

Analysis tools are high fidelity and use local optimisation methods to find the optimal trajectory with this additional information. They are usually the final stage of the trajectory optimisation/design process and are mostly considered high precision enough for operational use.

The following programs are analysis tools. Their primary purpose is to analyse the feasibility of a mission, including a previously determined trajectory to check the practicality of the route; Many additional models are added to the problem at this point such as aerodynamics, satellite coverage, the effects of gravitational perturbations and solar radiation pressure.

5.4.1 STA-Space Trajectory Analysis

In this paper, *Space Trajectory Analysis (STA) "Silurean" v5.0*, test release 1 (February 2016) is considered. [102]

The STA software suite's main aim is to support and enable the analysis of a space mission through its capabilities in trajectory analysis, determination, simulation, and visualisation of a variety of space trajectories- focussing on re-entry trajectories, orbits and interplanetary missions.

STA allows considerable analysis through many selectable parameters. The following are some examples:

- Trajectory- Times, positions, spacecraft attitude etc.
- Aerodynamics and re-entry- Temperature, pressure, deceleration, mach number, flight path angle etc.
- Coverage and communication- Access time, range, azimuth, elevation
- SEM data- payload elements, thermal subsystem, structural subsystem elements, telemetry etc.

ITO requires the acquisition of an optimal route for an interplanetary mission involving multiple gravity assists. The analysis offered by this tool allows some simplification of the problem, as the analysis can be carried out after a trajectory has been determined and not added to the optimisation model as an additional constraint.

STA contains an "interplanetary" tool, which allows the calculation of interplanetary routes given the following input data:

- ordered list of flyby planets,
- objective function (deltaV or ToF or simple combination of both),
- EPOCH window,
- maximum C3 for the mission.

Although a useful feature, especially when it offers such aesthetically pleasing visuals, it requires some effort to input any meaningful parameters. Prior analysis must be completed in order to determine an appropriate ordered fly-by list, smaller launch EPOCH window and to choose an upper limit for C3. The methodology is based on patched conics.

The solvers used for this optimisation are unclear as the documentation with the software is not exhaustive.

STA has some difficulty with more complex tours (such as moon tours, containing many fly-by's) as it struggles to accurately compute such a complex trajectory. The software can still be used for the simulation however as the trajectory arcs can be input by hand, though the fly-by's remain as discontinuities (as often occurs with simple patched conic models). it would appear that inputting the trajectory arcs by hand is the best choice for the simulation, however using STA for proper analysis after this would be difficult given that the fly-by phases could not be included in this analysis.

5.4.2 GMAT

The General Mission Analysis Tool (GMAT) [103] is developed and maintained by NASA, Private Industry and public and private contributors. It's main purpose is to model and optimise space missions between orbits. It does not, at present, offer any optimisation or modelling capabilities for launch or re-entry.

The fundamental capability of GMAT is its ability to propagate Spacecraft trajectories. This allows for accurate simulation, analysis and local optimisation of interplanetary trajectories. A mission can be input as a series of epochs (arrivals and departures from planets, DSM's GA's etc.) based on a preliminary trajectory calculated from a lower fidelity model. Local optimisation (multiple shooting) is then used to optimise the trajectory in GMAT's higher fidelity framework.

As GMAT is a tool for general mission analysis, it is capable of much more sophisticated analysis. For example, GMAT can model the attitude of a spacecraft in addition to other useful models embedded within the tool.

GMAT is considered to be a very high fidelity tool and the analysis and resulting calculations are certified by NASA to be used in mission operations.

GMAT is extremely well maintained and documented. As the tool is so well used and supported, it is clear that it will continue to improve over the coming years. The ”Modular design” means updating GMAT is easy and many different features and options for analysis can continue to be added and updated.

GMAT also aims to make use of parallel processing, allowing solvers to solve quickly across multiple processors. GMAT is designed so that in the future it may utilise parallel computing, however it doesn't currently make use of this feature.

5.4.3 Freeflyer

FreeFlyer [104] is a commercial software for space mission design, analysis, and operations. It has the capability to support mission design throughout its lifecycle from preliminary analysis right through to mission operations. Freeflyer is not a tool specifically for interplanetary missions, but does offer some capability in this area using some built in optimisers and targeters.

FreeFlyer has been independently verified and validated for flight-testing, proven accuracy, and is used for spacecraft analysis and operations by NASA, NOAA,USAF, DoD, and other commercial satellite providers.

License acquisition for this product proves difficult, but its wide commercial use implies a well developed tool, at least with regards to satellite missions. Although the software would seem to provide support for interplanetary mission analysis it is clearly not the primary focus of this software so is likely to be underdeveloped as with most other commercial software of this type.

5.4.4 Copernicus

Copernicus [105, 106] is a generalised spacecraft trajectory design and optimisation system, capable of solving a wide range of trajectory problems, including ITO.

It's biggest advantage is its flexibility. Many software are limited to only specific trajectory design problems, but Copernicus is problem agnostic. This is particularly useful for someone that designs many different types of mission trajectories, however is somewhat unnecessary for pure ITO.

Copernicus is a high fidelity solution based on a numerical integration procedure for solving multi-point boundary and non-linear problems. The problems are defined and input by the user in segments.

The software has a GUI, but requires considerable human input to setup the problems. Copernicus has a NASA webpage that displays regular updates [106].

5.5 Conclusion

Analysing the models behind existing software packages is not always a simple task; Many tools do not provide sufficient documentation to fully understand their inner workings and using them is not adequate to fully describe how they function.

The current approaches to ITO consist of many stages, each of which can benefit from the use of toolboxes or software applications. There are a number of these available, though commercial drivers often dictate the prioritised software capabilities.

More specialist tools have been developed for ITO (namely PyKep, IGATO, EMTG and other NASA tools) however these are either funded and supported by scientific/academic organisations (ESA/NASA/Universities) or fall quickly into a state of neglect as a small market fails to pay for their development.

In general, the software and toolboxes outlined in this chapter enable ease of access to solvers and propagators. This helps to enable further research into the best algorithms and optimisers to solve the problem. The addition of low and high fidelity purpose built software enables easier identification and refinement of trajectories for missions today. Understandably, the software are often delayed in their incorporation of the latest advancements in global optimisation methodology as it takes considerable effort to keep them up to date.

The process of solving ITO problems is time consuming, with many separate analyses, models and decisions required. Software could be improved to better combine and automate the flow through the stages- perhaps combining low and high fidelity models into one software, with a simple capability to pass the output of the low fidelity model, to the higher fidelity model.

Chapter 6

Conclusions and Opportunities for Further Advancement

It is clear that in the last 2 decades, capabilities in the global optimisation of interplanetary trajectories have advanced.

Generally, understanding of global optimisation problems and their search spaces has improved, allowing for a surge in global optimisation algorithm development. Monte Carlo Tree Search and hybrid methods stood out as the most effective. MCTS for its inclusion of the flyby sequence into the problem and its ability to prune the search space as it constructs solutions and hybrid methods for their ability to consistently out-perform individual algorithms.

Search space reduction has also been a priority in the literature. GASP [75] introduced the logic to reduce a search space to a purely feasible one, though further reduction to eliminate areas unlikely to contain good solutions has been slower. Machine learning has been useful in this area. The classification of start points as seen in LeGo [83] successfully prioritises start points for global search algorithms, though cannot offer an absolute reduction in search space size.

Machine learning has also benefitted the problem through approximation of the computationally expensive objective function. The computational cost of a Lambert solver has reduced, though this was not adequate and was bettered by the approximation which not only reduced computational complexity, but also smoothed the search space, reducing the number of local optima.

More specifically for ITO problems, the complexity reduction for Lambert solvers enabled faster function evaluations, allowing for more extensive searches of the search space.

Software and Toolboxes are available to increase the accessibility of the problem, however they remain a few steps behind the literature, and continue to require considerable human input.

6.1 Remaining Challenges and Opportunities

As introduced in 4.2 The main challenges for ITO are:

- The computational cost of objective function evaluations,
- The size and multimodality of the search space,
- The requirement for good initial solutions,
- The manual/separate selection of combinatorial elements of the problem,
- Solution Robustness.

Many of the methods discussed in this thesis go some way to remove the obstacles outlined above though none are completely disappeared.

There remains significant opportunity for further improvement in each of the areas outlined above. More specifically, overcoming these challenges will enable a solution to the full ITO problem.

6.1.1 The Full ITO problem

The problem of interplanetary trajectory optimisation is rarely attempted in its fullest form in the literature. Most commonly considered are sub-problems with fixed flyby sequences, limited launch windows, fixed spacecraft characteristics constrained propulsion.

However, [68, 51] successfully incorporated the flyby sequence into the optimisation problem and the software copernicus [105] showed that multiple trajectory problems could be included in the same modelling framework.

Still to be incorporated are allowances for all propulsion types within the same formulation, both flyby models, and high-fidelity modelling from the beginning.

Mathematics is not holding back progress to the full ITO problem, computational complexity and run-time is. The simplest way to solve this issue is to reduce computational complexity, increase algorithm efficiency or reduce the size of the search space. The following sections outline possible methods for this.

6.1.1.1 Parameter Optimisation

Almost all global search and machine learning algorithms have control parameters, and the settings can dramatically change the algorithm's performance. Parameters can impact the algorithms ability to locate a good solution, or simply how quickly it got there. For some algorithms, it is possible to use existing problem understanding and knowledge of the algorithms to set parameters appropriately in the first instance.

However, for more complex algorithms, optimal parameter choices are less clear, and can vary a lot by problem. For example, Evolutionary Algorithms (4.4.1.3) are dependent on multiple parameters and choices; What is the population size? What is the crossover operator? How will mutations be applied? etc. Whilst some knowledge and context of the parameters will help to reduce the list of possible parameter combinations, finding the optimal combination is not an easy task. The topic is discussed more extensively in the literature [107], [108], [109].

Almost all papers cited in this thesis that made use of algorithms with parameters noted that they did not explicitly tune their parameters. Whilst there is of course no guarantee that parameter tuning will yield improved results over the default or intuitively chosen parameters currently used, no literature could be found to prove the contrary.

Parameter tuning is standard practice for modern Data Scientists and there are open source packages available for commonly used classification and regression models. One of the most common of these is the R package CARET [110]. The main features of CARET are to evaluate, using resampling and cross-validation, the effect of model tuning parameters on performance and to choose the optimal model parameters across these parameters. A similar methodology could be developed and applied to global optimisation search algorithms, using a small sample of the problem at hand.

The analysis completed in [5] could provide useful indication of starting points for a parameter optimisation algorithm, depending on the search algorithm chosen. For example, using knowledge of the distance between local optima may help to inform the magnitude of a perturbation parameter capable of jumping to the next local optimum, instead of jumping over it. Similarly, understanding the likely number of nested optima could help to inform the magnitude of a parameter needed to prevent it from converging too early.

Additionally, the analysis in [5] specifically related to the distances between groups of nested optima could help to inform appropriate distances between starting points for search algorithms to reduce the number of iterations required for convergence.

6.1.1.2 Machine Learning

Machine learning can be particularly useful in problems where there are many variables with complex interactions. They often far surpass the capabilities of humans to quickly understand the patterns and drivers of a particular outcome. Given the cyclical motion of planets, and predictable dynamics of spacecraft in space, it seems that ITO problems should be a successful application for machine learning.

6.1.1.3 Machine Learning for Search Space Reduction and Understanding

GASP reduces the search space to a purely feasible one with reasonable cost in polynomial complexity [75]. The next step would be the ability to remove parts of the search space that are highly unlikely to contain good solutions.

Building on the method initially proposed in LeGo [**?**]; A classification model could be used to classify individual solutions. Tree based classification algorithms could be used with the benefit that their output provides actionable descriptions of regions of a search space that can be pruned or prioritised as relevant.

Data to train the classification model could be collected in a similar way to LeGo, using the iterations of a global search algorithm such as differential evolution. An improvement to create a less biased training data set would be to parameterise agents differently. Half of the agents should look to minimise the objective function, whilst the remaining half look to maximise it. This will help to better distinguish the features of a solution that drive it to be good or bad.

Analysis on this representative data set, can provide a better understanding of the search space landscape in each region. For example, the following metrics could be used to evaluate and prioritise parts of the search space:

- Descriptive statistics on the objective function value (Mean, standard deviation, minimum, maximum,range, interquartile range etc.) ,
- Ratio of good/bad solutions,
- Distances between local optima (inspired by the metrics in [5].

6.1.1.4 Machine Learning for Global Interplanetary Trajectory Approximation

Machine learning has been successfully implemented to predict the objective function for ITO problems [80, 82, 81]. An evolutionary algorithm or other global search algorithm can then be used to search the approximated search space with much cheaper objective function evaluations.

An ambition here is to expand Izzo et al's work to build a model capable of predicting the objective function to a sufficient accuracy given any parameters for the full ITO problem (including changing the spacecraft and target bodies). In the future, this would serve as a central intelligence hub for space missions that could be consulted and continually updated with all calculated interplanetary trajectories. The more missions that are planned and evaluated, using different spacecraft specifications and target bodies, the better the search space approximation will become for the full ITO problem.

Currently, each objective function evaluation is stored and used only in the specific mission it was evaluated for. However, unlike most environments where data is captured, there aren't many external factors in Space that are unaccounted for, so old data is still good data. Storing the costs for each individual leg of a trajectory would enable an approximation agnostic to the constraints of the problem.

Developing this ITO approximation model would take time and a lot of data, but this is data that is/has already been generated for missions that are/have already been planned. It is also possible to generate training data for the model for missions that there are no current plans to undertake. There are no measurements involved in collecting this data, only expensive function evaluations. This could be outsourced using idle machines as successfully demonstrated by SETI@home [111] who use internet-connected computers in the Search for Extraterrestrial Intelligence (SETI). Anyone with a computer can participate by running a free program that downloads and analyses radio telescope data, sending the results back to the chief analysts.

The model would need constant training to keep up with technological advancements in spacecraft engineering, but it may also inspire some; Understanding feature importance and drivers of good/bad trajectories could help to direct and prioritise the developments that are likely to have the most impact.

The approximation could remove the need for low fidelity models if trained on data from high fidelity models.

6.2 Closing Remarks

This thesis has introduced the problem of ITO with relevant models and equations. It has also presented the many challenges that prevent a simple solution. Literature was presented describing the existing solution methods and available toolboxes and software, in addition to their strengths and weaknesses. Finally, recommendations for further work are made, describing their potential benefits.

Bibliography

[1] [Online]. Available: https://en.wikipedia.org/wiki/Orbital$_e$*lements/media/File* : *Orbit*1*.svg*

[2] [Online]. Available: http://mathworld.wolfram.com/EccentricAnomaly.html

[3] N. J. Stange and J. M. Longuski, "Graphical method for gravity-assist trajectory design," *Journal of Spacecraft and Rockets*, vol. 39, no. 1, pp. 9–16, Jnuary-February 2002.

[4] M. Vasile and M. Ceriotti, "Incremental techniques for global space trajectory design," *Spacecraft Trajectory Design*, pp. 202–237, August 2010.

[5] M. Vasile, E. Minisci, and M. Locatelli, "Analysis of some global optimisation algorithms for space trajectory design," *Journal of Spacecraft and Rockets*, pp. 334–344, 2010.

[6] J. Kepler, *Astronomia Nova*, 1609.

[7] I. Newton, *Philosophiae Naturalis Prinicipa Mathematica*, 1687, vol. 1.

[8] NASA, "Juno's two deep space maneuvers are 'back-to-back' home runs." [Online]. Available: https://www.nasa.gov/mission$_p$*ages/juno/news/juno*20120917*.html*

[9] J. A. Sims and S. N. Flanagan, "Preliminary design of low-thrust interplanetary missions," in *AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, Alaska, 1999.

[10] O. Schütze, M. Vasile, O. Junge, M. Dellnitz, and D. Izzo, "Designing optimal low-thrust gravity- assist trajectories using space pruning and a multi-objective approach," *Engineering Optimization*, vol. 41, no. 2, pp. 155–181, January 2009.

[11] A. E. Petropoulos and J. M. Longuski, "Shape-based algorithm for automated design of low-thrust, gravity-assist trajectories," *Journal of Spacecraft and Rockets*, vol. 41, no. 5, pp. 787–796, Seeptember-October 2004.

[12] ESA, "Ikaros (interplanetary kite-craft accelerated by radiation of the sun)." [Online]. Available: https://earth.esa.int/web/eoportal/satellite-missions/i/ikaros

[13] N. J. P. Laboratory, "Voyager factsheet." [Online]. Available: https://voyager.jpl.nasa.gov/frequently-asked-questions/fact-sheet/

[14] J. C. Niehoff, "Gravity-assisted trajectories to solar-system targets," *Journal of Spacecraft and Rockets*, vol. 3, no. 6, pp. 1351–1356, September 1966.

[15] R. B. Adamas and G. A. Richardson, "Usng the two-burn escape maneuver for fast transfers in the solar system and beyond," *NASA*.

[16] H. Oberth, "Ways to spacefilght," *Agence Tunisienne de Public Relations*, 1970.

[17] V. I. Arnold, *Mathematical Methods of Classicl Mechanics*, 2nd ed. New York: Springer-Verlag, 1989.

[18] W. Hohmann, "Die erreichbarkeit der himmelskorper (the possibility of reaching the stars), verlag r," 1925.

[19] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*, ser. Education Series. AIAA, 1999.

[20] E. R. Lancaster and R. C. Blanchard, "A unified form of lambert's theorem," *NASA Technical Note, Goddard Spaceflight Center*, September 1969.

[21] D. Izzo, V. Becerra, D. Myatt, S. J. Nasuto, and J. M. Bishop, "Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories," *Journal of Global Optimisation*, vol. 38, pp. 283–296, 2007.

[22] R. H. Gooding, "A procedure for the solution of lambert's orbital boundary-value problem," *Celestial Mechanics and Dynamical Astronomy*, vol. 48, no. 2, pp. 145–165, 1990.

[23] M. V. P. De Pascale, "Preliminary design of low-thrust multiple gravity-assist trajectories," *Journal of Spacecraft and Rockets*, vol. 43, no. 5, pp. 1065–1076, 2006.

[24] J. V. Breakwell, "Researches in interplanetary transfer," *ARS Journal*, vol. 31, no. 2, pp. 201–208, 1961.

[25] F. Peralta and S. Flanagan, "Cassini interplanetary trajectory design," *Control Engineering Practice*, vol. 3, no. 11, pp. 1603–1610, 1995.

[26] A. E. Petropoulos, J. M. Longuski, and E. P. Bonfiglio, "Trajectories to jupiter via gravity assists from venus, earth and mars," *Journal of Spacecraft and Rockets*, vol. 37, no. 6, pp. 776–783, Nov-Dec 2000.

[27] L. D'Amario, D. Byrnes, and R. Stanford, "A new method for optimizing multiple-flyby trajectories," *Journal of Guidance and Control*, vol. 4, no. 6, pp. 591–596, Nov-Dec 1981.

[28] L. D'Amario, D. Byrnes, and R. H. Stanford, "Interplanetary trajectory optimization with application to galleo," *Journal of Guidance*, vol. 5, no. 5, pp. 465–471, Sept-Oct 1982.

[29] O. von Stryk and R. Bulirisch, "Direct and indirect methods for trajectory optimization," *Annals of Operational Research*, pp. 357–373, 1992.

[30] T. Vinko, D. Izzo, and C. Bombardelli, "Benchmarking different global optimisation techniques for preliminary space trajectory design," in *58th International Astronautical Congress*, September 2007.

[31] "Gtoc." [Online]. Available: https://sophia.estec.esa.int/gtoc$_portal/$

[32] C. F. Gauss, *Theory of the Motion of the Heavenly Bodies Moving About the Sun in Conic Sections*, 1857.

[33] E. R. Lancaster, R. C. Blanchard, and R. Devaney, "A note on lambert's theorem," *Journal of Spacecraft and Rockets*, vol. 3, no. 9, pp. 1436–1438, 1966.

[34] R. H. Battin, "Lambert's problem revisited," *American Institute of Aeronautics and Astronautics*, pp. 18–20, July 1976.

[35] R. H. Battin and R. M. Vaughan, "An elegant lambert algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 7, no. 6, pp. 662–670, 1984.

[36] R. H. Battin, T. Fill, and S. Shepperd, "A new transformation invariant in the orbital boundary problem," *Journal of Guidance*.

[37] L. A. Loechler, "An elegant lambert algorithm for multiple revolution orbits," Master's thesis, Massachussetts Institute of Technology, May 1988.

[38] R. H. Gooding, "On the solution of lambert's orbital boundary value problem," ROYAL AEROSPACE ESTABLISHMENT FARNBOROUGH, Tech. Rep., 1988.

[39] S. L.Nelson and P. Zarchan, "Alternative approach to the solution of lambert's problem," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 4, pp. 1003–1009, 1992.

[40] D. Izzo, "Lambert's problem for exponential sinusoids," Advanced Concepts Team, ESA/ESTEC, Noordwijk, The Netherlands, ACT/ESTEC INTERNAL REPORT ACT-RPT-4100-DI-LMSP01, April 2005.

[41] G. Avanzini, "A simple lambert algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 6, pp. 1587–1594, 2008.

[42] M. Bando and H. Yamakawa, "New lambert algorithm using the hamilton-jacobi-bellman equation," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 3, pp. 1000–1008, 2010.

[43] D. Izzo, "Revisiting lambert's problem," *Celestial Mechanics and Dynamical Astronomy*, vol. 121, pp. 1–15, 2015.

[44] D. de la Torre Sanga and E. Fantino, "Review of lambert's problem," in *25th international symposium on space flight dynamics*, October 2015.

[45] A. R. Klump, "New developments in astrodynamics algorithms for autonomous rendezvous," Jet Propulsion Laboratory, Pasadena, California, Tech. Rep., November 1991.

[46] N. Parrish, "Accelerating lambert's problem on the gpu in matlab," June 2012.

[47] B. A. Conway, Ed., *Spacecraft Trajectory Optimization.* Cambridge, 2010.

[48] D. Wales and J. Doye, "Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms," *The Journal of Physical Chemistry*, vol. 101, no. 28, pp. 5111–5116, 1997.

[49] S. L. McCarty and M. L. McGuire, "Parallel monotonic basin hopping for low thrust trajectory optimization," in *AIAA SciTech Forum,*, Kissimmee, Florida, January 2018.

[50] J. A. Englander and A. C. Englander, "Tuning monotonic basin hopping: improving the efficiency of stochastic search as applied to low-thrust trajectory optimization," in *24th International Symposium on Space Flight Dynamics*, Laurel, MD, May 2014.

[51] J. Englander, "Automated trajectory planning for multiple-flyby interplanetary missions," Ph.D. dissertation, University of Illinois, Urbana- Champaign, 2013.

[52] J. A. Englander, D. H. Ellison, and B. A. Conway, "Global optimisation of low-thrust, multiple flyby trajectories at medium and medium-high fidelity," in *AAS/AIAA Space-flight mechanics meeting*, Santa Fe, NM, January 2014.

[53] C. Yam, D. di Lorenzo, and D. Izzo, "Constrained global optimization of low-thrust interplanetary trajectories," *IEEE Congress on evolutionary computation*, 2010.

[54] D. Izzo, M. Martens, and B. Pan, “A survey on artificial intelligence trends in spacecraft guidance dynamics and control,” *Astrodynamics*, vol. 3, pp. 287–299, 2019.

[55] R. Storn and K. Price, “Differential evolution- a simple and efficient heursitc for global optimisation over continuous spaces,” *Journal of Global Optimisation*, no. 11, pp. 341–359, 1997.

[56] D. R. Myatt, V. M. Becerra, S. J. Nasuto, and J. M. Bishop, “Advanced global optimisation for mission analysis and design,” Department of Cybernetics, University of Reading, Final Report Ariadna ID 04/4104, 2004.

[57] A. D. Olds and C. A. Kluever, “Interplanetary mission design using differential evolution,” *Journal of Spacecraft and Rockets*, vol. 44, no. 5, September 2007.

[58] D. Izzo, L. F. simoes, M. Martens, G. de Croon, A. Heretier, and C. H. Yam, “Search for a grand tour of the jupiter gallilean moons,” in *15th annual conference on Genetic and Evolutionary computation*, July 2013, pp. 1301–1308.

[59] W. Yao, J. Luo, M. Macdonald, M. Wang, and W. Ma, “An improved differential evolution algorithm and its applications to orbit design,” *Journal of Guidance, Control and Dynamics*, November 2017.

[60] M. Vasile, M. Locatelli, and E. Minisci, “An inflationary differential evolution algorithm for space trajectory optimisation,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 2, pp. 267–281, 2011.

[61] F. A. Zotes and M. S. Penas, “Particle swarm optimisation of interplanetary trajectories from earth to jupiter and saturn,” *Engineering Applications of Artificial Intelligence*, vol. 25, no. 1, pp. 189–199, February 2012.

[62] B. A. Conway and M. Pontani, “Particle swarm optimisation applied to space trajectories,” *Journal of Guidance, Control and Dynamics*, vol. 33, no. 5, pp. 1429–1441, September 2010.

[63] M. Vasile, “A behavioural based meta-heuristic for robust global trajectory optimization,” *Evolutionary Computation*, pp. 2056–2063, 2007.

[64] F. Glover, “Future paths for integer programming and links to atificial intelligence,” *computers and operations research*, vol. 13, no. 5, pp. 533–549, 1986.

[65] G. P. Johnson, “A tabu search methodology for spacecraft tour trajectory optimisation,” Ph.D. dissertation, University of Texas, 2014.

[66] M. R. Sentinella and L. Casalino, “Enhanced continuous tabu search in a hybrid evolutionary algorithm for the optimization of interplanetary trajectories,” *Journal of Spacecraft and Rockets*, vol. 46, no. 2, March-April 2009.

[67] F. Simeoni and L. Casalino, “Evolutionary optimization of interplanetary trajectories: improvements from initial diversification,”,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 225, no. 11, pp. 1277–1288, 2011.

[68] R. Coulom, “Efficient selectivity and backup operators in monte-carlo tree search,” in *5th international conference on computer and games*, Turin, Italy, May 2006.

[69] D. Hennes and D. Izzo, “Interplanetary trajectory planning with monte carlo tree search,” in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015, pp. 769–775.

[70] J. A. Englander, B. A. Conway, and T. Williams, “Automated mission planning via evolutionary algorithms,” *Journal of Guidance, Control and Dynamics*, vol. 35, no. 6, pp. 1878–1887, November 2012.

[71] M. R. Sentinella and L. Casalino, “hybrid evolutionary algorithm for the optimisation of interplanetary trajectories,” *Journal of Spacecraft and Rockets*, vol. 46, no. 2, March-April 20019.

[72] M. Schlueter, "Nonlinear mixed integer based optimization technique for space applications," Ph.D. dissertation, University of Birmingham, May 2012.

[73] T. Grain, R. H. Bishop, W. Fowler, and K. Rock, "Optimal interplanetary trajectory design via hybrid genetic algorithm/recursive quadratic program search," *Advances in the Astronautical Sciences*, pp. 449–466, December 2019.

[74] M. Vasile and P. de Pascale, "preliminary design of multiple gravity assist trajectories," *Journal of Spacecraft and Rockets*, 2006.

[75] V. Becerra, D. Myatt, S. Nasuto, J. Bishop, and D. Izzo, "An efficient pruning technique for the global optimisation of multiple gravity assist trajectories," *Acta Futura*, pp. 35–42, 2003.

[76] J. T. Olympio and J.-P. Marmorat, "Global trajectory optimisation: Can we prune the solution space when considering deep space maneuvers?" ARIADNA Final Presentation, September 2007.

[77] P. D. Lizia, R. Armellin, F. Topputo, F. Bernelli-Zazzera, M. Berz, and M. Lavagna, "Gravity assist space pruning based on differential algebra," *Celestial Mechanics and Dynamical Astronomy*, vol. 106, no. 1, pp. 1–24, 2010.

[78] B. Dachwald, "Optimization of very-low-thrust trajectories using evolutionary neurocontrol," in *55th International Astronautical Congress*. German Aerospace Centre (DLR), 2004.

[79] I. Carnelli, B. Dachwald, and M. Vasile, "Evolutionary neurocontrol: A novel method for low-thrust gravity assist trajectory optimisation," *Journal of Guidance, Control and Dynamics*, vol. 32, no. 2, pp. 615–624, March-April 2009.

[80] C. Ampatzis and D. Izzo, "Machine learning techniques for approximation of objective functions in trajectory optimisation," in *Artifical Intelligence in Space (IJCAI-09 Workshop)*, vol. 673. Advanced Concepts Team, European Space Agency, September 2009.

[81] D. Hennes, D. Izzo, and D. Landau, “Fast approximators for optimal low-thrust hops between main asteroid belts,” in *IEEE Symposium Series on Computational Intelligence SSCI)*, 2016.

[82] A. W. Alessio Mereta, Dario Izzo, “Machine learning of optimal low-thrust transfers between near-earth objects,” in *International Conference on Hybrid Artificial Intelligence Systems*, 2017, pp. 543–553.

[83] A. Cassioli, D. di Lorenzo, M. Locatelli, F. Schoen, and M. Sciandrone, “Machine learning for global optimization,” *Computational Optimization and Applications*, vol. 51, no. 1, pp. 279–303, January 2012.

[84] D. Izzo, C. Sprague, and D. Tailor, *Modeling and Optimization in Space Engineering.* Springer, September 2018, ch. Machine Learning and Evolutionary Techniques in Interplanetary Trajectory Design, pp. 191–210.

[85] “Pykep.” [Online]. Available: https://esa.github.io/pykep/

[86] “Tudat.” [Online]. Available: http://tudat.tudelft.nl/

[87] P. Musegaas, “Optimization of space trajectories including multiple gravity assists an deep space maneuvers,” Master’s thesis, Tu Delft, 2013.

[88] L. Hoving, “Optimisation strategies for galilean moon tours: Low-thrust multiple gravity assist trajectory design for gtoc6,” Master’s thesis, Tu Delft, 2015.

[89] “Poliastro software.” [Online]. Available: https://github.com/poliastro/poliastro

[90] “Orekit.” [Online]. Available: https://www.orekit.org/

[91] “Pygmo/pagmo.” [Online]. Available: https://esa.github.io/pagmo2/

[92] F. Biscano, D. Izzo, and C. H. Yam. (2018, October) A global optimisation toolbox for massively parallel engineering optimisation. [Online]. Available: arxiv.org/pdf/1004.3824.pdf

[93] “Worhp.” [Online]. Available: https://worhp.de/

[94] "Midaco." [Online]. Available: http://www.midaco-solver.com

[95] "Snopt." [Online]. Available: https://ampl.com/products/solvers/solvers-we-sell/snopt/?gclid=CjwKCAjwmtDpBRAQEiwAC6lm4-0bJpu3ueRT5gCt4nCYDR-doSobsmSDBOkiD4A4$_3 7SyNexIfe3nRoCwOkQAvD_BwE$

[96] "Astos." [Online]. Available: https://www.astos.de/

[97] J. M. Bryan, "Global optimisation of mga-dsm problems using the interplanetary gravity assist trajectory optimizer (igato)," Master's thesis, California Polytechnic State University, San Luis, Obispo, 2011.

[98] F. Johnson, "Approximate finite-thrust trajectory optimization," *Aiaa Journal - AIAA J*, vol. 7, pp. 993–997, 06 1969.

[99] "Malto." [Online]. Available: https://software.nasa.gov/software/NPO-43625-1

[100] G. J. Whiffen, "Mystic: Implementation of the static dynamic optimal control algorithm for high-fidelity, low-thrust trajectory design," in *AIAA/AAS Astrodynamics Specialists Conference*, August 21-24 2006.

[101] "Otis." [Online]. Available: https://otis.grc.nasa.gov/references.html

[102] "Sta." [Online]. Available: https://github.com/toxygen/STA

[103] [Online]. Available: https://sourceforge.net/projects/gmat/files/GMAT/GMAT-R2011a/

[104] [Online]. Available: https://uk.mathworks.com/products/connections/product$_detail/freeflyer.html$

[105] G. Johnson and S. Munoz, "Copernicus, a generalised trajectory design and optimisation system," Department of Aerospace Engineering and Engineering Mechanics, University of Texas, Tech. Rep., December 2003.

[106] (2018). [Online]. Available: https://www.nasa.gov/centers/johnson/copernicus/index.html

[107] S. K. Smit and A. E. Eiben, "Comparing parameter tuning methods for evolutionary algorithms," in *IEEE Congress on Evolutionary Computation*. Trondheim, Norway: IEEE, May 2009.

[108] H. H. Hoos, *Autonomous Search.* Springer, 2011, ch. Automated Algorithm Configuration and Parameter Runing, pp. 37–71.

[109] R. H. Margarita Antoniou and G. Papa, *Optimization Under Uncertainty with Applications to Aerospace Engineering.* Springer, 2020, ch. Parameter Control in Evolutionary Optimisation.

[110] M. Kuhn. Classification and regression training. [Online]. Available: https://cran.r-project.org/web/packages/caret/caret.pdf

[111] U. of Berkeley, "Seti@home." [Online]. Available: https://setiathome.berkeley.edu/