# Design of executable space mission architectures using network flow optimization

by

Olivier I. Cornes

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Diplôme d'ingénieur Sup'Aéro

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

October 2016



Author .....
Department of Aeronautics and Astronautics
26 october, 2016

Certified by .....
Prof. Olivier L. De Weck
Professor of Aeronautics and Astronautics and Engineering Systems
Thesis Supervisor

Accepted by .....
Prof. Stéphanie Lizy-Destrez
Professor of Space Systems at ISAE-Sup'Aéro
Sup'Aéro Advisor

**Design of executable space mission architectures using network flow optimization**

by

Olivier I. Cornes

Submitted to the Department of Aeronautics and Astronautics
on 26 october, 2016, in partial fulfillment of the
requirements for the degree of
Diplôme d'ingénieur Sup'Aéro

## Abstract

As humans explore Space, the demands for space logistics and infrastructure has increased dramatically since the pioneering suborbital flights: because space mission architecture and planning has become far more complex than simply sending a crewed mission to its destination with everything on board like Apollo, new methods have to be devised in order to address the problems of long duration space flight. One of these is the use of linear multi-commoditiy network flows to optimize simultaneously the mission planning and the vehicles and infrastructure within it. However, this method has shortcomings concerning the fidelity of the models because of its linear network nature. In this work, these problems are addressed and solved using mixed integer linear programming and a tracking algorithm. The case study considered is a manned exploration mission to Mars with identical requirements to DRA5.0 using NTR, ISRU, and aerobracking. An architecture of this mission is obtained using this optimization method. It is found that lunar ISRU only reduces the IMLEO by 5% compared to a direct Mars mission. Lunar ISRU is found to be ineffective at reducing IMLEO below the critical value of 3kilograms of propellant mined per kilogram plant per year. Removing NTR and aerocapture technologies is found to increase IMLEO by 33.0 and 27.5% respectively with respect to the baseline scenario using both technologies.

Thesis Supervisor: Prof. Olivier L. De Weck
Title: Professor of Aeronautics and Astronautics and Engineering Systems

## Acknowledgments

This thesis is the conclusion of an intense 6 year adventure during which I did my studies at EPFL, Sup'Aéro and now MIT to realize my dream of becoming both a Mechanical Engineer and an Aerospace Engineer. This extraordinary path could not have been followed without the people that accompanied me along the way:

First I would like to thank my advisor at MIT, Prof. Olivier De Weck, for seeing potential in me and and inviting me to join the Strategic Engineering Group at MIT. I am thankful to him for giving me the opportunity of traveling in the United States to conferences and other laboratories, and giving me the responsibility of contributing to a research topic that is important to the laboratory and has great potential. I could not thank Olivier without forgetting Julie, the secretary and nervous system of the laboratory, who greatly contributed in helping me overcome the administrative beast.

Going to MIT would not have been possible without the help of Muriel Richard, who facilitated the contact to MIT. Muriel led the teams of engineers that built the first entirely swiss satellite. Her class on space system engineering has greatly contributed to my interest for space systems and I still regard it as being the best systems engineering course I have had. Her enthusiasm and resourcefulness and will continue to inspire me.

Inspiration and motivation are critical, especially during difficult times: thanks to their friendship and caring, Sara Lopez-Sanchez, Pierre-François Conzelmann, Thomas Wipf, Bertil Trottet and Ambroise Laurent kept me afloat during my time at Sup'Aéro.

Doing the EPFL-Sup'Aéro double-degree is a prestigious opportunity that I would not have obtained had it not been for professors and researchers at EPFL who saw potential in me before I could, and encouraged me to keep going: in particular, I would like to thank Claude Ramseyer for having believed in my capabilities and offering me my first job as a T.A. at EPFL, Dr. Joel Cugnoni and Prof. Thomas Gmür for their guidance and help in my research projects and classes at EPFL.

Then there are the people in Boston, such as the 33-409 crew, especially Narek Shougarian, Marc Sanchez, Sydney Do and Andrew Owens for their advice in being a grad student at MIT and conducting research in aerospace systems engineering. Although I spent much of my time at MIT itself, Boston was made enjoyable by the time I spent outside MIT with my friends and roomies, such as Alix, Constance, Ola, Pablo, Alex, Elena, Bastien, Cosima,

Anna, Laura, Jérôme, Patrick, Saviz, Linus and Ronnie.

My studies as an engineer would not have come to a successful end without my homies from the Morges-Coast: Jérôme Amiguet, Mike Kalomeni, the Poffet brothers, David Wuthier, Olivier Mila, Matt Béguin, Ricc, Chris, Ben, Lydiane and a few more that I should mention.

The final thanks goes to my close family: Mum and Dad, who have put an enormous effort into their children, even by parents standards. Patrick and Josi, who have saved me from the fate of growing up alone, and have made sure our household be an entertaining one.

# Contents

**1 Introduction** **26**

**2 Background** **32**

2.1 Simulation of space logistics . . . . . 33

2.2 Space logistics using network flows (GMCNF) . . . . . 36

2.2.1 Static GMCNF . . . . . 37

2.3 Example Problem . . . . . 44

2.3.1 Numerical implementation . . . . . 50

2.4 Dynamic GMCNF . . . . . 55

2.4.1 Shortcomings of the GMCNF applied to Space Mission Design . . . 59

2.5 Summary . . . . . 65

**3 Optimization of executable space missions** **66**

3.1 Development of an executable GMCNF problem . . . . . 68

3.1.1 Semi-continuous GMCNF formulation . . . . . 68

3.1.2 Integer GMCNF formulation: predefined hardware sizes . . . . . 69

3.1.3 Applying the integer GMCNF to the Reference problem . . . . . 72

3.1.4 Complexity and solver requirements for MILP problems . . . . . 78

3.2 Tracking elements through a GMCNF network . . . . . 81

**4 Case study I: design of a manned mission to Mars** **86**

4.1 Setup of the case-study . . . . . 86

4.2 Results of the case-study . . . . . 94

4.3 Discussion . . . . . 96

4.3.1 IMLEO . . . . . 100

4.3.2 Number of launches . . . 101
4.3.3 Number of dockings . . . 102
4.4 Conclusion . . . 104

**5 Case study II: Impact of technological choices on a Mars mission architecture 106**
5.1 Aerocapture . . . 108
5.2 NTR . . . 109
5.3 ISRU . . . 110
5.4 Conclusion . . . 113

**6 Conclusion 114**
6.1 Summary of the thesis: . . . 114
6.2 Futur research directions and potential applications . . . 115

**A Intermediate Problem 118**
A.1 Problem setup: . . . 118

**B Interface between SpaceNet and GMCNF optimization 121**
B.1 Requirements for a SpaceNet-INFINITeX interface . . . 121
B.2 Software architecture . . . 122
B.2.1 GMCNF post-processing (**GMCNFpP**) . . . 124
B.2.2 Input file (in.xls) . . . 124
B.2.3 Post-processed GMCNF file (res.xls) . . . 124
B.2.4 Java interface . . . 124
B.3 General remarks . . . 124

# List of Figures


1-1 Image of the spacecraft *New Shepard* taking off for an unmanned test-flight. Multiple companies are developing space-based tourism. . . . . 27

1-2 In space manufacturing of goods may soon become widespread: [12] shows that crystal growing in micro-gravity allows for qualities not attainable on earth. Above a VDA syringe with protein droplet extruded on syringe tip: one of the earliest experiments on the crystallization of proteins in micro-gravity (image from [12]) . . . . . 27

1-3 Artist's view of a space-based solar power generator . . . . . 28

1-4 Picture of the HARP cannon firing a projectile into a suborbital trajectory (picture from [5]) . . . . . 28

1-5 Thesis roadmap: the chapters are represented as black boxes, around which are the concepts discussed within the chapter. The reader can focus on these instead of reading the thesis from front to end. The scientific contribution of the thesis is found in chapter 3, and results using this contribution are presented in chapters 4 and 5.) . . . . . 31

2-1 Example of a SpaceNet simulation: in this case a lunar module flies from the earth surface at time $t$ and lands on the lunar surface at $t+\Delta t$ and flies back. In this example the agent is the lunar module and the network is represented by the time-expanded Earth-Moon network. Image from [27] . . . . . 33

2-2 Procedure diagram of SpaceNet 2.5. Note the central GUI. Image from [27]. 34

2-3 Network for a lunar campaign including surface (yellow) and orbital (red) nodes, and surface (green), space (red), and flight (yellow) edges. This is an example of a spacial network (only the "places" as opposed to time-expanded network where the trajectories can be seen in a $x-t$ plot). Note that there exist different types of edges: space flight edges, flight edges (a generalized for of space flight edge to allow lower fidelity modeling) and surface edges (on the moon itself). The edges are different because the information they contain varies from type to type (i.e. a $\Delta V$ is irrelevant for a surface edge). Image from [27]. . . . . 35
2-4 Classification of the different agents or elements evolving within the simulation. Image from [27]. . . . . 36
2-5 Classes of Supply (COS): classification of the different classes of material that are modeled in space logistics. Classes of Supply are found both in SpaceNet as well as GMCNF. . . . . 37
2-6 Process of creating a mission scenario in SpaceNet. Note that the user is expeceted to intervene in order to produce a working scenario. Image from [27]. . . . . 38
2-7 Effect of delivering a water recycling plant with better closure: when the system is delivered (on July 2013), the demands shrink. Image from [27]. . 39
2-9 Example of a interplanetary transport network. Image from [37]. . . . . 40
2-10 Nodes $i$ and $j$, directed edge $(i,j)$, and flow $x_{ij}$. Figure from [37]. . . . . 40
2-11 Nodes $i$ and $j$, directed edge $(i,j)$, and flows $\mathbf{x}_{ij}^{+}$ (outgoing) and $\mathbf{x}_{ij}^{-}$ (incoming). Figure from [37]. . . . . 42

2-12 Graph representation of the basic space logistics problem: The starting point is the Earth (E), which is a source of vast amounts of outgoing hardware, but is a returning harware sink, forcing some hardware to flow to Mars (S) where it can be transformed (with the Mars-Mars loop) to returning hardware that flies back. This is done by transiting Low Earth Orbit (LEO), or even flying by the Moon (M) where resources can be collected by ISRU. The ISRU generation depends linearly on the amount of hardware that can be sent to the Moon, where is transformed via the Moon-Moon loop. Note that this is a "toy problem", it not intended to represent the true Earth-Mars-Moon system, but rather a simple problem on which representative phenomena can be modeled and tested. . . . . 46

2-13 Representation of the possible paths that can satisfy the example problem: either a direct path (from $E$ to $L$ to $MS$) or an indirect path (going via $MN$ in order to collect propellant). This trade off is qualitatively the same as seen in full space mission optimizing when ISRU is considered. . . . . 50

2-14 Flows of outgoing hardware (commodity 1) for the optimal reference problem: edge numbers are the same as in fig.2-12. Notice that when an edge is not used, that flow is zero. The network topology changes at the peaks: as the ISRU efficiency increases, the optimal solution goes over the $M$ node to harvest fuel. Once the switch is made, the amount of hardware decreases because the ISRU demands mass for the same quantity of fuel produced. Notice that the specific impulse alters the point at which the network swtiches topology. The different curves correspond to specific impulses: $250s$ (low efficiency chemical propulsion), $450s$ (current chemical propulsion) and $925s$ (nuclear thermal rocket propulsion). . . . . 52

2-15 Flows of returning hardware (commodity 2) for the optimal reference problem: edges are the same as in fig.2-12. Because the constraint of 3 mass units (in this example $kg$) of returning hardware is not altered by the increase in ISRU capability or specific impulse, the lines remain constant. The different curves correspond to specific impulses: $250s$ (low efficiency chemical propulsion), $450s$ (current chemical propulsion) and $925s$ (nuclear thermal rocket propulsion). . . . . 53

2-16 Flows of propellant (commodity 3) for the optimal reference problem: edges are the same as in fig.2-12. The different curves correspont to specific impulses: $250s$ (low efficiency chemical propulsion), $450s$ (current chemical propulsion) and $925s$ (nuclear thermal rocket propulsion). Notice the severe mass increase as $Isp$ drops to $250s$. . . . . 54
2-17 Example of a time-expanded network: the static equivalent are nodes $i$,$j$,$k$. Notice that the expansion process makes the network much larger, both in nodes (at most $N \times n_t$, where $N$ is the number of nodes and $n_t$ the number of time steps) and edges (at most $N^2 \times n_t$). Figure from [33] . . . . . 56
2-18 The node/edge aggregating process: as the aggregating algorithm [33] goes through the network, nodes and edges are aggregated reducing the size of the network and lowering the computational needs to solve the problem. Image source: [33] . . . . . 56
2-19 Image source: [33] . . . . . 57
2-20 Relative positions of the results of the optimization of time-expanded GMCNF: the restricted network produces an upper bound (feasible, but suboptimal), the aggregated network produces a lower bound (infeasible). The feasible and optimal solution is located between both. The static GMCNF can be seen as a "super aggregated network". Image source: [33] . . . . . 58
2-21 Depiction of a clustering process of a time-expanded transport network. Note the dual time-scale: one long time-scale between clusters, and a short time-scale within the clusters. Image source: [33] . . . . . 58
2-22 Depiction of a partially static time-expanded network. In [33], it is found that this method is particularly effective for the space mission application. This type of network is the one used for all time-expanded networks in this thesis. Image source: [33] . . . . . 59

2-23 The difference between both optimal vectors $x^*_{cplex}$ and $x^*_{gurobi}$ show that although the objective function value is the same, the solution vectors may differ substantially in some parts. Note that most elements are identical between both vectors (most elements between the vectors are identical and thus show a zero difference). Note how this graph is almost symmetrical along the $y = 0$ axis: this is because the values of the different components are identical between the CPLEX result and the Gurobi result, but they are shifted in position by a small amount. This reflects differences in schedule in the mission: if a given flight taking place in both optimization results differs slightly in the date at which it is done, then the differences between the two solutions are non zero and show this behavior seen above. This happens because the sensitivity of the objective function to these slight changes in schedule is close to zero, and thus note captured by the optimizer. . . . . 61

2-24 This chart depicts the values of the different heat-shields used in the Mars only scenario described in [33]. The "size" indicates the number of different heatshield mass values throughout the mission (or the number of data points in this case). . . . . 61

2-25 This chart shows the same as the previous one, except that it does this for the size of LH2/LOX rocket engine size along an edge. This figure means that 176 different sized rocket engines must be built to fly this mission. One could argue that a cut-off size be chosen, in order to keep the large elements driving the network and neglecting the small flows. Unfortunately this cannot work as the rocket engine sizes are distributed across the mass spectrum. . . . . 62

2-26 This chart shows the same as the previous one, except that it does this for the amount of hydrogen carried along an edge. Note that the GMCNF will size the spacecraft and hardware around these values, thus their number is an indicator of the number of times a vehicle assembly is reconfigured. Notice the continuous variation between 0 kg and $1 \times 10^4$ . . . . . 62

2-27 This shows how NTR engines move along the network. The different colours indicate different edges, and the numbers next to these indicate the size of NTR in kg. Note that NTR are left to "dissociate" and "recombine". This is not a surprise given the amount of different flow sizes transting the network. Note that this is only for the inert mass of the NTR. Similar plots can be drawn for any other commodity (there are 23 in total). . . . . 63

2-28 This shows the same as the previous page, but for the network flow of the heatshield, as to show that the complexity of the network of transiting hardware does not only affect the NTR, but all commodities. . . . . 64

3-1 Abstract link between event-based simulation (SpaceNet) and flow-based optimization (GMCNF). The event-based simulation is constituted of elements in a logisitics network, possing a time dependant vector of states $s(t)$. The events driving the simulation a comprised by a series of canonical events, such as *move*, *create*,*consume*, etc. For example, a burn is a combination of moving and consuming. The GMCNF optimization is flow-centric, where the flow $x_{ijkt}$ carries a commodity $k$ between nodes $i$ and $j$ at time $t$. Translating the output of the GMCNF into an input for SpaceNet requires that every flow $x_{ijkt}$ be matched to an element in the SpaceNet network: this enforces that the flows in the GMCNF be traceable. Events and states can be obtained by looking at the evolution of the flow or element: a flight from node $i$ to $j$ may for example show a burn. Determining what events happen when is a post-processing task once the flows have been allocated to elements. . 67

3-2 Flows of outgoing hardware (commodity 1) for the mixed integer reference problem: edges are the same as in fig.2-12. Notice that the network topology switch apparently does not depend on $m_{unit}$. Notice how for large $m_{unit}$ the curve becomes less smooth: this is due to the optimizer having to switch between integers (or discrete numbers of rocket elements) that each are not optimal (because of the inequality constraint the stages are not optimally loaded). . . . . 74

3-3 Flows of returning hardware (commodity 2) for the mixed integer reference problem: the results do not change (all curves are superposed) no matter $m_{unit}$ or the IRSU rate $\chi$. This happens because this particular commodity is not affected by the others (e.g. it does not limited in any way by the quantity of others) but it drives the problem. Another example of this is the "astronaut" commodity in the previous GMCNF studies [33],[37]. Note that in these works, this commodity flows through the optimal network with no separation or recombination, like other commodities such as rocket stages do. 75

3-4 Flows of outgoing hardware (commodity 3) for the mixed integer reference problem: The comments made for figure 3-2 can also be made here. However, this commodity is not constrained to be a multiple of $m_{unit}$. The peaks observed for $m_{unit} = 10kg$ follow those of the outgoing hardware, simply because any excess outgoing hardware requires excess propellant to fly it. This illustrates the real-life problem of having to size a piece of hardware for an entire mission and not simply around a specific operating point. . . . . 76

3-5 This plot illustrates the relative difference between the objective function value at optimum for the MILP problem and the LP problem: notice the log-plot: the IMLEO increases drastically when $m_{unit}$ becomes large, illustrating the sharp mass increase due to the rocket equation applied to excess mass due to a rocket stage that cannot resize itself for a specific mission leg, like it does in the continuous case. Obviously, when $m_{unit}$ becomes small, the problem goes towards becoming a continuous one, hence the small differences for small $m_{unit}$ are expected. . . . . 77

3-6 The amount of additional time required to solve a MILP instead of an LP in the GMCNF formulation is critical, as the resolution of larger networks with more fidelity might become prohibitive. In this case, a third of the variables have been constrained to take integer values, and the mean time to solve the new problem is 30 times larger. If this applies to the networks used in [33], the resolution time becomes close to 2 hours and 30 minutes, as opposed to 5 minutes. Note that these computation times remain practicle. . . . . 77

3-7 Comparison of the CPU time between CPLEX and Gurobi to solve identical optimizations problem: on average, Gurobi is twice as fast as CPLEX, making it the optimizer used for the rest of this work. . . . . 78
3-8 Convergence of the objective function value with CPU time: the problem is the full problem discussed in chapter 4. The computer is a high performance desktop computer with 16 intel Zeon cores. The nature of the variables (integer/non integer) can be seen in table 4.1 . . . . . 81
3-9 Reference problem for the tracking algorithm: the locations and time are generic ones. The goal of this reference problem is to test the tracking algorithm and illustrate its functioning. Here, a commodity of 3 leaves location 1 to split into different parts, then come together again etc. This illustrate typically the phenomena occurring in an integer GMCNF result: elements split, recombine or are abandoned. . . . . 82
3-10 Mass flow of the NTRs throughout the network for an executable GMCNF after 5000 seconds of optimisation. . . . . 85
3-11 Mass flow of the NTRs throughout the network for an executable GMCNF after having been processed by the tracking algorithm: the different colors show how different elements fly through the hardware. . . . . 85

4-1 Illustration of the space-network used for the case-study. . . . . 87
4-2 Vehicle flow of the Mars mission case study of [33] . . . . . 88
4-3 Payload flow of the Mars mission case study of [33] . . . . . 88
4-4 Example of a launch concept (OTRAG launcher, [4]) with multiple unit rockets strapped together to form a large rocket. . . . . 90
4-5 Mass flow of the aeroshell throughout the network. Although the general scheme is the same as the continuous problem shown in fig.2-28, there are slight differences in the topology locally, but mostly it becomes clear on how to track elements through the network. . . . . 94
4-6 Mass flow of the vehicle throughout the network. . . . . 94
4-7 Mass flow of the NTRs throughout the network. . . . . 95

4-8 Number of different masses throughout the network for all commodities: because of the integer constraints, the number of different instances becomes manageable for all commodities that are discretized. Commodities like hydrogen and oxygen may vary continuously throughout the mission, so their high number of different masses is not an issue as hydrogen and oxygen does not have to be tracked through the network. . . . . 95

4-9 Mission concept of DRA5.0. Note the two mission segments: the first installs the space-based infrastructure, and the second sends the humans there and back. . . . . 97

4-10 Mass flow diagram of the dynamic GMCNF mission architecture: the commodities are grouped into different categories for better visualization. All propulsion elements (LOX rockets, NTRs, LCH4) are grouped, as well as all tank types, vehicles, and astronauts. Note the two segment architecture, where the first segment lasts over a long time in order to accumulate the different propellants, and then the astronauts are sent. . . . . 98

4-11 Executable mission architecture overview. Note that it has less flows than the dynamic GMCNG architecture. . . . . 99

4-12 ITS mission architecture: note the relative simplicity of the mission. Instead of deploying a large in space infrastructure like the GMCNF cases or a complexe mission such as in DRA5.0, the mission architecture pays less attention to IMLEO and allows for larger spacecraft that do not need intricate in-space infrastructure to function. Image from [11] . . . . . 99

4-13 IMLEO for the four missions considered in this study: because the ITS mission is has a different purpose (colonize Mars instead of exploring it), it has a much larger launch mass, and should not be compared directly to the other 3 missions. These have different IMLEOs: assuming a launch cost of $10'000$ USD per kilogram, the dynamic GMCNF represents a cost reduction of $4,173$ billion USD (49% reduction in IMLEO), but unfortunately this architecture is not executable. Even though the executable GMCNF mission only has a 37% reduction is IMLEO, it still saves 3.165 billion USD with respect to the reference architecture. . . . . 100

4-14 Number of launches for a single mission to Mars for all 4 missions: DRA5.0 requires 9 cargo launches, the dynamic GMCNF require 7 cargo launches, the executable one requires 6, and the ITS requires 5 cargo launches. All missions send astronauts once the cargo and/or infrastructure is in place, except for the ITS mission. . . . . 101
4-15 Mass flow diagram of the continuous GMCNF. The manned segment is contained in the red rectangle. 7 docking/assembly operations are counted around the manned vehicle. . . . . 102
4-16 Mass flow diagram of the continuous GMCNF. The manned segment is contained in the red rectangle. 5 docking/assembly operations are counted around the manned vehicle. Note that this is 2 less (or a 28% relative reduction) than for the dynamic GMCNF. . . . . 103
4-17 Number of dockings for the manned segments of each mission. Note that all are the same 5 dockings except for the dynamic GMCNF that needs a higher number because the mission is more complex . . . . . 104
4-18 . . . . . 105

5-1 Artist rendering of an inflatable heat-shield executing aerocapture in the Martian atmosphere . . . . . 107
5-2 Schematic of an "Expander cycle" NTR Engine with dual $LH_2$ turbopumps [10] . . . . . 107
5-3 Artist rendering of ISRU water extraction unit . . . . . 108
5-4 Mission architecture used when aerobracking is not used. Although the general architecture does not change, IMLEO increases by 33.0 % because propellant used to perform Martian capture must be carried along. . . . . 109
5-5 Mission architecture used when NTR is not used. Again, the general mission architecture does not change with respect to the base case, but IMLEO increases by 27.5 % with respect to the reference executable GMCNF scenario.110
5-6 Mission architecture to Mars using no ISRU. Note that the architecture is the same as DRA5.0, which contributes to validating the executable GMCNF methodology. . . . . 111

5-7 IMLEO versus Lunar ISRU rate: note that although lunar ISRU is overall beneficial to IMLEO, it only represents a 5% IMLEO gain at 5 kg/kg/year. An aspect to investigate is the discrepancy between the IMLEO of the dynamic GMCNF at no Lunar ISRU and the DRA5.0 IMLEO: these values should be the same, validating the GMCNF model. [33] uses the same model and falls within 10% of the IMLEO of DRA5.0. The discrepancy presented here is likely due to the Martian ISRU extraction rates that are too optimistic in GMCNF with respect to DRA5.0. . . . . 112

5-8 IMLEO vs lunar ISRU rate. Note the sharper decrease in IMLEO due to ISRU using static GMCNF. Image from [33]. . . . . 113

6-1 Artist rendering of the Zee Aero personal air vehicle. A potential application for GMCNF related methods is the design of urban air mobility networks and vehicles. . . . . 115

A-1 Graph representation of a smaller test problem. Here we only have 2 commodities, namely a commodity $C_1$ (such as hardware) and a commodity $C_2$ (such as fuel). Here, node $A$ can provide up to 5 counts of $C_1$, but consumes 2 counts of $C_2$. This experiment aims at observing the trade-off between obtaining $C_2$ by transforming it from A (arc 1), or sending it through the network to B, where it can be multiplied at a rate $\chi$. There is a second variable, which is the cost of transportation from node A to B and back, which is $\phi$. This problem represents a reduced version of the reference problem, and can be solved by hand. The transportation cost of all commodities is 1 on arcs 1, $\phi$ on arc 2 and 3, and 0 on arc 4. . . . . 119

A-2 The result of the study of the intermediate problem: these contour maps represent the total (sum of all commodities) optimized flows through the indicated edge (1,2,3 and 4) for a network problem with the parameters $\phi$ and $\chi$. Note the sharp contrast on edges 1 and 4: this comes from the optimal network topology switch occurring along the analytically calculated line $\chi = \frac{\phi}{1-\phi}$. For the other two edges this is less obvious, as the $\chi$ parameter varies the amount of flow that is needed to be sent from node A, e.g. a higher conversion rate lowers the amount of flow needed to be transported out to B for a given amount of flow being sent back to A. . . . . 120

B-1 At first there is the matlab module that produces the optimized transportation network, **INFINITeX**, which corresponds to Ho's work [33]. This feeds **GMCNFpP.m**, which post-processes the restults into an excel file that will then be read by the **javaInterface**, which makes use of SpaceNet's classes to write an .xml file which is then read by **SpaceNet**. . . . . 123

# List of Tables

2.1 Isp: [b]: shuttle crew operations manual, [d]: Apollo $\Delta V$ budget [c]: Isp of the Apollo service module, [a]: Isp of the Saturn V first stage [e] . . . . . 48

2.2 Optimization results for a NEO only scenario. *: the constraints are of the form $\mathbf{A}x \leq b$, thus is the constraints are met then $max(\mathbf{A}xb) \leq 0$. Note that the very low value show that all constraints are met, as the values shown are little over zero. . . . . 60

3.1 Example of a integer GMCNF output. Once this is processed using the tracking algorithm, the results become executable. . . . . 83

4.1 Summary of the parameters in the executable problem: the table shows the discretization of the different commodities. If the variable is chosen to be continuous, the upper bound is the maximum mass in kg, and if it is chosen to be discrete, the upper bound is the maximum number of intances along one edge. . . . . 93

5.1 Summary of results concerning technology switches: . . . . . 113

# Nomenclature

## Abbreviations

| | |
|---|---|
| COS | class of supply |
| DEIM | Deimos |
| DRA | design reference architecture |
| DSS | decision support system |
| DTO | Deimos transfer orbit |
| ECLSS | environmental control and life support system |
| EML | Earth-Moon Lagrange point |
| EVA | extra-vehicular activity |
| GC | Gale Crater |
| GEO | geostationary Earth orbit |
| GMCNF | generalized multi-commodity network flow |
| GPS | Global Positioning System |
| GTO | geostationary transfer orbit |
| GUI | graphical user interface |
| ILP | integer linear programming |
| IMLEO | initial mass in LEO |
| IP | integer programming |
| ISRU | in-situ resource utilization |
| ISS | International Space Station |
| ITS | Interplanetary Transport System |
| JPL | Jet Propulsion Laboratory |
| JSC | Johnson Space Center |

| | |
|---|---|
| KSC | Kennedy Space Center |
| LEO | low-Earth orbit |
| $LH_2$ | liquid hydrogen |
| LLO | low lunar orbit |
| LMO | low Mars orbit |
| LOX | liquid oxygen |
| LP | linear programming |
| LSP | lunar south pole |
| MCNF | multi-commodity network flow |
| MILP | mixed integer linear programming |
| MINLP | mixed integer nonlinear programming |
| MIT | Massachusetts Institute of Technology |
| MOE | measure of effectiveness |
| NASA | National Aeronautics and Space Administration |
| NEO | near-Earth object |
| NLP | nonlinear programming |
| NTR | nuclear thermal rocket |
| OPEX | operational expenditure |
| OTV | orbital transfer vehicle |
| PAC | Pacific Ocean splashdown |
| PHOB | Phobos |
| PTO | Phobos transfer orbit |
| TDN | time-expanded decision network |
| USD | United States dollar |
| USD | United States dollar |

## Symbols

| | |
|---|---|
| $\mathbf{A}$ | requirement matrix |
| $\tilde{\mathbf{A}}_{ij}$ | mass normalized equality constraint matrix (integer problem) |
| $\tilde{\mathbf{A}}_{ij}^{ineq}$ | mass normalized inequality constraint matrix (integer problem) |

| | |
|---|---|
| $\mathcal{A}$ | set of directed edges |
| $\mathbf{B}$ | transformation matrix |
| $\tilde{\mathbf{b}}_j^{eq}$ | RHS of equality constraints for integer problem |
| $\tilde{\mathbf{b}}_j^{ineq}$ | RHS of inequality constraints for integer problem |
| $\mathbf{C}$ | concurrency matrix |
| $\tilde{\mathbf{c}}_i$ | mass normalized cost vector (integer problem) |
| $E$ | Set of non-zero edges leaving a node in the time-expanded logistics network |
| $\mathcal{G}$ | directed network graph |
| $I_{\text{sp}}$ | specific impulse |
| $\mathcal{J}$ | objective function |
| $\mathcal{K}$ | set of commodities |
| $N$ | dimension of LP problem |
| $\mathcal{N}$ | set of nodes |
| $b$ | net supply/demand |
| $\mathbf{b}$ | net supply/demand vector |
| $c$ | cost per unit flow |
| $\mathbf{c}$ | unit cost vector |
| $f$ | Darcy-Weisbach friction coefficient |
| $f_{\text{inert}}$ | inert mass fraction |
| $g_0$ | standard gravity ($= 9.80665$ m/s$^2$) |
| $i$ | node index ($\in \mathcal{N}$) |
| $i$ | variable index |
| $j$ | node index ($\in \mathcal{N}$) |
| $(i, j)$ | edge from node $i$ to node $j$ ($\in \mathcal{A}$) |
| $k$ | commodity index ($\in \mathcal{K}$) |
| $k$ | number of types of commodities |
| $l$ | edge lower bound |
| $l$ | number of concurrency constraints on each edge |
| $m$ | number of directed edges |
| $m$ | mass |
| $\tilde{m}$ | element or unit mass |
| $\tilde{\mathbf{M}}$ | Mass normalizing matrix |

| | |
|---|---|
| $m_{\text{as}}$ | aeroshell mass |
| $m_{\text{crew}}$ | crew mass |
| $m_{\text{dr}}$ | dry mass |
| $m_{\text{p}}$ | mass of water pumped |
| $m_{\text{pr}}$ | propellant mass |
| $m_{\text{sc}}$ | spacecraft mass |
| $m_{\text{st}}$ | structure mass |
| $m_{\text{vehicle}}$ | vehicle mass |
| $n$ | number of nodes |
| $n_x$ | number of variables |
| $n_{\text{eq}}$ | number of equality constraints |
| $n_{\text{ineq}}$ | number of inequality constraints |
| $u$ | edge capacity |
| $\tilde{u}_i^{lower}$ | mass normalized minimum edge capacity |
| $\tilde{u}_i^{upper}$ | mass normalized maximum edge capacity |
| $\mathbf{u}$ | edge capacity vector |
| $x$ | flow variable |
| $\mathbf{x}$ | flow variable vector |
| $\Delta m$ | change in mass |
| $\Delta t$ | duration of edge |
| $\Delta V$ | change in velocity |
| $\alpha$ | proportional constant for ISRU maintenance mass |
| $\beta$ | proportional constant for ISRU resource production |
| $\gamma$ | minimum vehicle mass per crew member |
| $\eta$ | ratio of structure mass to propellant mass |
| $\theta$ | aeroshell mass fraction |
| $\mu$ | multiplier for generalized network flow |
| $\mu$ | mixture ratio of oxidizer to fuel |
| $\xi$ | mass fraction of return samples (only applies to example problem) |
| $\phi$ | propellant mass fraction |

## Superscripts and Subscripts

| | |
|---|---|
| $(\cdot)^{+}$ | outflow from tail node |
| $(\cdot)^{-}$ | inflow into head node |
| $(\cdot)^{\pm}$ | both outflow and inflow |
| $(\cdot)^{\mathrm{T}}$ | transpose |
| $(\cdot)^{k}$ | commodity $k$ |
| $(\cdot)_{i}$ | node $i$ |
| $(\cdot)_{j}$ | node $j$ |
| $(\cdot)_{ii}$ | graph-loop associated with node $i$ |
| $(\cdot)_{ij}$ | edge $(i, j)$ |
| $(\cdot)_{pq}$ | $(p, q)$ entry of a matrix |
| $(\cdot)_{\mathrm{eq}}$ | equality constraint |
| $(\cdot)_{\mathrm{ineq}}$ | inequality constraint |

## Definitions

**Executable space mission** An executable mission is a mission architecture in which all events and elements can be traced and scheduled and flown. A counter example of this is a mission that contains time-paradoxes or in which elements such as vehicles cannot be traced.

**INFINIT** **INFINIT** is the GMCNF software developed by [37] which is applied to a space logistice problem (flying humans to Mars).

**Mass flow diagram** A mass flow diagram is a diagram showing how a commodity or multiple commodities flow through a time-expanded network. The $x$ axis is time, the $y$ axis represents the locations, and the legs in the digram represent the flow. Different colors, unless mentioned otherwise, represent a switch from one edge of the network to the next.

**Mass spectrum** In this work, a mass spectrum is a diagram which shows the number of instances, or number of edges, along which a given mass value flows. The $x$ axis

represents the mass values, and the $y$ axis represents the number of times this values appears in the network.

**SpaceNet** **SpaceNet** is the software developed by MIT and JPL for studying detailed space logistics.

**Time-Paradoxe** A time-paradoxe is a phenomena occuring when a dynamic problem is solved using a static method. In the logistics contexte, this means that resources at consumed at a given point bevore the are supplied.

# Chapter 1

# Introduction

A key factor in space exploration or any other major scientific endeavour is bringing its cost down to an affordable level. This is especially true for space exploration, where it has become difficult to find financial means since the underlying military and political interest dating from the cold war has largely died out.
However, recent developments in space technology have bolstered a wide number of applications that are rapidly developing: these include space tourism (fig.1-1) or in orbit manufacturing of goods (fig.1-2). In a more distant future, large scale industrial applications such as energy harvesting (fig.1-3) might add themselves to a broad spectrum of activities creating a space based economy. Expanding the human presence and allowing humans to prosper in beyond the atmosphere requires solving a wide variety of problems, among which are the logistics problems of deploying and maintaining a space based infrastructure. These problems are at the heart of this thesis. On of the problems opposing advances in space exploration and exploitation in the 21st century is finding novel approaches to making space operations affordable. A major bottleneck in space mission costs is mass sent to low earth orbit (LEO): SpacEx's falcon heavy allows costs 2200$/kg, and more classical rockets such as ESA's Ariane 5 or the US Airforce's Delta IV cost 10'476 and 13'182 $/kg respectively [1].

Two classes of approaches may be taken to address this cost issue:

- Either ways are found to drop the launch cost by developing better launch vehicles or different launch capabilities all together,
- Or the amount of mass needed to be sent into space is reduced


Figure 1-1: Image of the spacecraft *New Shepard* taking off for an unmanned test-flight. Multiple companies are developing space-based tourism.

Figure 1-2: In space manufacturing of goods may soon become widespread: [12] shows that crystal growing in micro-gravity allows for qualities not attainable on earth. Above a VDA syringe with protein droplet extruded on syringe tip: one of the earliest experiments on the crystallization of proteins in micro-gravity (image from [12])

Note that these solutions are not mutually exclusive, both directions must be pursued. The first approach has been worked at since the beginning of space flight, however, given the extreme variations in velocity and altitude, only chemically-rocket propelled vehicles have ever flown to space (with few remarkable exceptions, one of which was project HARP , see figure 1-4). Some alternative approaches have been studied and tested to some extent, such as space elevators, rail guns, launch towers, laser pulsed rockets and even nuclear pulsed rockets [43], [7].

It can be shown that the propulsive efficiency of chemical rockets, or specific impulse (Isp) is limited by the nature of the chemical elements available: the highest efficiency non-lethal propellant combination (hydrogen and oxygen) has a theoretical Isp of 459 seconds

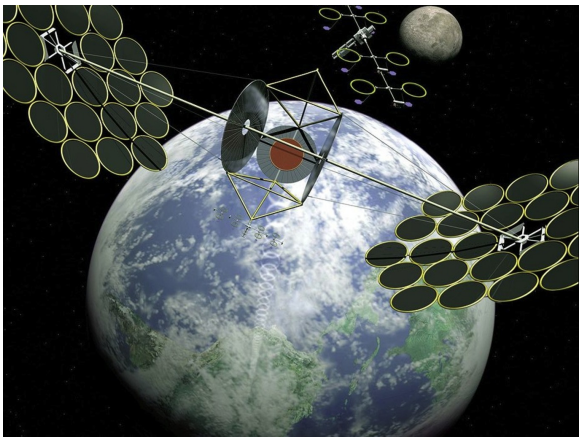


Figure 1-3: Artist's view of a space-based solar power generator

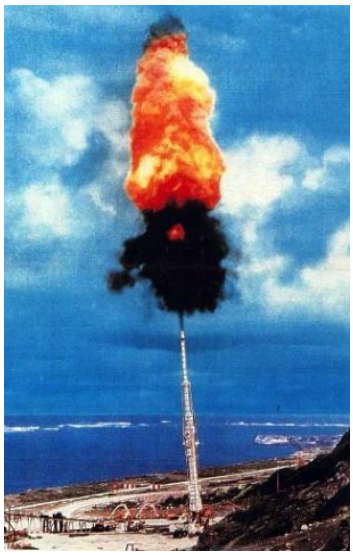


Figure 1-4: Picture of the HARP cannon firing a projectile into a suborbital trajectory (picture from [5])

(in vacuum), while the space shuttle main engines (SSMEs) have an Isp of 452.3 seconds [6], showing that little efficiency can be gained from chemical rocket propulsion. The latest trend in chemical-rocket development has been making the launch vehicle reusable. To date however, reusable vehicles have not demonstrated sustainable low launch costs, despite many efforts on the subject (example include [17], [32], [44]).

The second approach, e.g. reducing the launched mass can be done in multiple ways:

- Reduction of mass of the physical systems themselves, which always has been a concern in space systems design. One of the more radical manifestations of this effort has been the recent and intense efforts is the design of micro- and nano-satellites, and the

technologies associated with them.

- Reduction of mass over the entire life-cycle of a space campaign. This is more subtle, because the optimization and engineering happen at the mission level planning and writing of design specifications for the spacecraft. This is called space-logistics, and although its importance has been pointed out by Wernher Von Braun as early as 1960, serious efforts have only started in the early 2000's [27].

Space logistics has great potential: unlike developing exotic launch capabilities that are either too large in scale to be initiated or pursuing incremental development such as focusing on spacecraft design alone, the cost reduction due to proper mission planning and subsequent design of the hardware has shown promising results: in 2013, using a new optimization technique in space logistics, Dr. T. Ishimatsu by was able to create a mission to Mars that would drop the mass in LEO by 67.5% with respect to NASA's reference mission (DRA5.0) [37] Unfortunately, this mission was not feasible due to the fact that the method used to devise is allowed for time-paradoxes, e.i. cases where resources would be used at a given point before they were sent there.
In 2015, Dr. K. Ho expanded Ishimatsu's work by adding the time-dependency, which removed the time-paradoxes. The recurring problem behind these approaches is that their level of fidelity does not allow to determine whether a scenario is feasible or not.
Although only the cost of reaching orbit is paramount in the work mentioned above, other aspects of space missions have to be optimized. This is within the scope of the methods developed in [37] and [**?**], as long as these quantities stay global. For detailed mission feasibility analysis another approach is used: mission simulation. Mission simulation and the work around it at MIT is embodied by the Space Net software, allowing the detailed simulation of a space exploration campaign.

This work seeks to bridge the gap between the high-level, low-fidelity optimization and the high-fidelity, time dependent simulation allowing to verify the feasibility of the result issued from the optimization.
The road-map for this work is depicted in fig.1-5. The outline of this thesis is:

1. **Introduction**: this chapter introduces the subject of space logistics and the problematic of the thesis.

2. **Background**: the background explores the space logistics from the aspects of simulation of space missions and generalized multi-commoditiy network flows (GMCNF). It exposes both the simulation side (SpaceNet) and then goes on to introducing the GMCNF methods. The GMCNF methods are illustrated with an example problem. It is explained why current GMCNF methods cannot be interfaced with software such as SpaceNet because the outputs of GMCNF are not executable.

3. **Executable GMCNF**: This chapter presents the scientific contribution of this thesis. Because of shortcomings of the GMCNF formulations discussed in the previous chapter, a modified version of GMCNF formulation is proposed to produce executable outputs.

4. **Case-study I: mission to Mars**: the integer GMCNF is used to produce a mission architecture for a manned mission to Mars. The parameters and inputs of the problem are presented and the results are compared to DRA 5.0, the results in [**?**] and SpaceX's Interplanetary transport system (ITS).

5. **Case study II: technology evaluation**: in this case-study, GMCNF is used to determine the impact of unconventional technologies such as NTR, ISRU and aerobracking on IMLEO for a manned Mars exploration mission.

6. **Conclusion**: the thesis is closed by a concluding chapter summarizing the work and results and suggesting new research directions

7. **Appendix A**: Appendix A contains a simple 2 node GMCNF problem which is solved analytically to show the non-linear nature between parameters of GMCNF problems.

8. **Appendix B**: Appendix B suggests a software architecture to link GMCNF software to SpaceNet.

9. **Appendix C**: Appendix C presents the full results of the executable GMCNF for a manned Mars mission optimization with NTR, Aerobracking and ISRU rates of 5 kg/kg/year of oxygen on the Moon, 5 kg/kg/year of oxygen on the Mars and 10 kg/kg/year of hydrogen on Mars.

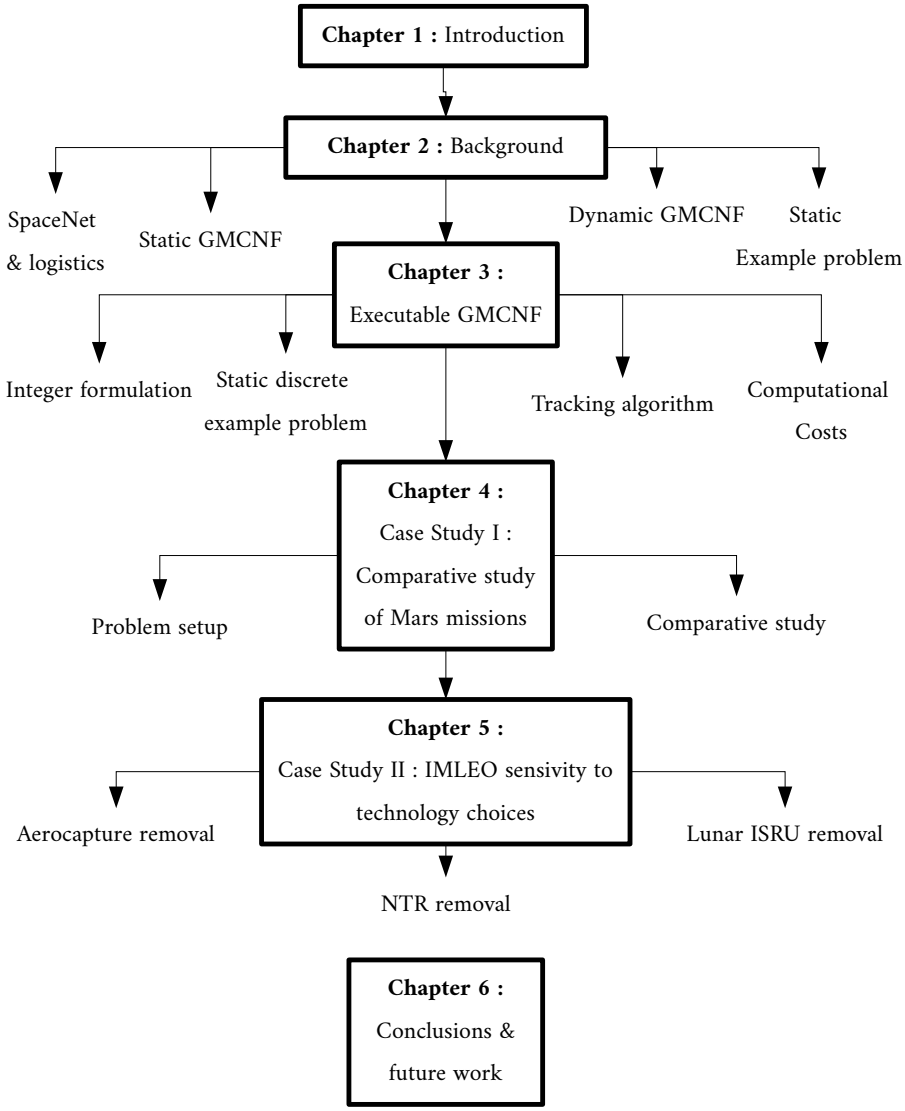



Figure 1-5: Thesis roadmap: the chapters are represented as black boxes, around which are the concepts discussed within the chapter. The reader can focus on these instead of reading the thesis from front to end. The scientific contribution of the thesis is found in chapter 3, and results using this contribution are presented in chapters 4 and 5.)

# Chapter 2

# Background

This chapter presents the state of the art by discussing two distinct approaches to the study of space logistics:

- The first is the small scale, high fidelity approach that is represented by the SpaceNet simulation software. This section describes the motivation for such an approach and some fundamental concepts on the subject before giving a short introduction to a software tool developed jointly by the Jet Propulsion Laboratory (JPL) and the space logistice group at MIT.
- The second is at large scale and low fidelity. This is the Generalized Multi-Commodity Network Flow (GMCNF) approach, that allows for a simultaneous and extensive optimization of the logistics network and the agents evolving within it. The section is introduced by the static GMCNF formulation, and followed by the dynamic GMCNF formulation. Both are presented and discussed in terms of their advantages and shortcomings in the space mission design context. The static GMCNF formulation is illustrated by an example problem, referred to as the Example Problem. This is followed by a presentation of the dynamic GMCNF and its inherent limitations.

The chapter is concluded by illustrating that the two approaches can not be coupled in their current form, and expresses the requirements on a new GMCNF to be compatible with an event based simulation.

## 2.1 Simulation of space logistics

Although Wernher Von Braun underlined the importance of the subject as early as 1960, space logistics has only recently gained interest by national space agencies and companies. This interest has been fueled by the desire of returning manned exploration missions to the Moon and Mars. The long term ISS missions have brought logistics challenges forward [24],[25] such as planning long term resuppy given all the different needs on the ISS.

A way to gain understanding in the problem and study different mission scenarios is to create a computer simulation tool. The idea behind this simulation is to use an event-based simulation of agents within a logistics network [46]:

The network would be populated with elements and these elements would have states and would be driven by events. A representation of this framework is summarized in fig.3-1. An example of an element in a network is presented below in fig.2-1 where a lunar module flies to the Moon. This is case, the logistics network is the time-expanded Earth-Moon network (i.e. any possible segments linking the Moon and Earth in the $x - t$ plot shown) and the agent is the lunar module. In this example, an event would be the flight of the lunar module to the Moon, and another one would be the return of the module from the Moon. However, if for some reason the lunar module cannot fly to the moon, then the second event cannot take place and the simulation stops: the mission is not executable.

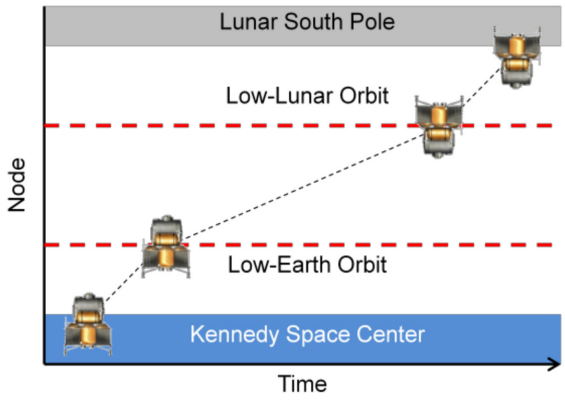



Figure 2-1: Example of a SpaceNet simulation: in this case a lunar module flies from the earth surface at time $t$ and lands on the lunar surface at $t + \Delta t$ and flies back. In this example the agent is the lunar module and the network is represented by the time-expanded Earth-Moon network. Image from [27]

The effort to produce SpaceNet started in 2007 and produced a Matlab® code [46]

that had the simulation capability, and optimization and trade study capability, but lacked structure and the software was difficult to improve. This issue was addressed in 2010 by Dr. P. Grogan [27], who wrote a Java based software. The object oriented capability of the programming language allowed for this "evolvability" of the code, and was better suited to the event-based simulation. The general structure of the code revolves around a graphical user interface (GUI) that coordinates the event-based simulation, as shown in fig.2-2

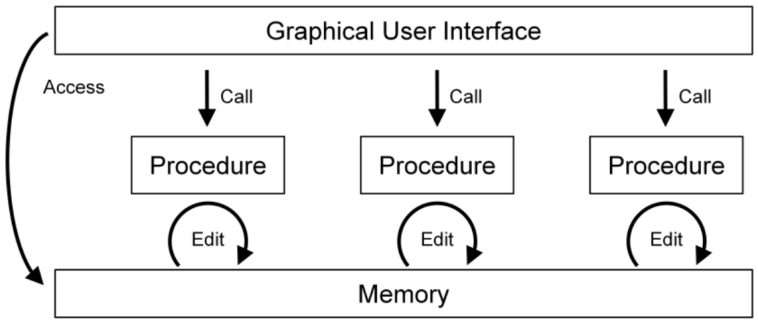



Figure 2-2: Procedure diagram of SpaceNet 2.5. Note the central GUI. Image from [27].

Two main categories of objects populate a SpaceNet mission scenario:

**Network objects:** these are the nodes and edges of a transportation network. An example of this is shown in fig.2-3: the nodes are locations at which elements can wait or do any other form of activity. Examples are the Kennedy Space Center (KSC) for launch, or low Earth Orbit (LEO), or Gale Crater (GC) on Mars. These nodes can be orbital nodes, or surface nodes, or space nodes (such as lagrange points for example).
The nodes are linked by edges along which elements can move. Examples of these include transfer orbits, a flight of some sort or a drive from one surface point to another.

**Agents or Elements** the network is populated by elements that accomplish actions such as moving in the network but also transfer loads between each other, or explore, or burn propellant, land etc. The organization of the classes with SpaceNet 2.5 is shown in fig.2-4. Another important aspect is the classification of the different supplies in space logisitics and within the SpaceNet software, which is depicted in fig.2-5

Building a mission scenario in SpaceNet follows the process depicted in fig.2-6. It is important to notice that the user is expected to intervene in order to produce a working

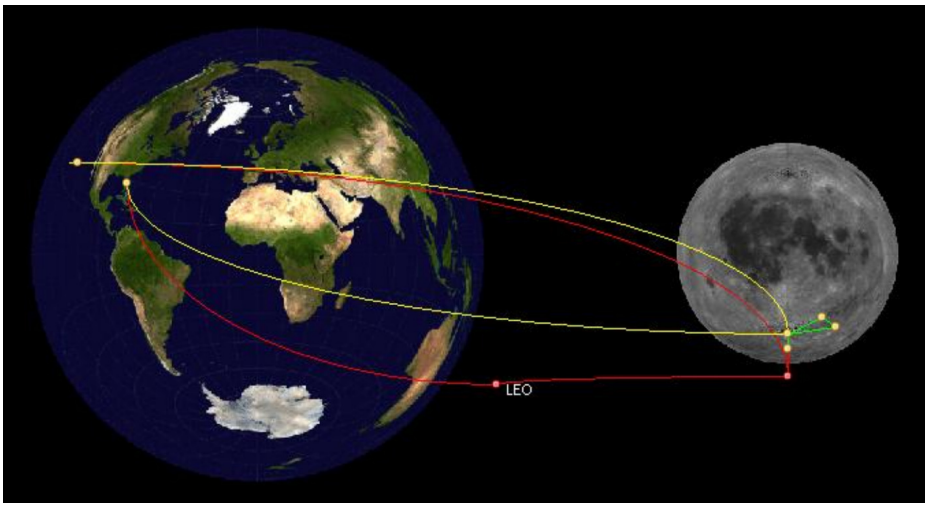



Figure 2-3: Network for a lunar campaign including surface (yellow) and orbital (red) nodes, and surface (green), space (red), and flight (yellow) edges. This is an example of a spacial network (only the "places" as opposed to time-expanded network where the trajectories can be seen in a $x-t$ plot). Note that there exist different types of edges: space flight edges, flight edges (a generalized for of space flight edge to allow lower fidelity modeling) and surface edges (on the moon itself). The edges are different because the information they contain varies from type to type (i.e. a $\Delta V$ is irrelevant for a surface edge). Image from [27].

mission scenario. Experience shows that user input and correction happens often before an working scenario can be produced [29],[47],[27],[48],[49]. This is due to the fact that because of the event-based nature of the simulation, certain events cannot take place if the previous events do not take place. An example of this is a miscalculation in propellant mass: if a carrier (such as a lunar module) is sent on edge of the transportation network that requires a $\Delta V$ that is larger than that element can afford, the element ends up "stuck" on that edge, and cannot continue its mission. If that module is supposed to pick up astronauts for example, than the simulation stops (note that it can be allowed for be forced to continue with logical contradictions, such as an element left mid-way). Because many different events and scenario parameters can be responsible for this type of occurrence, it is up to the user to trace back different events until the problem is found [29],[28].

This feature due to the event-based simulation makes this tool unpractical for studying large trade spaces (although it is capable of local optimization, such as cargo manifesting [30]) and generating different mission architectures.

But this is not the goal of SpaceNet: its primary use is the simulation of space mission logistics. An example of this is to study the effect of delivering a better ECLSS system to

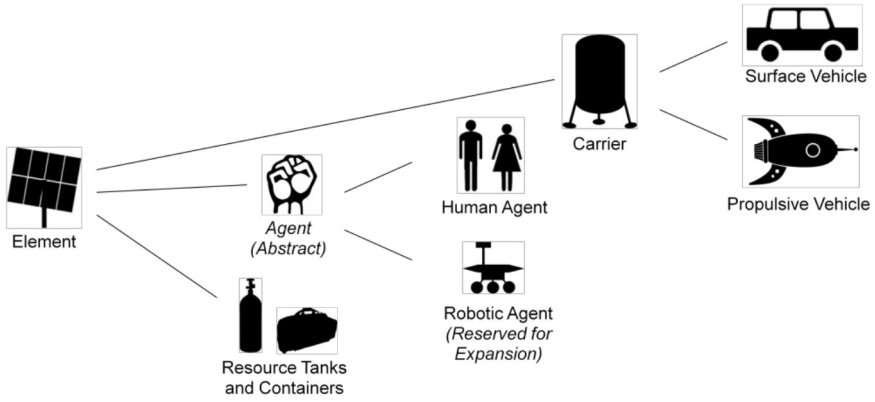



Figure 2-4: Classification of the different agents or elements evolving within the simulation. Image from [27].

the ISS, as shown in fig.2-7 In this study, a water recovery system is sent to the ISS to see how this affects overall demands. As can be seen from this simulation, a water recovery systems with higher recovery rate drops the demands for the ISS, which in turn means a lower need for resupply. SpaceNet helps answer questions such as "is it worth sending up a better ECLSS, and after how much time is the investment amortized".

Overall, the following points emerge from the discussion on SpaceNet:

- SpaceNet allows for a detailed simulation of space logistics.
- Because of its event based nature, missions that are not executable (not feasible for logistical reasons) cannot be simulated and SpaceNet will shown this in form of a series of simulation errors.
- SpaceNet is written in Java and the only way a mission scenario can be generated is either using the GUI or loading a previous mission stored as an XML file.

## 2.2 Space logistics using network flows (GMCNF)

This section introduces the concept of GMNCF and the work that has been done around it for space mission planning application. The first part discusses static GMCNF. A example problem is used to illustrate the concepts and ideas presented. The advantages of this method and major drawbacks are explained and provide the transition to the second part. The second part discusses dynamic GMCNF. It builds upon the concepts presented in the static GMCNF. It is concluded by outlining the shortcomings of this method.

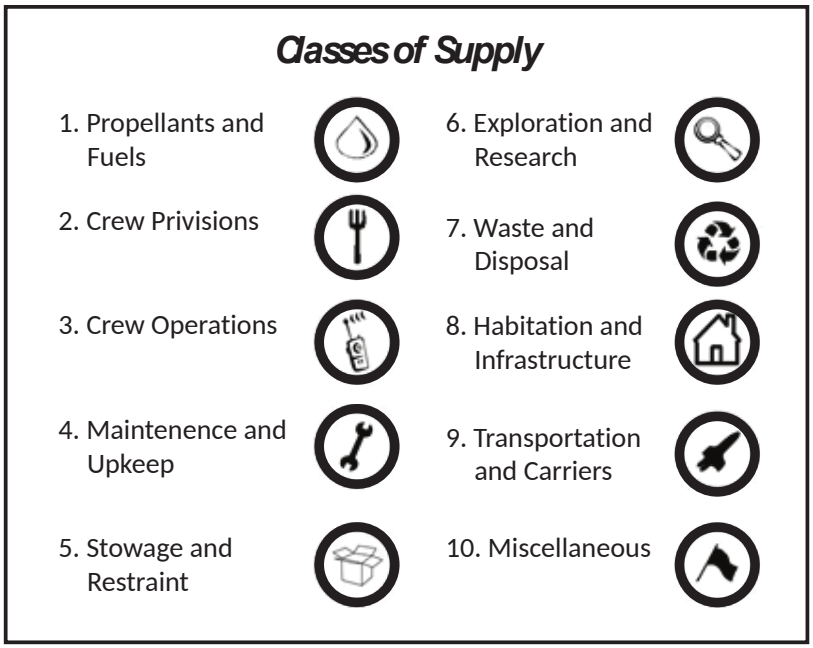



Figure 2-5: Classes of Supply (COS): classification of the different classes of material that are modeled in space logistics. Classes of Supply are found both in SpaceNet as well as GMCNF.

### 2.2.1 Static GMCNF

This section is based on the work presented in [37], [39], [40],[38]. In this work, GMCNF methods are applied to space logistics, however they have other applications such as electric vehicle infrastructure design [36] or energy/water infrastructure design [38], see fig.**??**. Other potential applications are discussed in 6.

The modeling of space system architecture with graph theory has also been investigated by others, such as [15]. Classical space mission architecture is an iterative process: at first, a set of trajectories is designed to to able to fly a mission architecture that the mission designer believes to be a good solution. Once the trajectories are designed, space infrastructure is designed around it, such as spaceships, launch vehicles, communication infrastructure both on Earth and in space etc. Once this is done, the trajectories are optimized a second time and the process goes back and forth between trajectory and vehicle design. GMCNF offers a different approach:

A planetary system can be viewed as a transportation network (see fig.2-9), with the nodes representing locations such as places on celestial bodies (like a crater of a launch complex) or specific orbits, such as LEO or Lagrange points. The edges connecting these nodes are

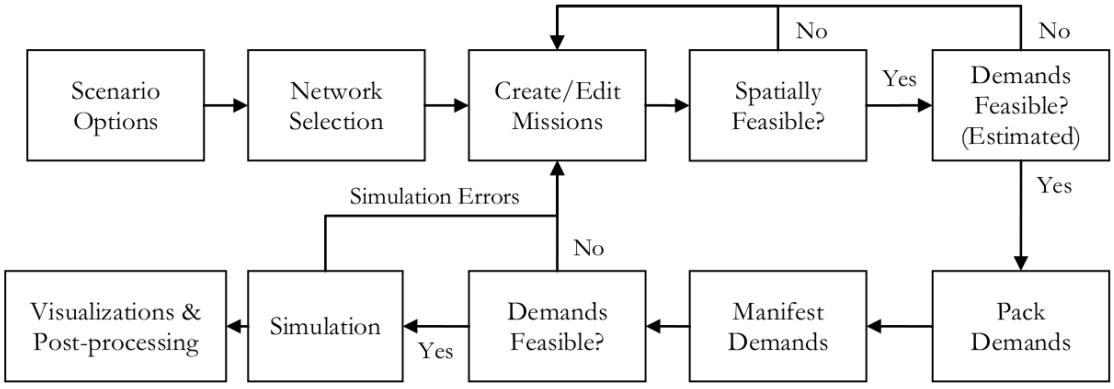



Figure 2-6: Process of creating a mission scenario in SpaceNet. Note that the user is expeceted to intervene in order to produce a working scenario. Image from [27].

trajectories, who's main characteristic is the $\Delta V$ required to fly that particular trajectory. The calculation of this value is left to astrodynamics. It is assumed that the network is static, e.g. the trajectories can always be flown with the same $\Delta V$. Notice that this method does not allow the optimization of a edge, but allows to choose which edge it is best to fly through.

The next step is the modeling of elements evolving within this network: human explorers, spacecraft, propellant, etc. can all be seen as flows going from one node to the next following the edges. This level of abstraction has effectively transformed the mission architecture problem into a network flow problem. Because multiple commodities have to be considered in order to model a space mission, the problem becomes a multi-commodity network flow problem. This class of problem is well known [14], [18] and has many applications [50], [65]. The goal is to minimize a quantity, such as IMLEO. This class of problem is called *the minimum cost flow problem*, and it can be put in the following form:

Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be a directed network defined by a set $\mathcal{N}$ of $n$ nodes and a set $\mathcal{A}$ of $m$ directed edges.

- Each edge $(i, j) \in \mathcal{A}$ has an associated *cost* $c_{ij}$ that denotes the cost per unit flow on that edge. Note that is it assumed that the flow cost varies linearly with the amount of flow $x_{ij}$.

- Each edge $(i, j) \in \mathcal{A}$ is associated with a *capacity* $u_{ij}$ that denotes the maximum

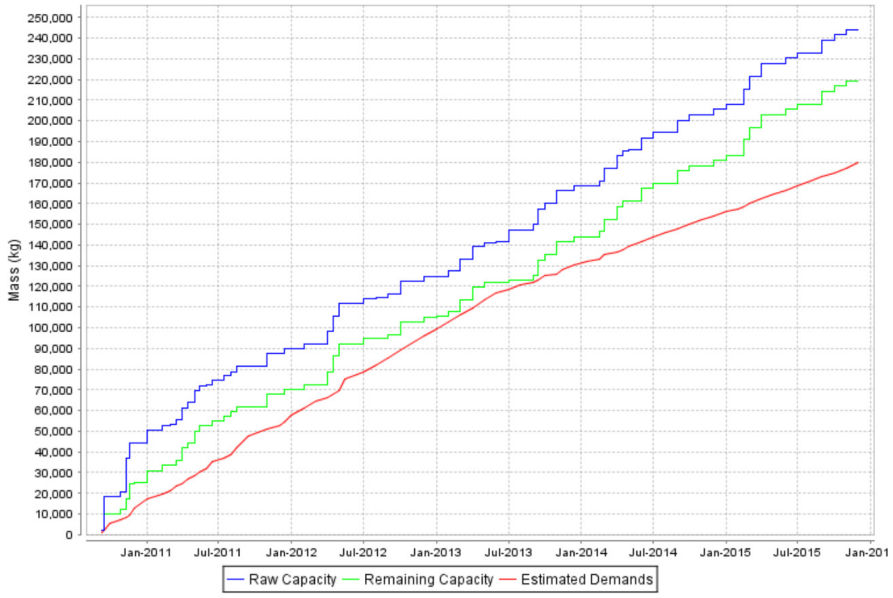
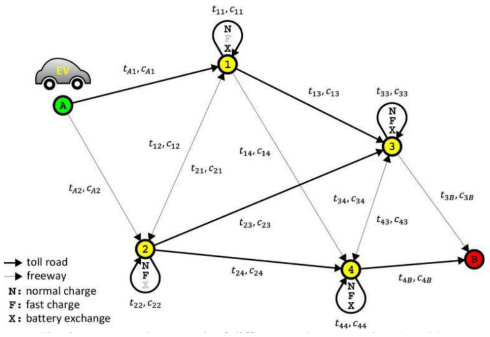



Figure 2-7: Effect of delivering a water recycling plant with better closure: when the system is delivered (on July 2013), the demands shrink. Image from [27].

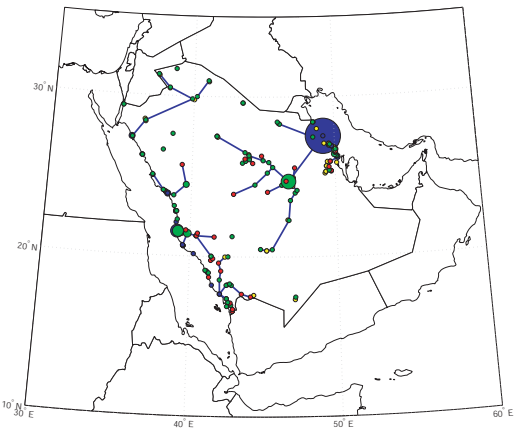
amount that can flow on the edge and a *lower bound* $l_{ij}$ that denotes the minimum amount that must flow on that edge.

- Each node $i \in \mathcal{N}$ is associated with a number $b_i$ representing its supply/demand. If $b_i > 0$, node $i$ is a *supply node*; if $b_i < 0$, node $i$ is a *demand node* with a demand $-b_i$; and if $b_i = 0$, node $i$ is a *transshipment node*.

The optimization variables in the minimum cost flow problem are edge flows which are represented by $x_{ij}$ for the edges $(i, j) \in \mathcal{A}$. This is represented in fig.2-10.




(a) GMCNF application to electric vehicles [36]




(b) GMCNF application to energy infrastructure [38]

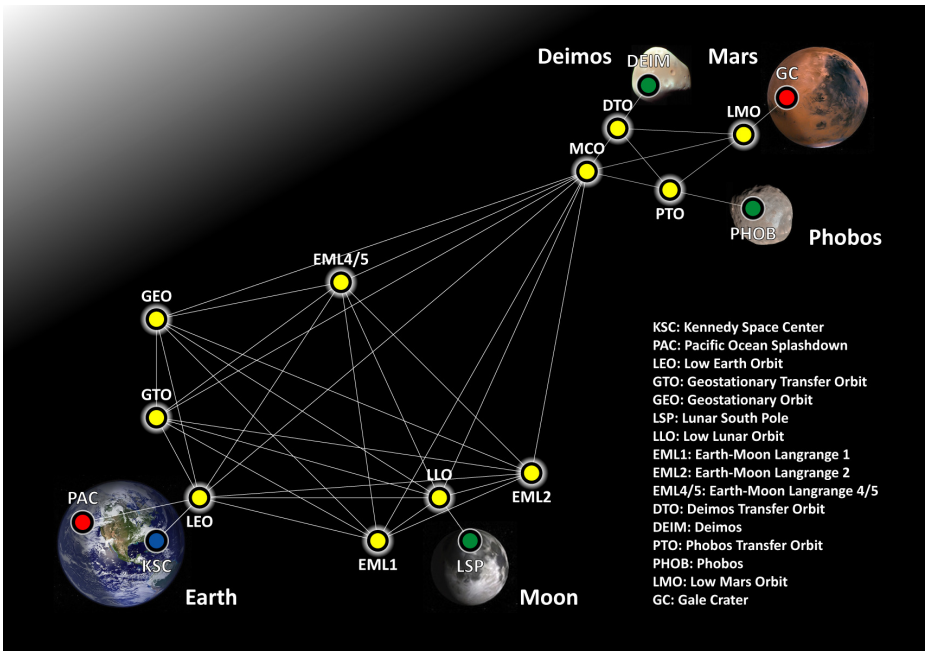



Figure 2-9: Example of a interplanetary transport network. Image from [37].

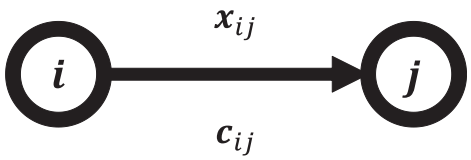



Figure 2-10: Nodes $i$ and $j$, directed edge $(i, j)$, and flow $x_{ij}$. Figure from [37].

The minimum cost flow problem is an optimization model formulated as follows:

Minimize

$$\mathcal{J} = \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} \tag{2.1}$$

subject to

$$\sum_{j:(i,j)\in\mathcal{A}} x_{ij} - \sum_{j:(j,i)\in\mathcal{A}} x_{ji} = b_i \qquad \forall\, i \in \mathcal{N} \tag{2.2a}$$

$$l_{ij} \leq x_{ij} \leq u_{ij} \qquad \forall\, (i,j) \in \mathcal{A} \tag{2.2b}$$

where

$$\sum_{i\in\mathcal{N}} b_i = 0 \tag{2.3}$$

However, this formulation is not sufficient for effectively modeling a space mission: flow is not necessarily conserved (i.e. losses along edges or in nodes). This can be included

into the model by adding a positive multiplier $\mu_{ij}$ with every edge and letting the sum of demands at the nodes be less of equal to the supply. If $0 < \mu_{ij} < 1$, the edge is *lossy*, and if $1 < \mu_{ij} < \infty$, the edge is *gainy*. In this model we assume that the lower bound on every edge flow is zero. This leads to the *generalized flow problem:*
Minimize

$$\mathcal{J} = \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} \tag{2.4}$$

subject to

$$\sum_{j:(i,j)\in\mathcal{A}} x_{ij} - \sum_{j:(j,i)\in\mathcal{A}} \mu_{ji}x_{ji} \leq b_i \qquad \forall\ i \in \mathcal{N} \tag{2.5a}$$

$$0 \leq x_{ij} \leq u_{ij} \qquad \forall\,(i,j) \in \mathcal{A} \tag{2.5b}$$

At this point, the problem still only possesses one single commodity flowing through the network. Multiple commodities can be included by applying the superposition principle to the problem, and having the constraints for the different commodities written in parallel, except for the maximum flow along an edge, where all commodities are bundled into a single one (i.e. the commodities share the same network), effectively linking all the commodities and uniting the problem:
Let $x_{ij}^k$ denote the flow of commodity $k$ (out of a set $\mathcal{K}$) on edge $(i,j)$, and let $c_{ij}^k$ denote the unit cost for commodity $k$ on edge $(i,j)$. Using this notation we can formulate the multi-commodity flow problem as follows:
Minimize

$$\mathcal{J} = \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} c_{ij}^k x_{ij}^k \tag{2.6}$$

subject to

$$\sum_{j:(i,j)\in\mathcal{A}} x_{ij}^k - \sum_{j:(j,i)\in\mathcal{A}} x_{ji}^k \leq b_i^k \qquad \forall\ i \in \mathcal{N} \text{ and } \forall\ k \in \mathcal{K} \tag{2.7a}$$

$$\sum_{k\in\mathcal{K}} x_{ij}^k \leq u_{ij} \qquad \forall\,(i,j) \in \mathcal{A} \tag{2.7b}$$

$$0 \leq x_{ij}^k \leq u_{ij}^k \qquad \forall\,(i,j) \in \mathcal{A} \text{ and } \forall\ k \in \mathcal{K} \tag{2.7c}$$

The "bundle" constraints in Eq. (2.7b) tie together the commodities by restricting the total flow of all commodities on each edge $(i,j)$ to at most $u_{ij}$. Note that we also impose individual flow bounds $u_{ij}^k$ on the flow of commodity $k$ on edge $(i,j)$.

If $\mathbf{x}_{ij}$ and $\mathbf{c}_{ij}$ are respectively the vector notations of $x_{ij}^k$ and $c_{ij}^k$ with respect to commodity, then the above formulation can be rewritten as:

Minimize

$$\mathcal{J} = \sum_{(i,j)\in\mathcal{A}} {\mathbf{c}_{ij}}^{\mathrm{T}} \mathbf{x}_{ij} \tag{2.8}$$

subject to

$$\sum_{j:(i,j)\in\mathcal{A}} \mathbf{x}_{ij} - \sum_{j:(j,i)\in\mathcal{A}} \mathbf{x}_{ji} \leq \mathbf{b}_i \qquad \forall\, i \in \mathcal{N} \tag{2.9a}$$

$$\mathbf{0} \leq \mathbf{x}_{ij} \leq \mathbf{u}_{ij} \qquad \forall\, (i,j) \in \mathcal{A} \tag{2.9b}$$

In some applications including space logistics, the edges do not have a maximum capacity (i.e. there is no upper limit to the amount that can be flow along an orbit). For this reason, the bundle constraints in Eq. (2.7b) are not imposed hereafter.

The final mathematical addition to the problem allows to model interactions between commodities including transformation as they flow along the network: by differentiating between a flow leaving a node (outgoing) and a flow a flow arriving (incoming) at a node, constraints can be written to impose transformations between the outgoing and incoming flow. This concept is represented in fig.2-11


Figure 2-11: Nodes $i$ and $j$, directed edge $(i,j)$, and flows $\mathbf{x}_{ij}^+$ (outgoing) and $\mathbf{x}_{ij}^-$ (incoming). Figure from [37].

This yields to the *Generalized Multi-Commodity Network Flow* problem:

Minimize

$$\mathcal{J} = \sum_{(i,j)\in\mathcal{A}} \left( {\mathbf{c}_{ij}^+}^{\mathrm{T}} \mathbf{x}_{ij}^+ + {\mathbf{c}_{ij}^-}^{\mathrm{T}} \mathbf{x}_{ij}^- \right) \tag{2.10}$$

subject to

$$\sum_{j:(i,j)\in\mathcal{A}} \mathbf{A}_{ij}^+ \mathbf{x}_{ij}^+ - \sum_{j:(j,i)\in\mathcal{A}} \mathbf{A}_{ji}^- \mathbf{x}_{ji}^- \leq \mathbf{b}_i \qquad \forall\, i \in \mathcal{N} \tag{2.11a}$$

$$\mathbf{B}_{ij} \mathbf{x}_{ij}^+ = \mathbf{x}_{ij}^- \qquad \forall\, (i,j) \in \mathcal{A} \tag{2.11b}$$

$$\mathbf{C}_{ij}^+ \mathbf{x}_{ij}^+ \leq \mathbf{0} \quad \text{and} \quad \mathbf{C}_{ij}^- \mathbf{x}_{ij}^- \leq \mathbf{0} \qquad \forall\, (i,j) \in \mathcal{A} \tag{2.11c}$$

$$\mathbf{0} \leq \mathbf{x}_{ij}^{+} \leq \mathbf{u}_{ij}^{+} \quad \text{and} \quad \mathbf{0} \leq \mathbf{x}_{ij}^{-} \leq \mathbf{u}_{ij}^{-} \qquad \forall\,(i,j) \in \mathcal{A} \tag{2.11d}$$

Three matrices are introduced: $\mathbf{A}_{ij}^{\pm}$, $\mathbf{B}_{ij}$, and $\mathbf{C}_{ij}^{\pm}$. $\mathbf{A}_{ij}^{\pm}$ is the *requirement matrix*, $\mathbf{B}_{ij}$ a *transformation matrix*, and $\mathbf{C}_{ij}^{\pm}$ a *concurrency matrix*. As can be seen from the constraints in Eqs. (2.11a) and (2.11b), $\mathbf{A}_{ij}^{+}\mathbf{x}_{ij}^{+}$ is required at node $i$ to send outflow $\mathbf{x}_{ij}^{+}$ into edge $(i,j)$, $\mathbf{x}_{ij}^{+}$ is transformed into $\mathbf{x}_{ij}^{-} = \mathbf{B}_{ij}\mathbf{x}_{ij}^{+}$, and $\mathbf{A}_{ij}^{-}\mathbf{x}_{ij}^{-}$ is received at node $j$ with inflow $\mathbf{x}_{ij}^{-}$ from edge $(i,j)$. Also in the concurrency constraints in Eq. (2.11c), the relationship between commodities on each edge $(i,j)$ is self-constrained such that the dot product with $\mathbf{C}_{ij}^{\pm}$ is less than or equal to zero.

With this modification, the following phenomena can be modeled:

- The flow gain and loss due to the interaction between commodities
- Transformation between commodities
- Additional requirements at nodes
- Concurrency on edges, i.e. the need of presence of certain commodities for another to flow.

If we consider $k$ types of commodities, that is, the decision variable vector includes $k$ components, then the $\mathbf{A}_{ij}^{\pm}$ and $\mathbf{B}_{ij}$ matrices must be $k$-by-$k$ square matrices while the $\mathbf{C}_{ij}^{\pm}$ matrix is a $l$-by-$k$ matrix, where $l$ is the number of concurrency constraints on edge $(i,j)$:

$$\mathbf{A}_{ij}^{\pm} = \begin{bmatrix} A_{11} & \cdots & A_{1p} & \cdots & A_{1q} & \cdots & A_{1k} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{p1} & \cdots & A_{pp} & \cdots & A_{pq} & \cdots & A_{pk} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{q1} & \cdots & A_{qp} & \cdots & A_{qq} & \cdots & A_{qk} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{k1} & \cdots & A_{kq} & \cdots & A_{1q} & \cdots & A_{kk} \end{bmatrix}_{ij}^{\pm} \tag{2.12a}$$

$$\mathbf{B}_{ij} = \begin{bmatrix} B_{11} & \cdots & B_{1p} & \cdots & B_{1q} & \cdots & B_{1k} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ B_{p1} & \cdots & B_{pp} & \cdots & B_{pq} & \cdots & B_{pk} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ B_{q1} & \cdots & B_{qp} & \cdots & B_{qq} & \cdots & B_{qk} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ B_{k1} & \cdots & B_{kq} & \cdots & B_{1q} & \cdots & B_{kk} \end{bmatrix}_{ij} \tag{2.12b}$$

$$\mathbf{C}_{ij}^{\pm} = \begin{bmatrix} C_{11} & \cdots & C_{1p} & \cdots & C_{1q} & \cdots & C_{1k} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ C_{r1} & \cdots & C_{rp} & \cdots & C_{rq} & \cdots & C_{rk} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ C_{l1} & \cdots & C_{lp} & \cdots & C_{lq} & \cdots & C_{lk} \end{bmatrix}_{ij}^{\pm} \tag{2.12c}$$

Note that if $\mathbf{A}_{ij}^{\pm}$ and $\mathbf{B}_{ij}$ are identity matrices and $\mathbf{C}_{ij}^{\pm}$ is a null matrix, the constraints in Eqs. (2.11a)-(2.11d) turn back to the constraints in the classical multi-commodity flow model in Eqs. (2.9a)-(2.9b). Also, if $\mathbf{A}_{ij}^{\pm}$ and $\mathbf{B}_{ij}$ are diagonal matrices and $\mathbf{C}_{ij}^{\pm}$ is a null matrix, there are no interactions between commodities and therefore the problem can be decoupled into $k$ independent generalized flow problems. To put it the other way around, the off-diagonal entries of $\mathbf{A}_{ij}^{\pm}$ and $\mathbf{B}_{ij}$ and non-zero entries of $\mathbf{C}_{ij}^{\pm}$ indicate that there are interactions between commodities. For example, the $(p, q)$ entry of $\mathbf{A}_{ij}^{+}$ means that if we send 1 unit of the $q$th commodity, we also need $[A_{pq}]_{ij}^{+}$ units of the $p$th commodity. The $(r, p)$ entry of $\mathbf{C}_{ij}^{+}$, which is $[C_{rp}]_{ij}^{+}$, is the $p$th commodity's contribution to the $r$th concurrency constraint. The next section presents an example problem to illustrate how a space mission using ISRU can be modeled using a GMCNF problem and solved.

## 2.3 Example Problem

The proposed example is that of a simplification of a Mars exploration campaign:
The network is comprised of 4 nodes (see fig.2-12): Earth (**E**), low earth orbit (LEO or **L**), the Moon (**M**) and Mars (**S**).
The nodes are linked by the following arcs: Earth-LEO, LEO-Earth, Mars-LEO, LEO-Mars,

Moon-LEO, LEO-Moon and Mars-Mars. This topology enables all types of phenomena that appear in GMCNF problems, such as graph loops (in particular that going to and from the Moon) and commodity transformation (on Mars, where outgoing crew are transformed into incoming crew). More arcs are not added, as to keep the problem as simple as possible.
The last step in defining the problem is the choice of commodities flowing through it: because we want to be able to represent transformation at a node (such as transforming an outgoing crew to a returning crew), 2 commodities are already included into the problem. If we want to be able to observe a trade-off between paths depending on ISRU for example, a third commodity must be included, in this case some form ISRU product. This product can also be used to model lossy or "gainy" arcs, so we choose the ISRU product as being some generalized form of fuel. However, if we want the concurrency matrices to be none zero, the fuel must be carried along the arcs with some form of hardware. In order to avoid over-complicate the problem, we assume a generalized hardware flow, that engulfs the crew. We thus have the following commodities:

$$\mathbf{x}_{ij}^{\pm} = \begin{bmatrix} \text{outgoing hardware} \\ \text{returning hardware} \\ \text{propellant} \end{bmatrix}$$

For sake of symmetry in the problem, we add an transformation arc on the Moon. This arc may be used to for example to "exchange" outgoing hardware for fuel (at a rate which does not have to be one to one), simulating an ISRU production plant. The network described is depicted below, with a numbering of the arcs.

Now that the topology has been set up, the different arc properties (or $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ matrices) can be set up. Let us point out that the following numbers are not supposed to match exactly those of a true Mars campaign study, but rather be close enough to illustrate phenomena such as trade offs between flying directly to Mars and to fly by some other location (in this case the moon) in order to collect fuel.

**Construction of the requirement $\mathbf{A}_{ij}^{\pm}$ matrices:** The $\mathbf{A}_{ij}^{\pm}$ matrices decribe the commodities that are additionally required by sending out or recieving the flow but do not flow on the edges themselves. Thus this matrix concerns the nodes, and not the edges like the $\mathbf{B_{ij}}$ and $\mathbf{C_{ij}}$ matrices. It can be incorporated in the simplified space

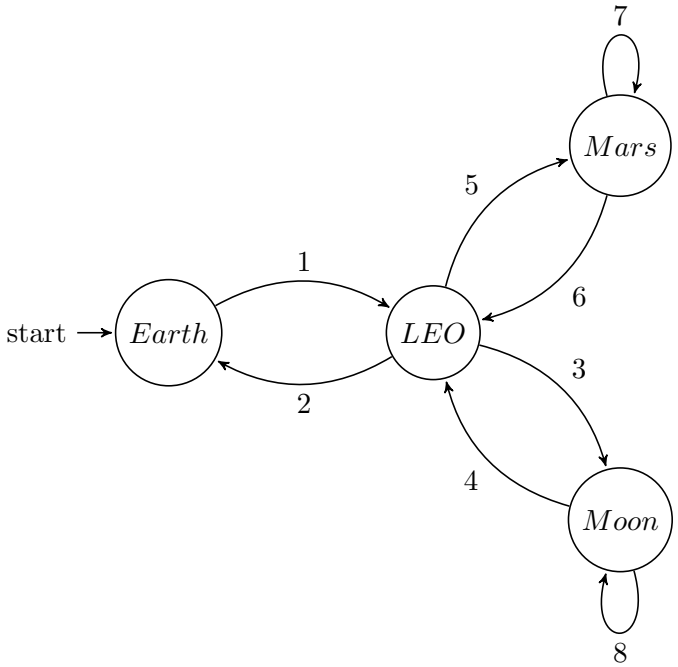



Figure 2-12: Graph representation of the basic space logistics problem: The starting point is the Earth (E), which is a source of vast amounts of outgoing hardware, but is a returning harware sink, forcing some hardware to flow to Mars (S) where it can be transformed (with the Mars-Mars loop) to returning hardware that flies back. This is done by transiting Low Earth Orbit (LEO), or even flying by the Moon (M) where resources can be collected by ISRU. The ISRU generation depends linearly on the amount of hardware that can be sent to the Moon, where is transformed via the Moon-Moon loop.
Note that this is a "toy problem", it not intended to represent the true Earth-Mars-Moon system, but rather a simple problem on which representative phenomena can be modeled and tested.

logistics problem by stating that the returning hardware must be accompanied by some extra mass, say for example returning rock samples. The rock-samples may be modeled by outgoing hardware:

$$\mathbf{A}_S = \begin{bmatrix} 1 & \xi & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Where $\xi$ represents the mass fraction of return samples to returning hardware. Although the $\mathbf{A}_S$ matrix is shown to illustrate the example, it will not be discussed further: the sole effect of $\xi$ is to increase the required mass of outgoing hardware proportionally to $\xi$. Because this effect is of no particular interest in this case, thus $\xi$ is kept to 0 throughout the rest of this work.

**Construction of the transformation $\mathbf{B_{ij}}$ matrices:** In this model, the following transformations occur:

- The transformation of "outgoing" crew into "returning" crew along the Mars-Mars arc. Because there is no loss or gain of crew in the process, and that the whole crew is to return, this process can be decribed in the following way:

  $$\mathbf{B}_{S-S} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

  Note that there is no remaining outgoing hardware after the process. This is not a problem, because the optimizer chooses how much hardware transits this arc, e.g. all the outgoing hardware must not be processed to returning hardware at the node. Note that the diagonal term for returning hardware is zero: this avoids spontaneous creation of mass flowing through the arc, e.g. the arc would feed itself producing non-physical results.

- The transformation, or burning, of propellant along the transportation arcs. The propellant mass fraction can be derived from the rocket equation, yielding:

  $$\phi_{ij} = 1 - \exp\left(-\frac{\Delta V_{ij}}{I_{sp} g_0}\right) \tag{2.13}$$

| edge | $\Delta V_{ij}$ $[km/s]$ | Isp $[s]$ | $\phi_{ij}$ $[-]$ |
|---|---|---|---|
| 1 | 9 | $263^a$ | 0.969 |
| 2 | $0.17^b$ | $314^c$ | 0.053 |
| 3 | $4.32^d$ | 314 | 0.754 |
| 4 | 4.32 | 314 | 0.754 |
| 5 | $10.2^e$ | 314 | 0.754 |
| 6 | 10.2 | 314 | 0.754 |
| 7 | N.A. | - | 0 |
| 8 | N.A. | - | 0 |

Table 2.1: Isp: $^b$: shuttle crew operations manual, $^d$: Apollo $\Delta V$ budget $^c$: Isp of the Apollo service module, $^a$: Isp of the Saturn V first stage $^e$

Where $\Delta V_{ij}$ is the change in velocity of the vehicle on edge $(i, j)$, $I_{sp}$ the specific impulse and $g_0$ the standard gravity. Let $m_{dr}$ be the dry mass of the vehicle and $m_{pr}$ the propellant mass. When $\Delta m_{pr}$ is consumed during a propulsive burn, the propellant fraction is given by:

$$\phi = \frac{\Delta m_{pr}}{m_{dr}^{+} + m_{pr}^{+}} \tag{2.14}$$

Thus the propellant mass right after burn becomes:

$$m_{pr}^{-} = m_{pr}^{+} - \Delta m_{pr} = -\phi m_{dr}^{+} + (1 - \phi) m_p r^{+} \tag{2.15}$$

Since the dry mass does not change during the burn and that the dry mass is the sum of outgoing and returning hardware, the transformation matrix $\mathbf{B_{ij}}$ becomes:

$$\mathbf{B}_{ij} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\phi & -\phi & 1 - \phi \end{bmatrix}$$

The values of $\phi$ for different edges are shown in the table 2.2:

- Because we choose to model the ISRU plant by a transformation of all outgoing hardware to fuel (i.e. no fuel is left once the process has been executed, but a quantity $\chi$ of propellant is created), this accounts for the third type of transformation. Let us define $\chi$ as the ratio between outgoing hardware flow and mined propellant flow. This value symbolizes the ISRU mass efficiency, and this value

will be varied in order to illustrate how the optimum network changes according to this parameter:

$$\mathbf{B}_{M-M} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ \chi & 0 & 0 \end{bmatrix}$$

Again, notice that because this matrix is on a graph loop, the diagonal term is left out to avoid the loop feeding itself.

**Construction of the concurrency $\mathbf{C}_{ij}^{\pm}$ matrices:** The concurrency matrices appear when a given flow-type cannot exist with the concurrency of another, for example a flow of crew cannot exist if a flow of spacecraft is not present. In the present problem, we will enforce the condition that propellant cannot travel on itself, e.g. it needs a spacecraft (modeled by outgoing hardware) to fly with it:

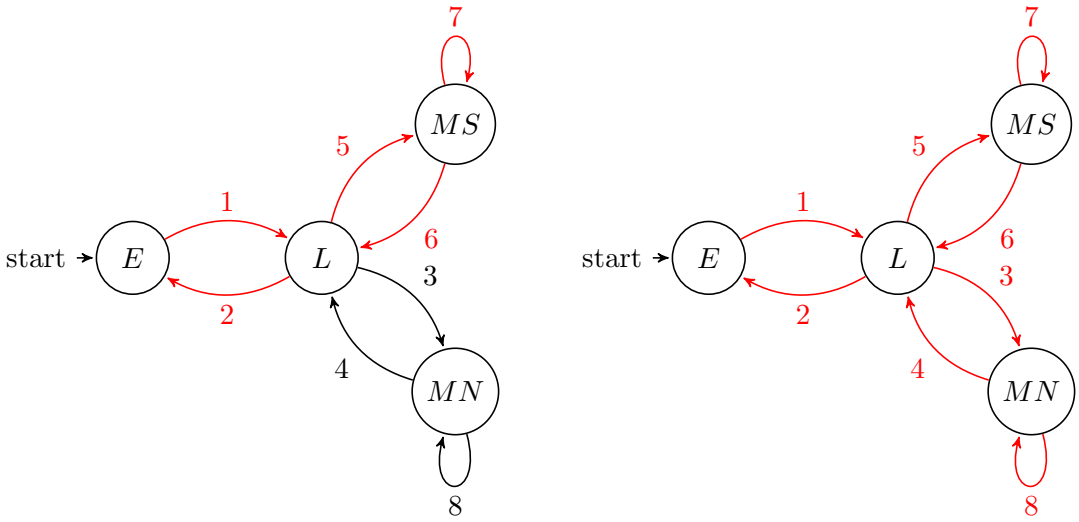
$$\mathbf{x}_{ij}^{\pm} = \begin{bmatrix} \text{outgoing hardware} \\ \text{returning hardware} \\ \text{propellant} \end{bmatrix}, \mathbf{C}_{ij}^{\pm} = \begin{bmatrix} -1 & 0 & \eta \end{bmatrix}$$

**Characterization of the edges:** The edges in a GMCNF problem have a maximum and minimum flow quantity. In this case, the minimum quantity is chosen to be zero, and the maximum quantity is chosen large enough as to not create any constraint on the optimum, e.g. large enough to be ignored.

#### 2.3.0.1 Physical insight to the example problem

The problem is constrained to return a quantity (3 units mass of returning hardware) to the $E$ node. The optimizer has 2 ways of doing this: either it chooses a direct path, namely from $E$ to $L$ to $MS$ and back (see fig.2-13 left) and not take advantage of the possibility of collecting propellant on the $MN$ node,

Or it sends some hardware to the $MN$ node to collect propellant to supply the $L$ node and have an overall lower mission cost (in this case, the total amount of mass being sent out of $E$), see fig.2-13 right. What is investigated by varying the parameters $\phi$ and $\chi$, is the point at which the optimal network switches from that represented in fig.2-13 left to that




(a) Optimal path (red) of commodities when $\chi < 7$

(b) Optimal path (in red) of commodities when $\chi > 7$

Figure 2-13: Representation of the possible paths that can satisfy the example problem: either a direct path (from $E$ to $L$ to $MS$) or an indirect path (going via $MN$ in order to collect propellant). This trade off is qualitatively the same as seen in full space mission optimizing when ISRU is considered.

represented by the right side. The results will show that this switch occurs at $\chi = 7$ for an Isp of 450 s and $\eta = 0$.

### 2.3.1 Numerical implementation

This small problem is written in Python to afford ease of experimentation, and because Python is easily connected with different optimizers, including commercial and open-source ones.

The code structure is the following:

- a file *network.py* which serves as a library for the classes and problem specific functions.
- an input file *inputFile.py* in which the user defines the problem at hand. This file also calls different functions from *network.py* in order to prepare the LP problem. This file in run by the main file *main.py*
- a main file *main.py* which runs the input file and runs the LP problem. The results are then displayed by specific display functions at the end of the file.

#### 2.3.1.1 Results and discussion of the example problem

As mentioned, one of the goals of the example problem is to compare optimizer performance. I has to be noted that from the three optimizers tested (Python's `linprog`, Matlab's `linprog` and CPLEX) only CPLEX displayed satisfactory results:

- Python's `linprog` ignored some of the constraints in the problem, producing incorrect results
- Matlab's `linprog` produced correct results, but only after an unsatisfactory time period.

One of the goals of this research is to incorporate the GMCNF optimization into SpaceNet, an open software. The experience given by the example problem shows that given the size of the GMCNF problem, *this cannot be done reliably using open-source or even non-specialized optimization software.*

The example problem is designed to be simple enough to understand without any computational processing:

When there is no ISRU, the optimal path for the space logistics problem is trivial: the network goes through LEO and directly to Mars (see fig.2-13). However, it becomes interesting to see at what point of ISRU efficiency the logistics network changes and goes by the moon. This phenomena of changing logistics networks with respect to parameters such as ISRU extraction rate or propellant boil-off rate is extensively investigated in [37].

The problem is solved for a range of ISRU extraction rates ($\chi$) and specific impulses ($\phi$) or vehicle mass fractions ($\eta$). The results are presented in forms of plots of flow magnitude of a certain commodity in a given edge with respect to ISRU extraction rates.

At first, a base configuration is solved, where $\phi$ and $\chi$ are varied, and where the mass fraction for the vehicles $\eta$ is zero, and the returning sample mass fraction $\xi$ is equally zero.

In the section where the discretized version is studied, $\eta > 0$ ($\eta = 0.1$), producing a time-paradox (see paragraph 3.1.3). However, this cannot be observed in the case presented below:

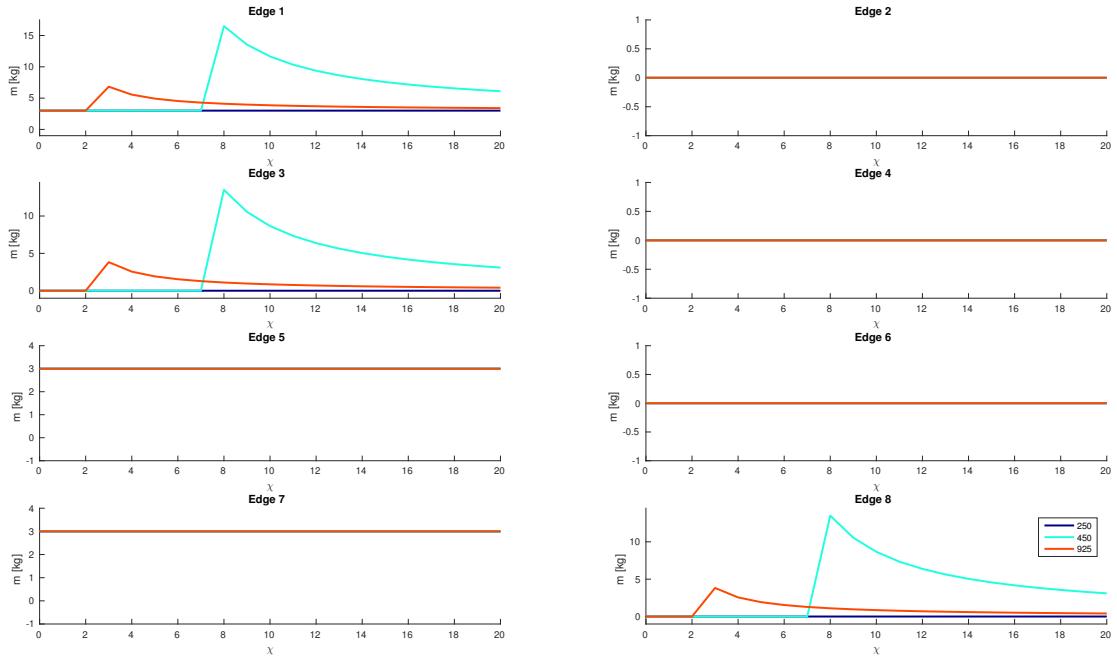



Figure 2-14: Flows of outgoing hardware (commodity 1) for the optimal reference problem: edge numbers are the same as in fig.2-12. Notice that when an edge is not used, that flow is zero. The network topology changes at the peaks: as the ISRU efficiency increases, the optimal solution goes over the $M$ node to harvest fuel. Once the switch is made, the amount of hardware decreases because the ISRU demands mass for the same quantity of fuel produced. Notice that the specific impulse alters the point at which the network swtiches topology. The different curves correspond to specific impulses: $250s$ (low efficiency chemical propulsion), $450s$ (current chemical propulsion) and $925s$ (nuclear thermal rocket propulsion).

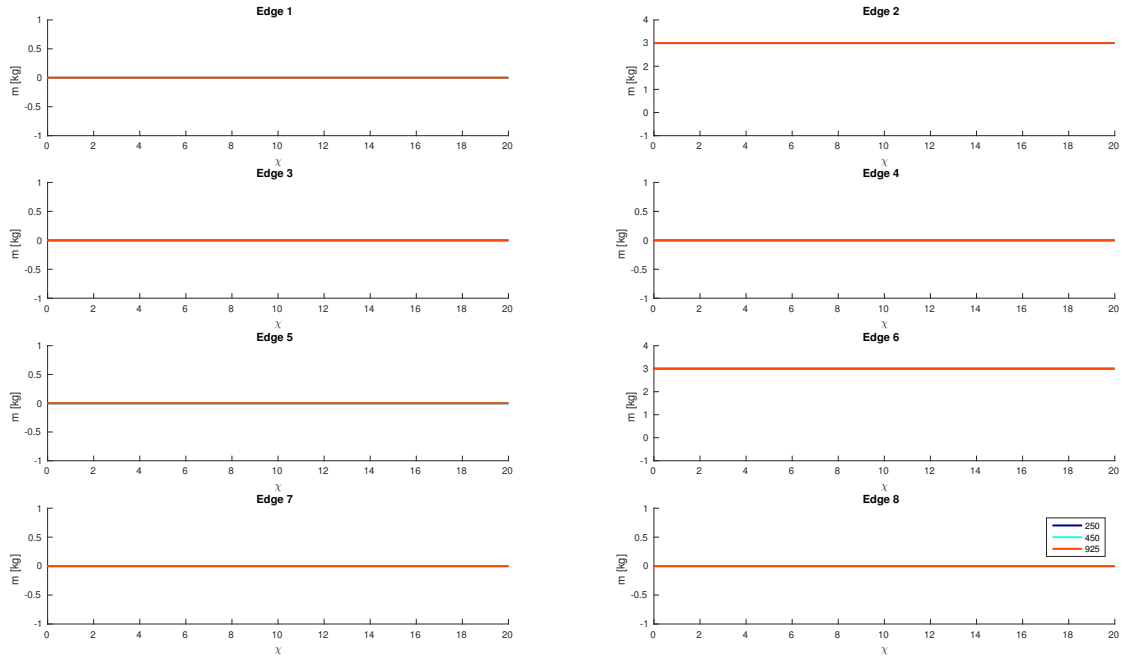



Figure 2-15: Flows of returning hardware (commodity 2) for the optimal reference problem: edges are the same as in fig.2-12. Because the constraint of 3 mass units (in this example $kg$) of returning hardware is not altered by the increase in ISRU capability or specific impulse, the lines remain constant. The different curves correspond to specific impulses: $250s$ (low efficiency chemical propulsion), $450s$ (current chemical propulsion) and $925s$ (nuclear thermal rocket propulsion).

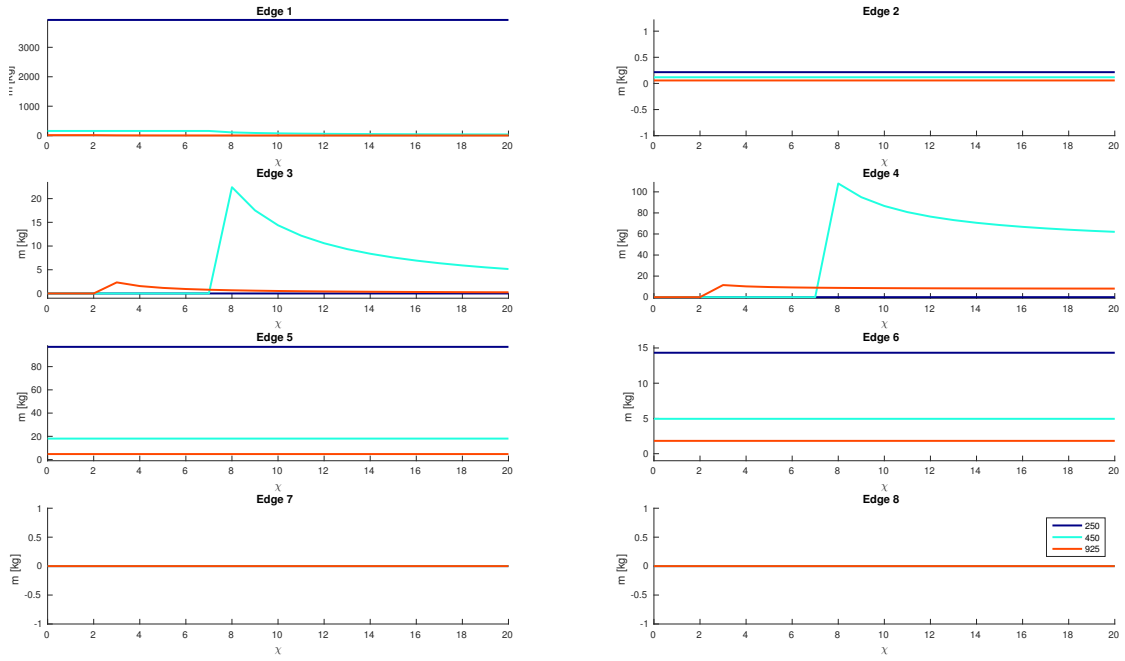



Figure 2-16: Flows of propellant (commodity 3) for the optimal reference problem: edges are the same as in fig.2-12. The different curves correspont to specific impulses: $250s$ (low efficiency chemical propulsion), $450s$ (current chemical propulsion) and $925s$ (nuclear thermal rocket propulsion). Notice the severe mass increase as $Isp$ drops to $250s$.

[37] and the example problem illustrate the following elements:

**Modeling capability** GMCNF can effectively model at high level the logistics of a space mission. Insight of a complexe problem such as planning a mission to Mars can be gained even through a simple model such as the example problem presented here.

**Optimization capability** This method allows for a global optimization of the transportation network. The result of a GMCNF optimization allows to determine optimal path through the spatial network as well as the high level requirements (sizing) of the elements flowing through it, as long as these do not deviate from linear models.

**Computational constraints** As shown by the example problem, non-specialized or open-source software do not produce satisfactory results, suggesting that a seamless integration of GMCNF optimization capability in an open-source code is not possible.

**Non linearity between parameters** The results show that although the model is fully linear, the relations between the parameters of the problem are highly non linear: as

can be seen in fig.2-14 to fig.2-16, there is a sudden topology switch at $\chi = 7$. A even simpler example is discussed in Appendix A where an analytical formula is given for the switch between optimal network topologies.

**Major drawbacks** despite remarkable properties, GMCNF optimization suffers from two major drawbacks:

The first is its static nature: the network is set up as if it were in a steady state. This might work very well for infrastructural planning needs, but certainly does work in the space logistics case when initial no infrastructure is deployed, as it is prone to generating time-paradoxes. These are situations where for example resources are used at a point A to enable production at a point B which in turn supplies A, i.e. the resources are consumed before they are mined. This is pointed out in [37], and addressed by [**?**] This can be solved by making the network dynamic, as will be presented in the next section.

The second is its linear nature: because it relies on linear constraints to function, the result is likely to not be close to what is feasible in terms of hardware design (i.e. the results are too idealistic). The next section describes a dynamic GMCNF formulation in order to address the time-paradox issue:

## 2.4 Dynamic GMCNF

One of the major drawbacks of the classical GMCNF framework is it's static or steady-state nature: optimal static solutions might be infeasible because the contain time-paradoxes. Addressing this problem implies taking into account the dynamic aspect of the problem. This was first carried out in [33], which was followed up by [35],[34].

The approach used in [33] is to use a time expanded version of the network (an example of a time-expanded network can be seen in 2-17. Unfortunately, a simple time expansion makes the problem so large that it becomes impossible to solve.

In order to find a solution, two methods are developed in [33]:

**First approach:Limiting the solution between upper and lower bounds:** As will be explained in the lower paragraph, there exist ways of calculating at lower cost an upper bound and a lower bound of the optimization problem. The bounds are calculated

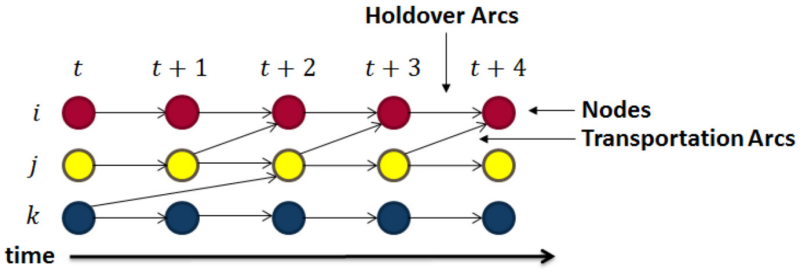



Figure 2-17: Example of a time-expanded network: the static equivalent are nodes $i$,$j$,$k$. Notice that the expansion process makes the network much larger, both in nodes (at most $N \times n_t$, where $N$ is the number of nodes and $n_t$ the number of time steps) and edges (at most $N^2 \times n_t$). Figure from [33]

using a network that have undergone a preprocessing operation in order to shrink the optimization problem. These operations are:

**Node/arc aggregation** The lower bound can be found with a low computational effort by optimizing a *node/edge aggregated network*. The basic concept is to combine multiple nodes as one group and bundle the edges that come into or from any nodes in that group together as shown in fig.2-18. Note that all group nodes resulting from aggregation contain self-loops that are aggregation of holdover edges. This also means that the resulting formulation from node/edge aggregation still potentially contains time paradoxes introduced above. The mathematical proof that a lower bound for the optimal solution as well as the full mathematical formulation can be found in [33]

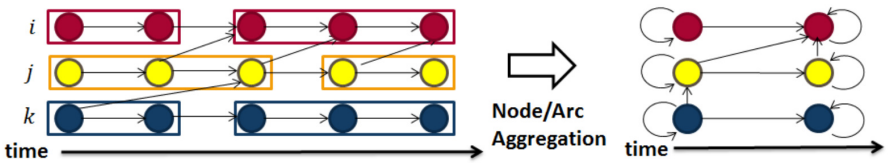



Figure 2-18: The node/edge aggregating process: as the aggregating algorithm [33] goes through the network, nodes and edges are aggregated reducing the size of the network and lowering the computational needs to solve the problem. Image source: [33]

**Second approach: Node/arc restriction** As mentioned, a lower bound of a full time-expanded GNCNF problem can be found with a low computational effort using node/arc aggregation, but the resulting solution can be infeasible for that full time-expanded GMCNF problem because of the cycles formed during ag-

gregation. In order to find a feasible (probably suboptimal) solution, an upper bound needs to be found. This can be done by eliminating multiple transportation arcs and restrict the number of transportation opportunities. However, only restricting transportation arcs does not reduce the numbers of constraints and variables significantly because a large number of holdover arcs still exist. Therefore, multiple nodes are eliminated (i.e., node time windows are restricted) after the transportation arc restriction step. This is depicted in fig.2-19. Because in this process no cycle are created within the network, the solution is guaranteed feasible. The mathematical proof that a lower bound for the optimal solution as well as the full mathematical formulation can be found in [33]

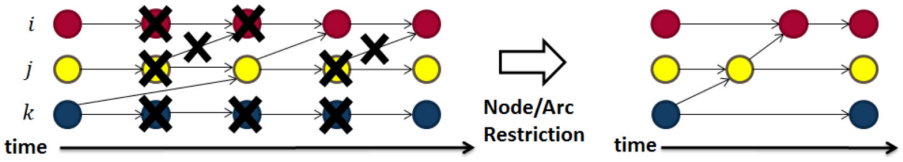



Figure 2-19: Image source: [33]

Using these two treatments on the network, the optimal value of a feasible network may be found. A summery is given in fig.2-20: the feasible optimum is bounded by the restricted solution (guaranteed feasible) and the aggregated solution (likely to be unfeasible). The static network can be seen as a super aggregated network, and thus defines a lower bound (i.e. most optimistic) to all solutions.

Note that this method has two major drawbacks:

- Although it is bounded, the feasible optimal value is not calculated
- The feasible optimal solution is not produced by the optimizer: in the space mission optimization case, the optimal mission is still to be found.

Despite these drawbacks, these methods can give insight into understanding how the network behaves and may give the inputs for the second class of methods:

**Cluster-Based Heuristic Methods** In cases when the time-expanded network can be clustered, especially when the link between specific nodes can only occur a precise intervals (like a launch window), the time can be made of variable size:

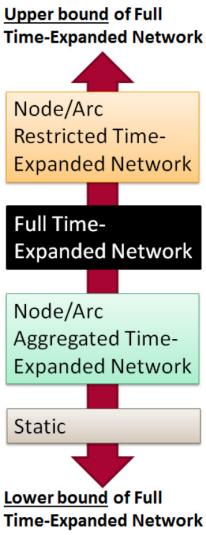



Figure 2-20: Relative positions of the results of the optimization of time-expanded GMCNF: the restricted network produces an upper bound (feasible, but suboptimal), the aggregated network produces a lower bound (infeasible). The feasible and optimal solution is located between both. The static GMCNF can be seen as a "super aggregated network". Image source: [33]

- low-frequency steps between the clusters, where the size of the time-step is driven by the physical constraints in the network, such as a Martian Launch window for example.
- high-frequency steps within a cluster, where the time-step size is dictated by the temporal resolution desired. For example, within the Earth-Moon system when launch opportunities are almost always present.

An example of this is depicted in fig.2-21

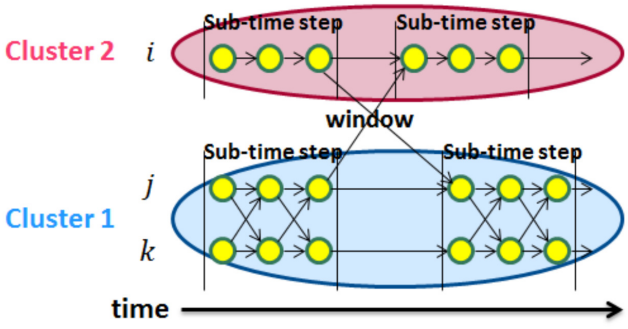



Figure 2-21: Depiction of a clustering process of a time-expanded transport network. Note the dual time-scale: one long time-scale between clusters, and a short time-scale within the clusters. Image source: [33]

Although the temporal bi-scale clustering is an effective method and it produces a solution and not just it's bounds, it remains expensive when considering network that expand over longer periods of time.

If the time step of the clusters tends to zero, effectively emitting the hypothesis that within a cluster phenomena happen instantaneously in comparison to the travels between clusters, the *semi-static time-expanded GMCNF* emerges. This is illustrated in fig.2-22. This reduces substantially the computational cost of long duration networks, and brings problems such as Mars mission optimization within the capability of a laptop computer. Semi-static GMCNF is used throughout the rest of this work.

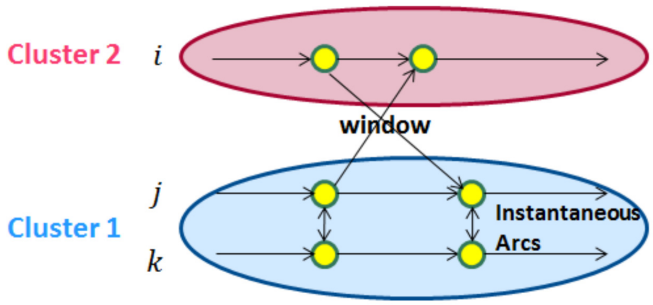



Figure 2-22: Depiction of a partially static time-expanded network. In [33], it is found that this method is particularly effective for the space mission application. This type of network is the one used for all time-expanded networks in this thesis. Image source: [33]

### 2.4.1 Shortcomings of the GMCNF applied to Space Mission Design

The first drawback to be pointed out is the size of the optimization problem: for their work, [37], [33] used the licensed software CPLEX in order to solve properly the problem. Any other free software tested did not yield satisfactory results. *This excludes any open-source application of the GMCNF formulation as we know it.*

For this work, the licensed software Gurobi was tested because of its advertised superior optimization capabilities. The CPLEX link to Matlab differs in implementation between [37] and [33], thus both are compared in order to verify the results of the Gurobi output.

Although there seems to be no difference between the solutions in terms of objective function value and consequently of solution vector, there is a substantial one, as shown in fig.2-23. Because both vectors satisfy the constraints and show the same objective function value, there is no reason one should be preferred to another.

This points to the fact that GMCNF methods are very sensitive to the optimizers used. This

| Optimizer | CPLEX [33] implementaion | CPLEX [37] implementaion | Gurobi |
|---|---|---|---|
| Verifying constraints* $max(Ax_{opt} - b)$ | $1.3145e-11$ | $1.3145e-11$ | $5.0733e-12$ |
| Verifying Objective function $c^T x_{opt}$ [kg] | $1.0310e+05$ | $1.0310e+05$ | $1.0310e+05$ |
| Computing time [s] | 35.58 | 39.91 | 11.75 |

Table 2.2: Optimization results for a NEO only scenario.
*: the constraints are of the form $\mathbf{A}x \leq b$, thus is the constraints are met then $max(\mathbf{A}x-b) \leq 0$. Note that the very low value show that all constraints are met, as the values shown are little over zero.

also means that there potentially exist many feasible solutions close to the global minimum, opening the door to other analyses such as iso-performance or other types of trade studies.

Although appealing, the static GMCNF formulation suffers from time-paradoxes, i.e. situations that can only be physically realized if the network is fully deployed and impossible when the network is not deployed. This is corrected by time-expanding the network [33], leading to a dynamic GMCNF network.

However, for a transportation network to be feasible, certain properties must be verified, in particular in the space logistics case. One of these properties is linked to the spectrum of the mass flows: commodities representing hardware, such as inert rocket mass, tank mass etc. should possess flows which rarely change value as they move through the network. Changing as they move through the network corresponds to either jettisoning a part of the hardware while the rest remains functional, or adding a part to the ensemble maintaining the whole functional. This equates to a "rubber" spacecraft, that resizes itself according to the needs of the network. Albeit the fact that this is not a feasible solution, it becomes difficult to follow a spacecraft through the network, as the flow that constitutes it separates, and merges with other flows. If the number of merges and seperations (or the different flow values) become too large, the tracking becomes completely untraceable. The following figures are generated from the results of [33]:

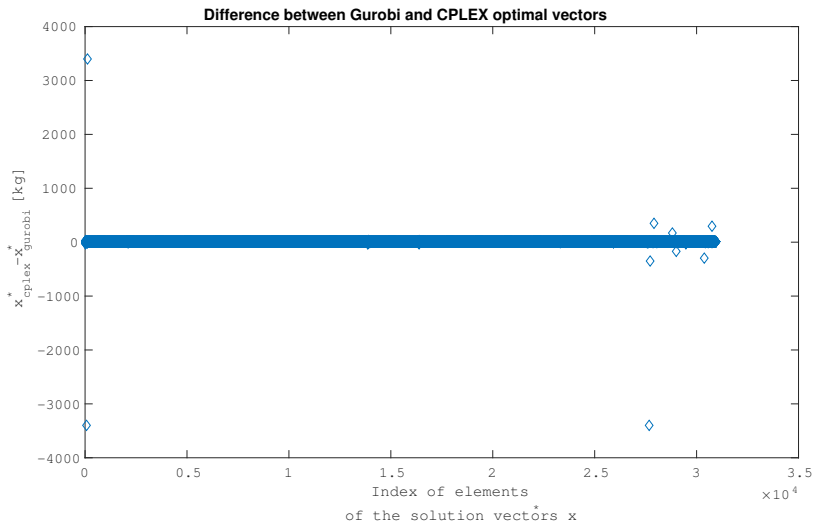



Figure 2-23: The difference between both optimal vectors $x^*_{cplex}$ and $x^*_{gurobi}$ show that although the objective function value is the same, the solution vectors may differ substantially in some parts. Note that most elements are identical between both vectors (most elements between the vectors are identical and thus show a zero difference). Note how this graph is almost symmetrical along the $y = 0$ axis: this is because the values of the different components are identical between the CPLEX result and the Gurobi result, but they are shifted in position by a small amount. This reflects differences in schedule in the mission: if a given flight taking place in both optimization results differs slightly in the date at which it is done, then the differences between the two solutions are non zero and show this behavior seen above. This happens because the sensitivity of the objective function to these slight changes in schedule is close to zero, and thus note captured by the optimizer.

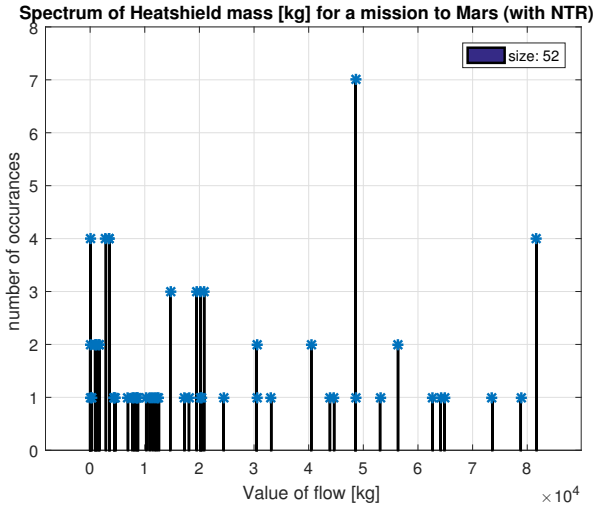



Figure 2-24: This chart depicts the values of the different heat-shields used in the Mars only scenario described in [33]. The "size" indicates the number of different heatshield mass values throughout the mission (or the number of data points in this case).

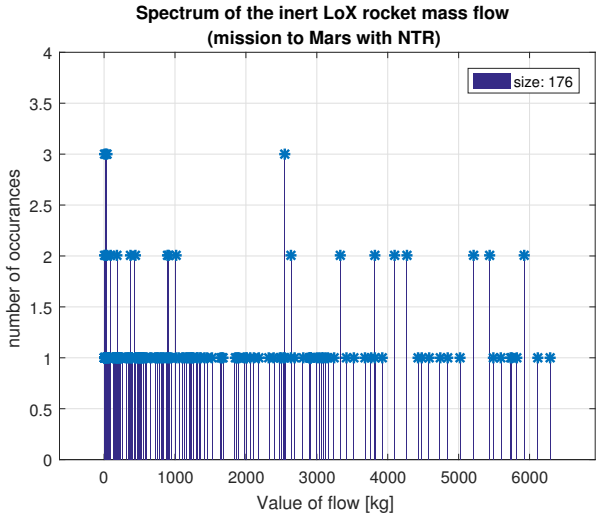



Figure 2-25: This chart shows the same as the previous one, except that it does this for the size of LH2/LOX rocket engine size along an edge. This figure means that 176 different sized rocket engines must be built to fly this mission. One could argue that a cut-off size be chosen, in order to keep the large elements driving the network and neglecting the small flows. Unfortunately this cannot work as the rocket engine sizes are distributed across the mass spectrum.

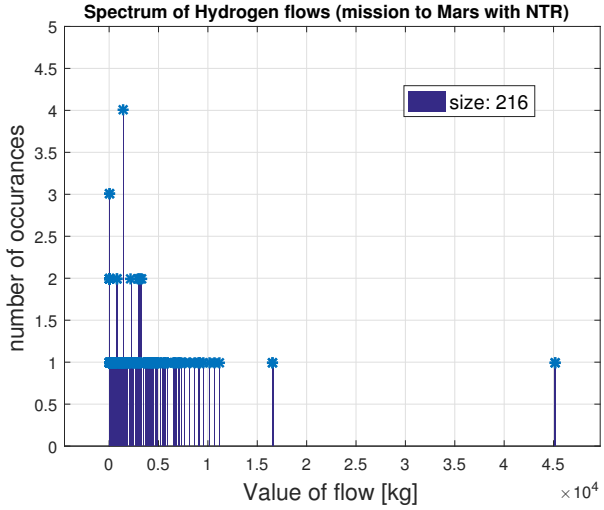



Figure 2-26: This chart shows the same as the previous one, except that it does this for the amount of hydrogen carried along an edge. Note that the GMCNF will size the spacecraft and hardware around these values, thus their number is an indicator of the number of times a vehicle assembly is reconfigured. Notice the continuous variation between 0 kg and $1 \times 10^4$

These results show that in its current state, *due to the complete linear nature of the the dynamic GMCNF formulation cannot be used to design practical or executable mission architectures.*

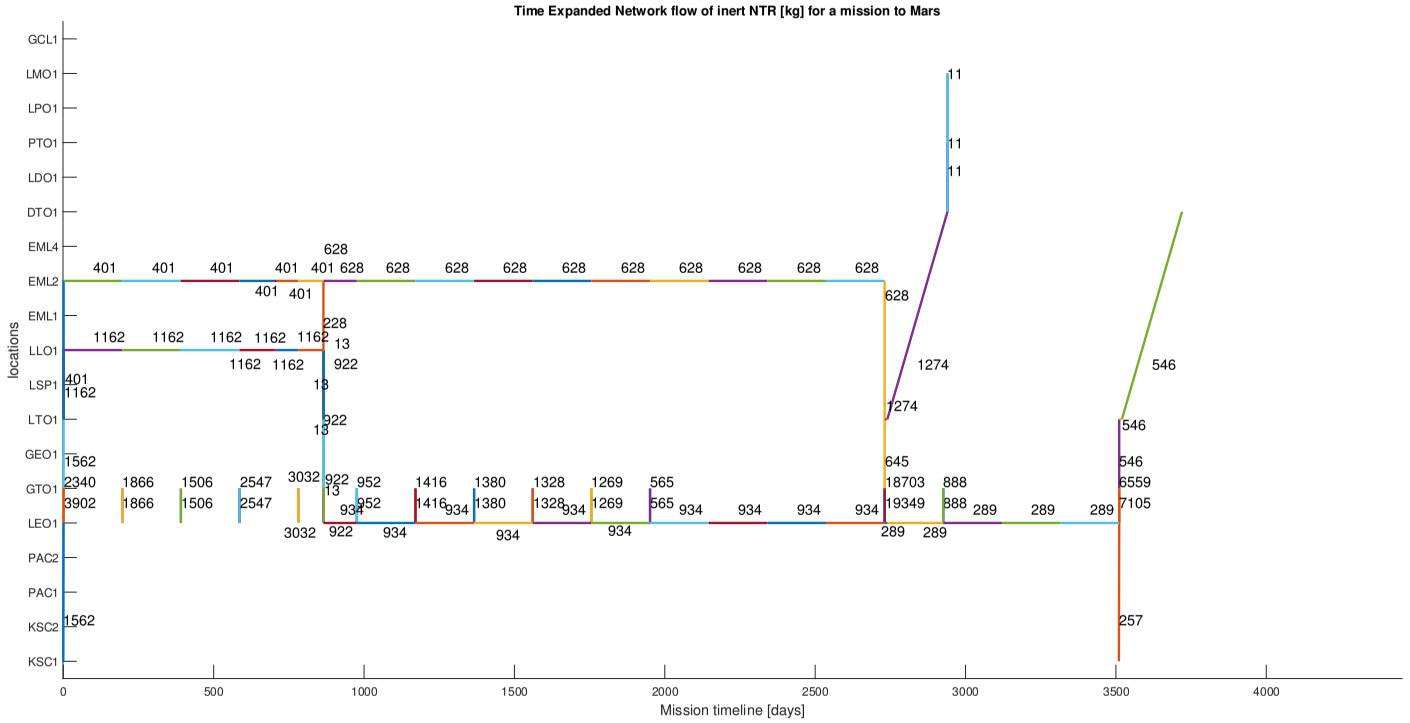



Figure 2-27: This shows how NTR engines move along the network. The different colours indicate different edges, and the numbers next to these indicate the size of NTR in kg. Note that NTR are left to "dissociate" and "recombine". This is not a surprise given the amount of different flow sizes transting the network. Note that this is only for the inert mass of the NTR. Similar plots can be drawn for any other commodity (there are 23 in total).

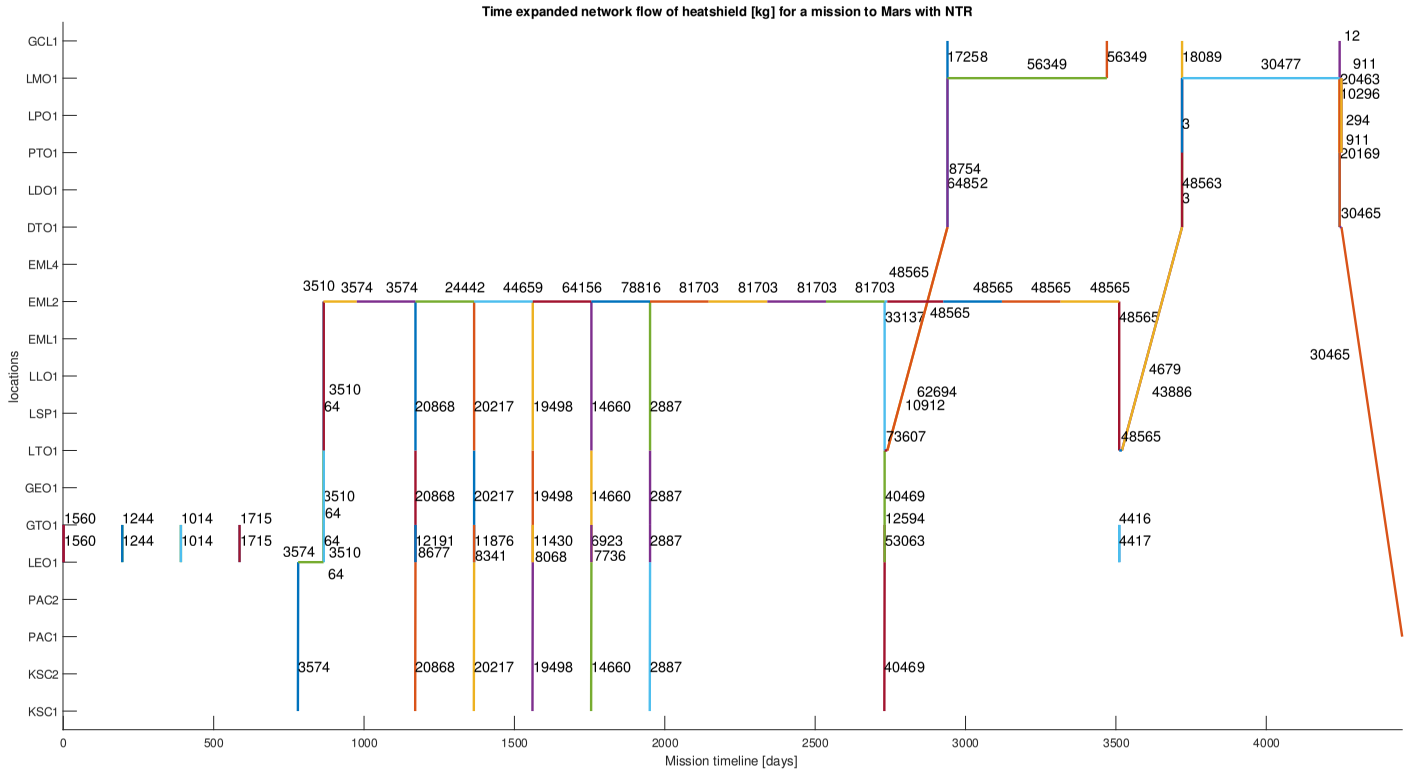



Figure 2-28: This shows the same as the previous page, but for the network flow of the heatshield, as to show that the complexity of the network of transiting hardware does not only affect the NTR, but all commodities.

## 2.5 Summary

The GMCNF formulation is interesting because it allows to optimize not only the network but also look at how the spacecraft are sized around this network. However this produces the results observed previously, giving a transportation network that cannot be used for anything else than gaining insight in the problem. A true mission architecture generation is obtained if one can see how the elements travel through the network.
The following must be done in order to correct the incompatibility between the event-based and flow-based abstractions:

- The GMCNF methods currently do not have any information pertaining to the fact that certain flows are in fact elements or agent, and that some flow are not. In order to be compatible with a software like SpaceNet, the notion of elements traveling within the network must be embedded in the problem.

- Once the GMCNF formulation has the element nature included in it, this information must be process into a form where the path in space and time taken by the various elements is clear and executable (for example, no element spontaneously appears in the network or no element is untraceable)

The executable formulation of the GMCNF problem is described in the next chapter.

# Chapter 3

# Optimization of executable space missions

Now that the background of space logistics and GMCNF has been discussed and a research problem formulated, namely that current GMCNF methods do not produce executable results: the original purpose of this work was to link GMCNF software (**INFINIT**) to SpaceNet, but this was found to be impossible because of the continuous nature of current GMCNF methods. As can be seen in fig.3-1, the GMCNF must have flows that can be associated with elements, or flows that can be traced in order to deduce the trajectory an element that can be fed into an event-centered simulation like SpaceNet.

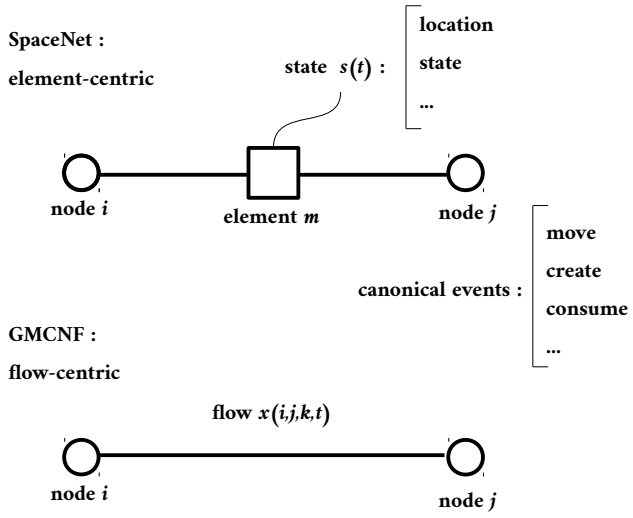



Figure 3-1: Abstract link between event-based simulation (SpaceNet) and flow-based optimization (GMCNF). The event-based simulation is constituted of elements in a logisitics network, possing a time dependant vector of states $s(t)$. The events driving the simulation a comprised by a series of canonical events, such as *move*, *create*,*consume*, etc. For example, a burn is a combination of moving and consuming. The GMCNF optimization is flow-centric, where the flow $x_{ijkt}$ carries a commodity $k$ between nodes $i$ and $j$ at time $t$. Translating the output of the GMCNF into an input for SpaceNet requires that every flow $x_{ijkt}$ be matched to an element in the SpaceNet network: this enforces that the flows in the GMCNF be traceable. Events and states can be obtained by looking at the evolution of the flow or element: a flight from node $i$ to $j$ may for example show a burn. Determining what events happen when is a post-processing task once the flows have been allocated to elements.

This chapter presents a solution to this problem. The general idea is the following:

1. Restrict hardware parts to a predefined set of sizes instead of a continuous linear span. An example of this is rocket engine sizing: by restricting the problem to a posses only one type of rocket engine or a multiple of this (so 2,3,4,etc. engines are also permitted).

2. Devise a tracking algorithm that allows to link the integer flows into a trajectory that can be followed by a element.

The problem with this is that the formulation is changed to a MILP, which is far more difficult to solve than an LP problem. The following challenges must be addressed:

- Modify the GMCNF formulation in order to allow for components with predefined

sizes

- Examine the computational costs of the new problem formulation and determine the limits in terms of computational capability of this formulation
- Outline modeling solutions allowing the new GMCNF formulation to be used on Mars and NEO mission planning problems

This chapter is articulated into two parts:
In the first, the MILP problem and its challenges are treated. This produces a network containing flows that can be tracked, but that are still in network flow form, i.e. there is no information of how individual elements travel through the network
In the second part treats the post-processing of the discretized network by introducing an algorithm that produces the of histories for each element traveling through the network base on the GMCNF solution. This step soles the tracking problem and allows the GMCNF results to be interpreted by other types of analyses and software, including SpaceNet.

## 3.1 Development of an executable GMCNF problem

### 3.1.1 Semi-continuous GMCNF formulation

One of the main issues when using linear models is that they tend to not reflect reality once one is too far away from the point around which one linearizes. As can be seen in figures 2-24 to 2-28, the optimizer will not refrain from using small and micro-sized flows to resize and redirect flows to obtain an optimal solution. This however is not physically possible for certain commodities, like NTR engines for example: in order to work, NTRs must have at least a critical mass of nuclear material, which cannot have an arbitrary mass.
Luckily, this type of problem is common in linear programming and optimizers are fitted to use so-called *semi-continuous constraints*, where the possible values for a variable $x$ vary in a range from a lower bound $x_{min} > 0$ to an uper bound $x_{max} > x_{min}$, and 0. A semi-continuous variable thus does not have a convex space, and the problem becomes a mixed-integer linear problem (MILP).
Apart from changing the variable type and adding the lower bound $x_{min}$ into the optimizer, this formulation is identical to the dynamic formulation. The advantages of using semi-continuous variables are the following:

- It has far better modeling capability than a continuous variable when applied to commodities that have minimal sizes. Examples include any form of hardware and infrastructure, as miniaturization and physical limits (such as critical mass in the case of NTRs) force a limit on the minimal size of the hardware and thus the flow
- It is cheap in terms of computation compared to integer programming, because the constraint is only binary: either the flow is linear, or it is zero, whereas in integer programming every integer within the range represents a different option to explore.
- It allows for the sizing of the element modeled semi-continuously: in the integer programming case, the sizing space is discrete, and the basic size of the elements is dictated by the user, choice that could lead to largely suboptimal results.

However, this idea suffers from the major flaw that is produces an optimal network that is still not executable: when only a minimum size flow is selected but the range remains continuous, tracking elements through the network remains a difficult task that is subject to a second optimization process. Note that this technique can be used simultaneously with the ones described below, so choosing to model a given commodity using semi-continuous variables does not prevent the use of the method described next.

### 3.1.2 Integer GMCNF formulation: predefined hardware sizes

As mentioned previously, the main flaw of GMCNF is that it is extremely difficult to track flows through the network in order to identify them with elements. A simple way to solve this is to force the flow to be the multiple of a reference value, and then the element can be associated with the reference value and thus by dividing the flow by this number we can track how many elements a flowing from one point to the next, label each element and thus track them as they flow through the network. This brings up the problem of how to modify the constraints of the GMCNF problem. The process boils down to a change of variables: If the reference mass of the element, or unit mass is $\tilde{m}$ we can write

$$\tilde{x}_i = \tilde{m}_i x_i \tag{3.1}$$

How does this change of variables affect the constraints of the problem? In this example, we look at how it affects a set of two linear constraints and two decision variables $x_1$ and

$x_2$:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

with

$$x_1 \in \begin{bmatrix} \tilde{m}_1, & 2\tilde{m}_1, & \cdots & M\tilde{m}_1 \end{bmatrix}$$

if we have

$$\tilde{x}_1 \in \begin{bmatrix} 1, & 2, & \cdots & M \end{bmatrix}$$

then the previous system becomes:

$$\begin{pmatrix} \tilde{a}_{11} & a_{12} \\ \tilde{a}_{21} & a_{22} \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

With $\tilde{a}_{i1} = a_{i1}\tilde{m}_1$. Because of its linear nature, the superposition principle applies to the constraint system: only the elements in the matrix that are effected by the change of variables are modified. This has the advantageous characteristic that it can be applied to any variable simultaneously, for example if we constrain $x_2$ to be a multiple of a different unit mass $\tilde{m}_2$:

$$\begin{pmatrix} \tilde{a}_{11} & \tilde{a}_{12} \\ \tilde{a}_{21} & \tilde{a}_{22} \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

with

$$x_1 \in \begin{bmatrix} \tilde{m}_1, & 2\tilde{m}_1, & \cdots & M\tilde{m}_1 \end{bmatrix}$$

$$x_2 \in \begin{bmatrix} \tilde{m}_2, & 2\tilde{m}_2, & \cdots & N\tilde{m}_2 \end{bmatrix}$$

With $\tilde{a}_{ij} = a_{ij}\tilde{m}_j$.

This can be generalized to any number of commodities and unit masses:

$$\begin{pmatrix} a_{11}\tilde{m}_1 & a_{11}\tilde{m}_2 & \cdots & a_{1N}\tilde{m}_N \\ a_{21}\tilde{m}_1 & a_{11}\tilde{m}_2 & & a_{2N}\tilde{m}_N \\ \vdots & & \ddots & \vdots \\ a_{M1}\tilde{m}_1 & a_{M1}\tilde{m}_2 & \cdots & a_{MN}\tilde{m}_N \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix}$$

Note that this allows any variable to be scaled according to a $\tilde{m}_i$ independently of others, and variables can be chosen to be continuous (classic linear constraints), semi-continuous or integer.

In terms of modeling, choosing integer variables will constraint the problem more than linear type constraints, but this is desirable: for example, in the linear case a propellant depot can resize itself throughout the mission, so it is always optimal for the propellant it carries at a given time. When integer constraints are added, this propellant depot will either be used and possibly run at reduced capacity, or not used at all.

The integer GMCNF formulation is the following:

The flow variable vector $\mathbf{x}$ becomes $\tilde{\mathbf{x}} = \tilde{m}^{unit} \odot \mathbf{x}$, where $\tilde{m} = (m_1\ m_2\ m_3\ \ldots\ m_N)$ is the vector of unit masses of the different commodities (in this case, the first commodity has a unit mass of $m_1$, the second commodity a mass of $m_2$ etc and $\odot$ is the element-wise multiplication symbol. For matrix multiplication, we have $\tilde{M}_{unit}$ which is a empty matrix with the $\tilde{m}_{unit}$ terms on its main diagonal.

We can rewrite the GMCNF problem as:

$$\begin{aligned} \text{Minimize} \quad & \mathcal{J} = \tilde{\mathbf{c}}_i \tilde{\mathbf{x}}_i \\ \text{Subject to} \quad & \tilde{\mathbf{A}}_{ij}^{ineq} \tilde{\mathbf{x}}_i \leq \tilde{\mathbf{b}}_j^{ineq} \\ & \tilde{\mathbf{A}}_{ij}^{eq} \tilde{\mathbf{x}}_i = \tilde{\mathbf{b}}_j^{eq} \\ & \tilde{u}_i^{lower} \leq \tilde{x}_i \leq \tilde{u}_i^{upper} \end{aligned}$$

Where

$$\begin{aligned} \tilde{\mathbf{c}}_i &= \tilde{m}_i^{unit} \odot \mathbf{c}_i \\ \tilde{\mathbf{A}}_{ij}^{ineq} &= \tilde{M}_{ij}^{unit} \mathbf{A}_{ij}^{ineq} \\ \tilde{\mathbf{b}}_j^{ineq} &= \mathbf{b}_j^{ineq} \\ \tilde{\mathbf{A}}_{ij} &= \tilde{M}_{ij}^{unit} \mathbf{A}_{ij} \\ \tilde{\mathbf{b}}_j^{eq} &= \mathbf{b}_j^{eq} \\ \tilde{u}_i^{lower} &= u_i^{lower} \odot \tfrac{1}{m_i^{unit}} \\ \tilde{u}_i^{upper} &= u_i^{upper} \odot \tfrac{1}{m_i^{unit}} \end{aligned}$$

With $\tilde{\mathbf{x}}_k \in \mathbb{Z}, k \in \mathcal{H}$, $\mathcal{H}$ representing the set of variables that are constrained to by multiples of the unit mass $m_{unit}$. The variables $x_l, l \in \mathcal{M}$ remain linear, where $\mathcal{M} \cup \mathcal{H}$ make the entire set of variables. In the next section, the integer GMCNF formulation is applied to the Example Problem of section.2.3.

### 3.1.3 Applying the integer GMCNF to the Reference problem

Applying a new formulation of a problem is not necessarily straight-forward:
In order to verify the computational feasibility and modeling accuracy the GMCNF with predefined harware sizes is tested on the Reference problem discussed in section 2.3: In this case, only the outgoing hardware is constrained to integers. This effectively makes a third of the variables integers. the results are compared to those of the continuous GMCNF with $Isp = 450$ seconds and $\xi = 0$, and $\eta = 0.1$.
The study of figures 3-2,3-3, 3-4, 3-5 and 3-6 lead to the following conclusions:

- In static networks, the network topology does not change, however IMLEO increases substancially for larger $m_{unit}$, see fig.3-5.
- The computational time increase is manageable if it scales properly to larger problems,

see fig.3-6.

It should also be noted that CPLEX indicates that many solutions have numerical problems. However, the solutions were checked to be similar to their continuous counterparts, yielding a certain level of confidence. In addition to that, solutions were identical between two different optimization softwares (CPLEX and Gurobi).

When compared with CPLEX, Gurobi was found to show an optimization time on average half of that of CPLEX, as shown in fig.3-7. Thus, Gurobi is used for the optimizations throughout the rest of this work.

Note that the network obtained in fig.3-2 through fig.3-4 shows a time-paradoxe: because of $\eta > 0$, outgoing hardware is used to ship propellant around the LEO-Moon system. Because this is a loop, there is more outgoing hardware in that loop and in the LEO-Mars system than is actually sent from earth: this is a good illustration of a time paradox, because resources (outgoing harware) is used and recycled without being initially sent from earth.

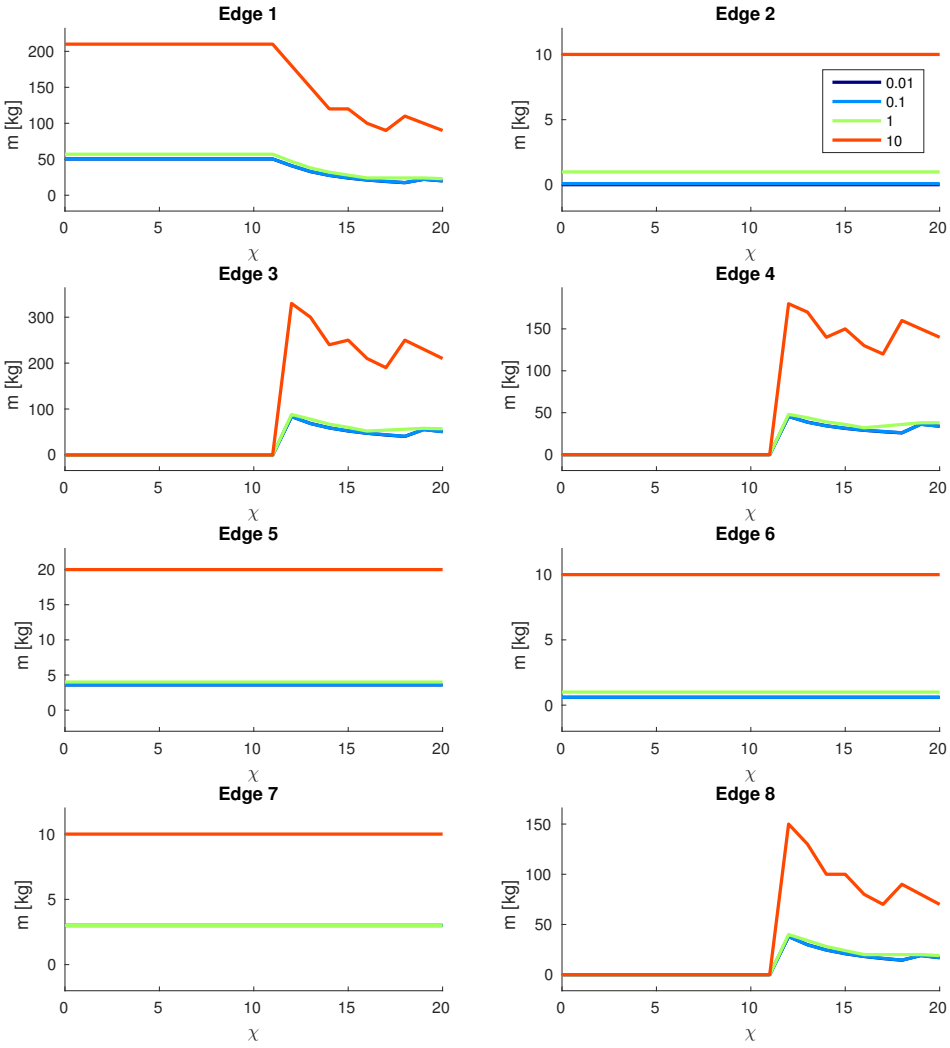



Figure 3-2: Flows of outgoing hardware (commodity 1) for the mixed integer reference problem: edges are the same as in fig.2-12. Notice that the network topology switch apparently does not depend on $m_{unit}$. Notice how for large $m_{unit}$ the curve becomes less smooth: this is due to the optimizer having to switch between integers (or discrete numbers of rocket elements) that each are not optimal (because of the inequality constraint the stages are not optimally loaded).

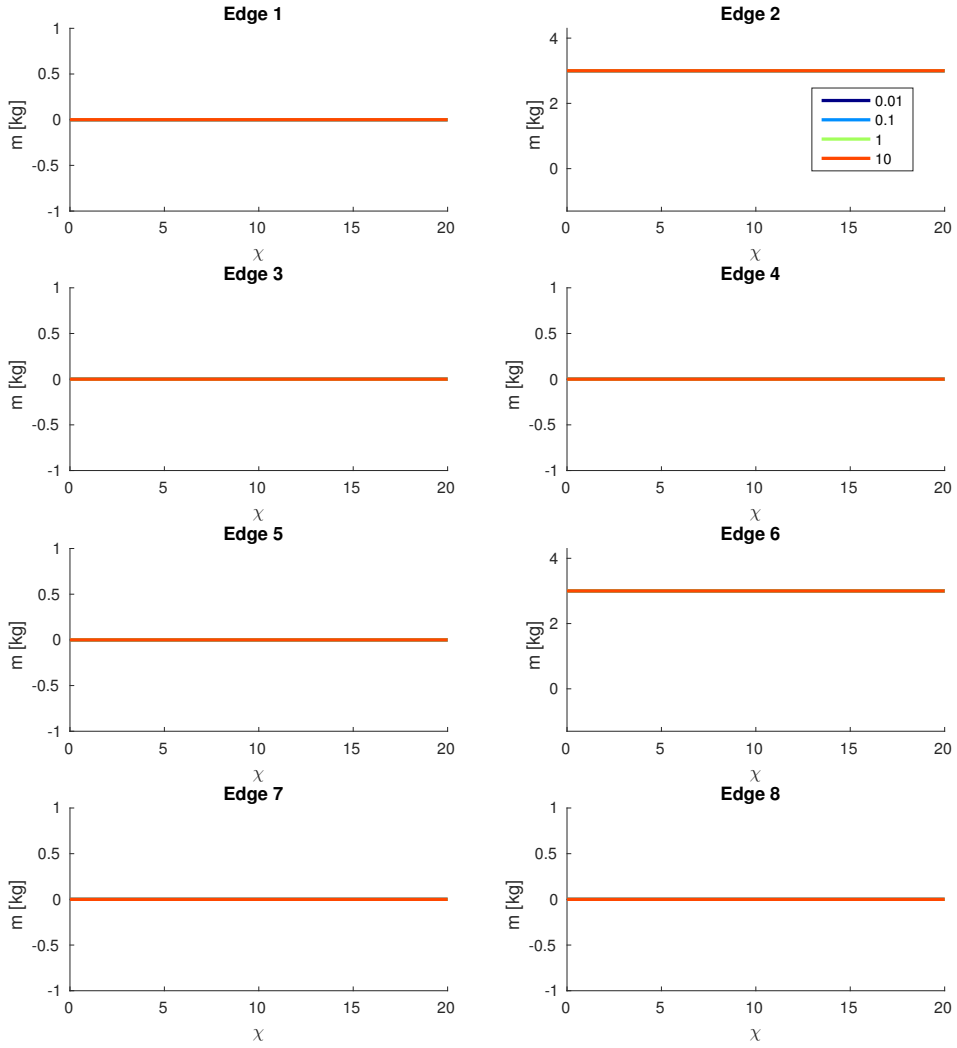



Figure 3-3: Flows of returning hardware (commodity 2) for the mixed integer reference problem: the results do not change (all curves are superposed) no matter $m_{unit}$ or the IRSU rate $\chi$. This happens because this particular commodity is not affected by the others (e.g. it does not limited in any way by the quantity of others) but it drives the problem. Another example of this is the "astronaut" commodity in the previous GMCNF studies [33],[37]. Note that in these works, this commodity flows through the optimal network with no separation or recombination, like other commodities such as rocket stages do.

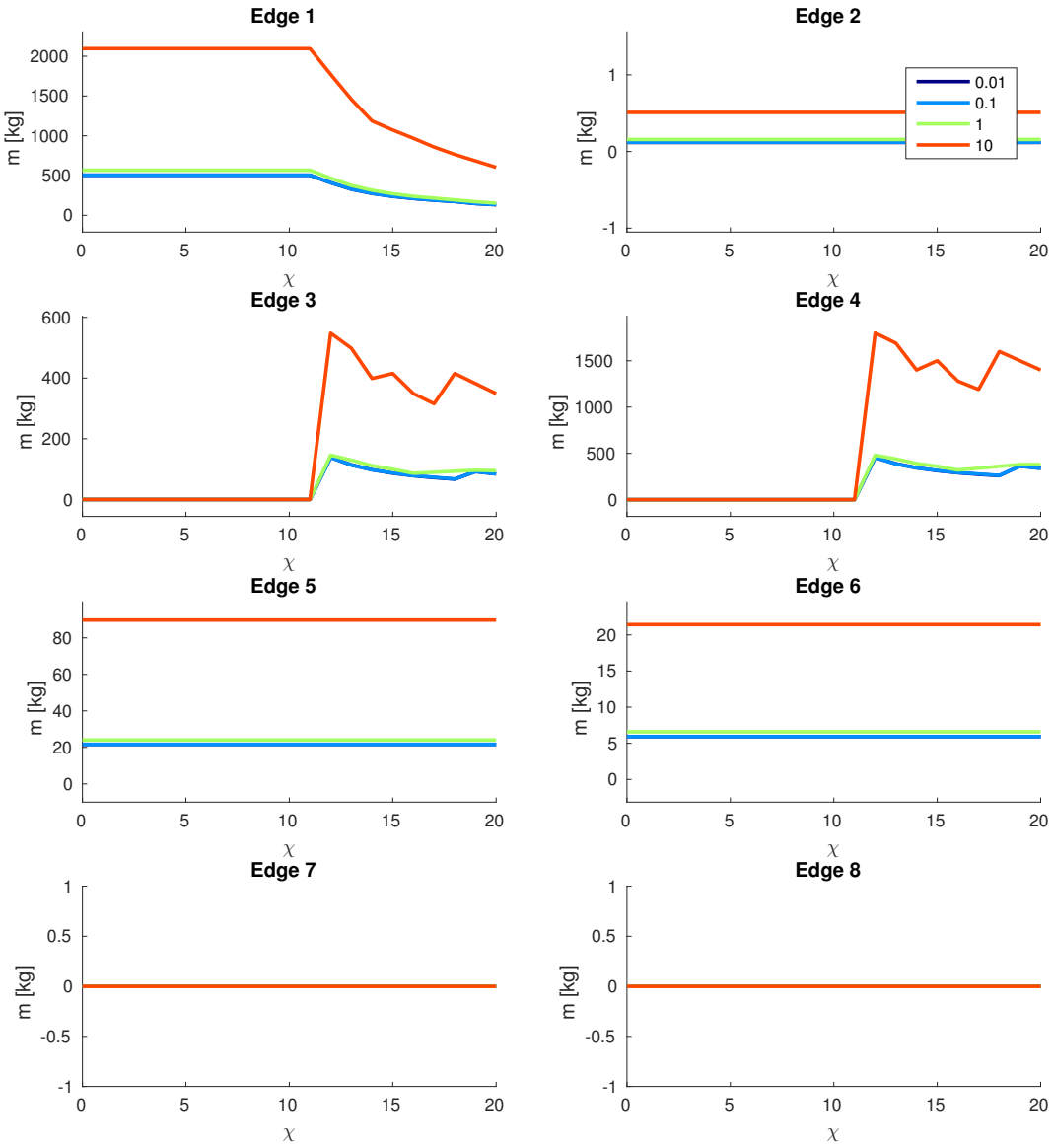



Figure 3-4: Flows of outgoing hardware (commodity 3) for the mixed integer reference problem: The comments made for figure 3-2 can also be made here. However, this commodity is not constrained to be a multiple of $m_{unit}$. The peaks observed for $m_{unit} = 10kg$ follow those of the outgoing hardware, simply because any excess outgoing hardware requires excess propellant to fly it. This illustrates the real-life problem of having to size a piece of hardware for an entire mission and not simply around a specific operating point.

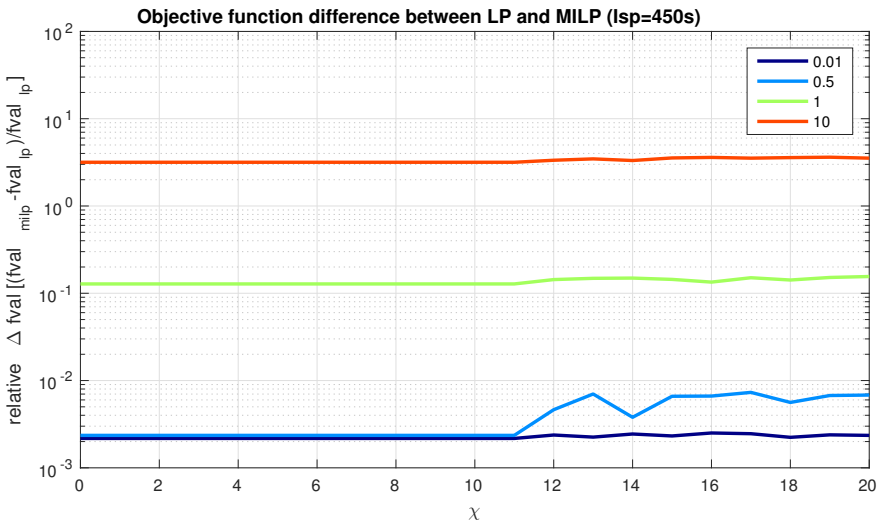



Figure 3-5: This plot illustrates the relative difference between the objective function value at optimum for the MILP problem and the LP problem: notice the log-plot: the IMLEO increases drastically when $m_{unit}$ becomes large, illustrating the sharp mass increase due to the rocket equation applied to excess mass due to a rocket stage that cannot resize itself for a specific mission leg, like it does in the continuous case. Obviously, when $m_{unit}$ becomes small, the problem goes towards becoming a continuous one, hence the small differences for small $m_{unit}$ are expected.

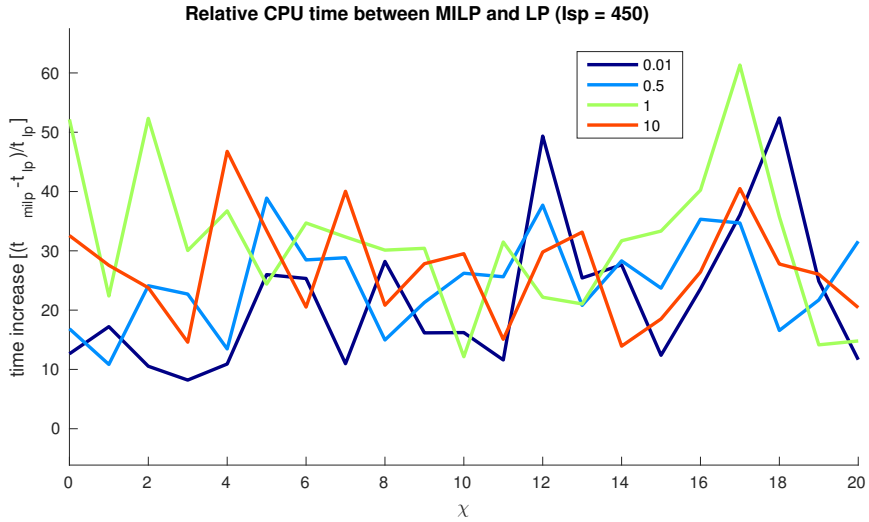



Figure 3-6: The amount of additional time required to solve a MILP instead of an LP in the GMCNF formulation is critical, as the resolution of larger networks with more fidelity might become prohibitive. In this case, a third of the variables have been constrained to take integer values, and the mean time to solve the new problem is 30 times larger. If this applies to the networks used in [33], the resolution time becomes close to 2 hours and 30 minutes, as opposed to 5 minutes. Note that these computation times remain practicle.

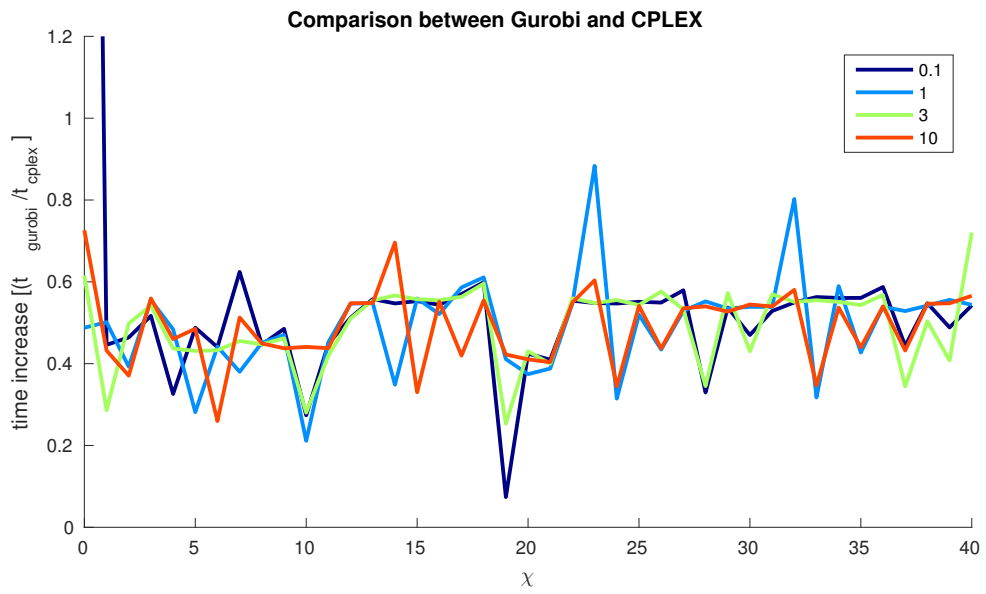



Figure 3-7: Comparison of the CPU time between CPLEX and Gurobi to solve identical optimizations problem: on average, Gurobi is twice as fast as CPLEX, making it the optimizer used for the rest of this work.

### 3.1.4 Complexity and solver requirements for MILP problems

This section is based on [64]. Many problems in operational research are solved using mixed integer linear programming (MILP) as opposed to linear programming (LP). Although the transition from LP to MILP might seem an easy one, it is in fact a major challenge:

Unlike LP problems that can be solved in polynomial time (using the ellipsoid algorithm for example [51]), MILP problems are NP-complete, and an polynomial time algorithm is unlikely to ever be found. Thus it is paramount to manage the size of the problem in order to ensure that a solution can be found in a reasonable time.

The algorithm used in Gurobi [3] and CPLEX [2] for MILP problems is branch and bound, which are heuristics based. Applying MILP to the control of hybrid systems, [64] make an empirical study of how an MILP complexity scales with the size of the problem. A MILP problem can be seen as a confluence of an integer problem with $d$ variables and a linear problem of size $m$, where the integer problem is measured as having a complexity ranging between $O(d)$ and $O(d^2)$, and the linear problems complexity ranges between $O(m)$ and $O(m^2)$. This leads to a computational complexity ranging between $O(m^2 d)$ and $O(m^3 d^2)$ for MILP problems.

These findings have underlined the fact that the number of integer variables must be kept

at a minimum, because the complexity of the problem grows as $d^2$. Note however that shrinking the linear problem size is more effective, as the complexity scales with $m^3$.
The next section discusses how the size of the problem can be reduced. From this point on, only the full Earth-Moon-Mars problem (shown in fig.4-1) is considered.

#### 3.1.4.1 Assessing computational demands of dynamic MILP GMCNF

[37] and [33] both make estimates on the sizes of the problems they solve in order to evaluate the computational demands. The same exercise is done here for dynamic MILP GMCNF problems.
Because a static GMCNF network is relatively small and required little computational power to be solved, there was no effort necessary in order to shrink the problem to more manageable sizes. However this is not true for time-expanded networks, because the size of the network grows exponentially with the number of time-steps. Hence techniques must be used to deal with this issue. They can be divided into two main categories:

**Shrinking the network** The first shrinking of a dynamic GMCNF problem may occur when heuristics described in [33] are used to reduce the size of the time-expanded network by node/arc aggregation, node/arc restriction or cluster-based heuristics. This is extensively discussed in [33], however it is useful to point out that this process (in this case the bis-scale time expanded network) reduces the number of variables from roughly $11'000'000$ to $67'956$ and the number of constraints from roughly $17'000'000$ to $101'868$.

Note that this operation occurs before optimization, and is likely to produce a sub-optimal solution. The bounds on the optimality gap are discussed in [33].

A second means to reduce network complexity is to use the LP solution to produce an optimal network, and use this network as a basis for building the time-expanded MILP. The assumption made here is that the MILP optimum will use a similar network or even identical network topology than the LP produced. Removing all non-used arcs and nodes might be too restrictive as the MILP might not be able to find a feasible solution, however removing spatial nodes that are not used at all by the time-expanded LP does afford a smaller problem and has little risk of yielding an infeasible solution.

**Reducing the number of commodities** Other steps than can be taken to reduce the

size of the problem is to reduce modeling fidelity: by aggregating multiple commodities: everything else being equal, the the number of variables and constraints reduce by twice the amount of commodities removed from the problem, if no new constraints are added to the problem. This operation occurs also before any optimization and will produce a sub-optimal, lower fidelity result. Because this is the realm of modeling, it is difficult to deduce how this might affect the optimality gap and is not discussed in previous work.

Just like the shrinking of the network, it is possible to reduce the number of commodities after the LP optimization. Because commodities interact and thus eliminating a commodity from an arc requires changing the transformation and concurrency matrices **B** and **C** to allow for this operation.

Another way is to aggregate and remodel certain commodities based on the results of the dynamic GMCNF LP problem.

Obviously, in order to keep computational complexity as low as possible, *the number of commodities treated as multiples of a minimum mass (integer programming) must be kept at a minimum.* The choice of variables to constrain to integers can be informed by the results of the LP problem.

Although reducing the number of commodities was done during this research, the results presented subsequently use the full range of commodities.

The two latter methods for shrinking the problem are were applied to the dynamic GMCNF during the development phase, but only the reduced network was implemented the results presented subsequently (i.e. all commodities are kept).

#### 3.1.4.2 Convergence requirements of an integer GMCNF problem

Despite the efforts to reduce the size of the problem in order to obtain reasonable computation times (less than a 24 hours on a high-performance desktop computer with 16 intel Zeon cores), the optimizer does not converge for problems with integer commodities that are extensively used (like LOX rocket engines).

However, the true problem in this case is not CPU time: if the problem is correctly formulated, it can be brought to a more powerful computer to solve in a longer time, as there is no need for a rapid solution like there would be in a dynamic setting such as optimization for flight vehicle control or advertisement placing on a web page. The real concern is

convergence, and if convergence cannot be achieved in a reasonable time, then a feasible solution, albeit suboptimal, must be found. It turns out that feasible solutions close to the LP optimum can be found in a short time, as shown in fig.3-8

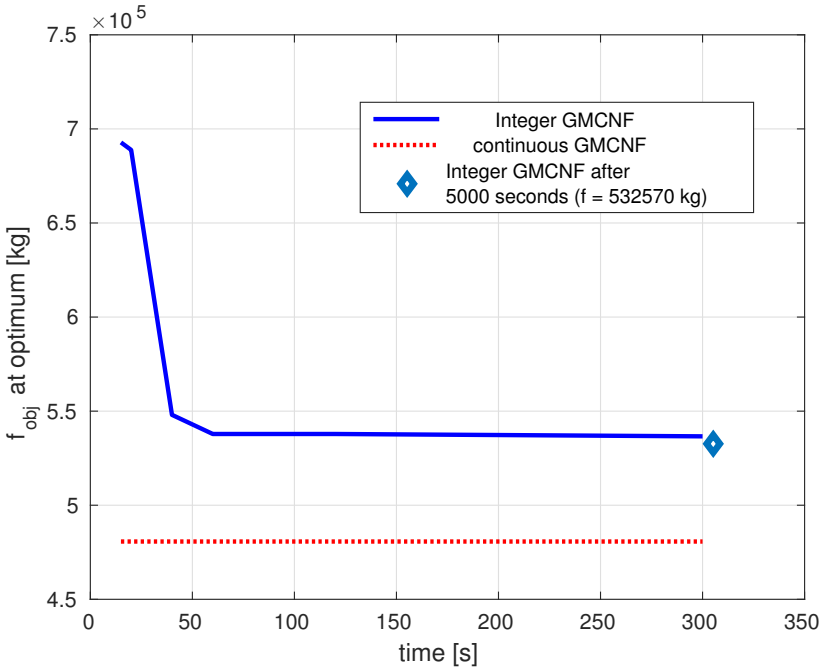



Figure 3-8: Convergence of the objective function value with CPU time: the problem is the full problem discussed in chapter 4. The computer is a high performance desktop computer with 16 intel Zeon cores. The nature of the variables (integer/non integer) can be seen in table 4.1

Fig.3-8 is an important finding: by accepting a solution close to the global optimum and not at the global optimum, fully discretized GMCNF problems become accessible. Another major advantage of this is that more commodities may be added into the problem formulation while keeping it within computational reach, which open the capability approximating non-linear sizing constraints by piece-wise linear constraints.

## 3.2 Tracking elements through a GMCNF network

In order to create a mission sequences from a network flow problem, the flows have to be assigned to elements traveling through the network. This process is called tracking of the flows.

The idea behind the algorithm is to follow a unit of a given commodity from one node to

the next, remove that unit from the network once it has been accounted for, and repeat this until the network is empty. The algorithm is presented below in the form of an example: the time-expanded network flow is shown in fig.3-9, and the table associated with it is tab.3.1.

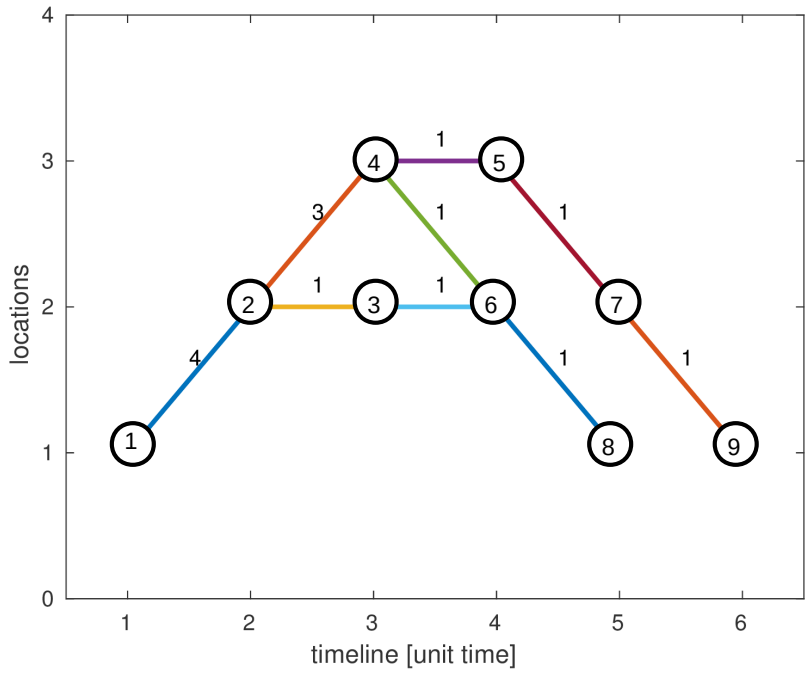



Figure 3-9: Reference problem for the tracking algorithm: the locations and time are generic ones. The goal of this reference problem is to test the tracking algorithm and illustrate its functioning. Here, a commodity of 3 leaves location 1 to split into different parts, then come together again etc. This illustrate typically the phenomena occurring in an integer GMCNF result: elements split, recombine or are abandoned.

| Locations | | Time | | Time | | Flow |
|---|---|---|---|---|---|---|
| Dep. | Arr. | Dep. | Arr. | Dep. | Arr. | |
| 1 | 2 | 1 | 2 | 1 | 2 | 4 |
| 2 | 3 | 2 | 3 | 2 | 4 | 3 |
| 2 | 2 | 2 | 3 | 2 | 3 | 1 |
| 3 | 3 | 3 | 4 | 4 | 5 | 1 |
| 3 | 2 | 3 | 4 | 4 | 6 | 1 |
| 2 | 2 | 3 | 4 | 3 | 6 | 1 |
| 3 | 2 | 4 | 5 | 5 | 7 | 1 |
| 2 | 1 | 4 | 5 | 6 | 8 | 1 |
| 2 | 1 | 5 | 6 | 7 | 9 | 1 |

Table 3.1: Example of a integer GMCNF output. Once this is processed using the tracking algorithm, the results become executable.

The output of the executable GMCNF gives a network like the one shown in fig.3-9, and the output itself is a table similar to tab.3.1. When tracking the flows, the algorithms first step is to pick a starting edge $a$, such that the flow along that edge is non-zero. The search is done along along the $t$ axis.

Then the algorithm searches for all edges $E$ with non-zero commodity flow that are connected to $a$. The algorithm then picks the first edge of $E$, removes a unit of commodity in edge $a$, and repeats the process until the list $E$ becomes empty.

This process is repeated until no more flow within the network can be found. This algorithm is a type of tree-searching algorithm. Several comments should be made:

- The general solution for tracking elements through a network is non-unique.
- The output of the aforementioned algorithm is unique because of the next edge to be chosen is always the first one of the list $E$.
- This algorithm can only guarantee the tracking of all edges if the network is oriented (the time-expanded nature of the network guarantees this).
- The output is a series of matrices that contain on the first line the times of the element track, on the second line the location of the track, and on the third line the node index

of the track.

- If the outputs of two or more element are identical, then the elements can be aggregated: they flow through the network not as $n$ separate entities, but as one entity of size $n$ times the unit mass.

**Algorithm 1** Tracking algorithm

```
1: procedure TRACK
2:     declare data structure Track
3:     StopTotal ← 0
4:     i =← 0
5:     while StopTotal == 0 do
6:         i ← i + 1
7:         StartingPointIndex ← find(Flow ≠ 0)
8:         if StartingPointIndex == ∅ then
9:             StopTotal ← 1
10:        else
11:            Select startingPointIndex as first of the list StartingPoint index
12:            StopLocal ← 0
13:            declare data structure LocalTrack
14:            Extract CurrentLocation, CurrentTime and CurrentNode
15:            Append CurrentLocation, CurrentTime, CurrentNode to localTrack
16:            while  doStopLocal == 0
17:                Extract CurrentLocation, CurrentTime and CurrentNode
18:                Append CurrentLoc, CurrentTim, CurrentNod to localTrack
19:                Extract all non-zero next flows nextFlows
20:                if nextFlows == ∅ then
21:                    StopLocal ← 1
22:                    Remove one unit flow from current edge
23:                else
24:                    Choose first possible flow from nextFlows list
25:                    Remove one unit flow from current edge
26:                    CurrentFlowIndex ← NextFlowIndex
27:            save LocalTrack into Track
```

This algorithm is applied to the executable GMCNF results to produce executable mission architectures issued from a global optimization process: an example of this is shown in the figures 3-10 and 3-11: the first is the output of the integer GMCNF, and the second is the output of the tracking algorithm.

The next step is to use the tracked output in order to evaluate and analyse the mission. An example of this is done in the next chapter. Another application is to interface GMCNF and SpaceNet: a programming architecture of this interface and theoretical work on it can

be found in the appendices of this thesis. However, the coding and testing of this interface is left as future work.

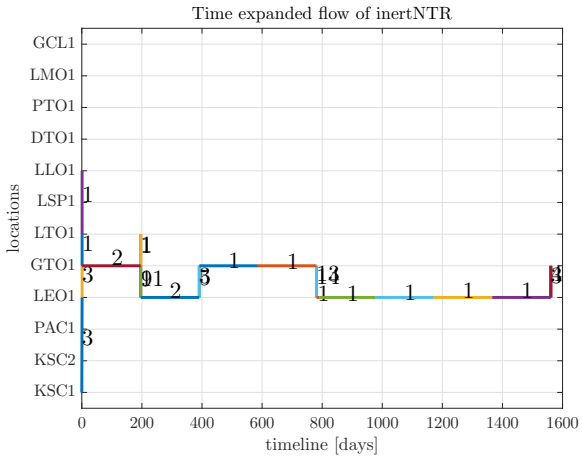



Figure 3-10: Mass flow of the NTRs throughout the network for an executable GMCNF after 5000 seconds of optimisation.

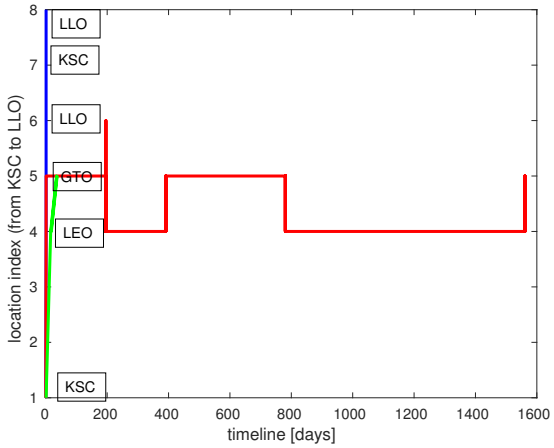



Figure 3-11: Mass flow of the NTRs throughout the network for an executable GMCNF after having been processed by the tracking algorithm: the different colors show how different elements fly through the hardware.

# Chapter 4

# Case study I: design of a manned mission to Mars

The chapter contains both the case-study setup and the discussion of its results. The goal it to obtain an executable mission.

## 4.1 Setup of the case-study

The case-study is designing a manned Mars mission through optimization. The spatial network is shown in fig.4-1. The time line extends over a 7 year time line. This case is presented in [33] using linear programming except for the crew and returning crew commodities. The case study of [33] is used to build the integer GMCNF case, so it is reffered to as "reference problem" or *Dynamic* GMCNF, because it integrates the time dimension but does not produce executable results. A complete list of assumptions about the modeling can be found in [33]. Because this case-study builds on top of that of [33], only the additional assumptions are listed below. These concern the modeling of the different commodities. For each commodity, the following modeling choices are made:

- Continuous, Semi-continuous or integer variable. Hardware commodities such as rockets must be tracked throughout the network, thus are modeled by integers. But continuous commodities such as hydrogen remain continuous in the optimization. There are no semi-continuous variables in the problem.

- The bounds must be chosen. In the continuous case, this not much of an issue because

there is not reason to fix a maximum bound on a commodity, and fixing a large bound does not prejudice the optimization process. This however is not true for the integer case, where the optimizer will explore different options (non-convex space because of the integer nature) and the number of options depend on how many integers are in the range. Thus the range must be chosen small for optimization speed, but large enough to make the problem solvable. The process by which this is made is by running the continuous problem, solving it, and incorporating the integer variables one by one, and choosing a bound based on the previous results.

which are presented in order.

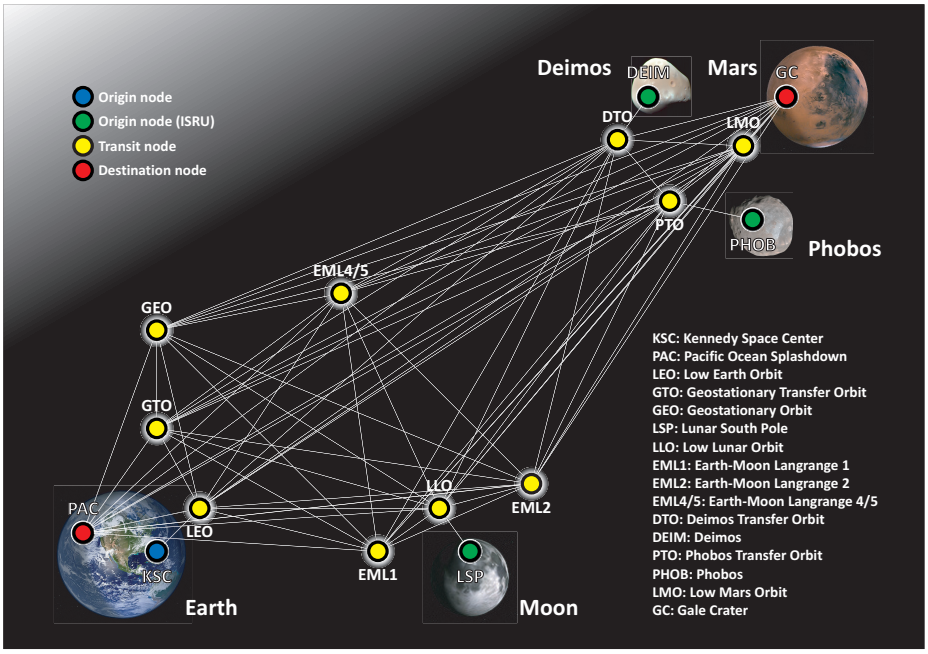



Figure 4-1: Illustration of the space-network used for the case-study.

The modeling of each commodity is discussed below:

**Vehicle** The vehicle is the commodity carrying the astronauts on their mission. Because it is a piece of hardware, it must be tracked through-out the network and thus cannot be left as a linear commodity. However, because crew and returning crew commodities are modeled as integers, even in [33], the vehicle sizes are not expected to change in the integer version. In the reference problem the vehicle takes masses of 37′540, 10′000 and 27′540 kg (as seen in fig.**??**). Thus a unit mass of 10′000 kg is chosen, which might be suboptimal when the vehicle is supposed to weigh 27′540 and 37′540 kg, but give a realistic vehicle size

and enable the vehicle units of 10′000 kg to be tracked throughout the network.

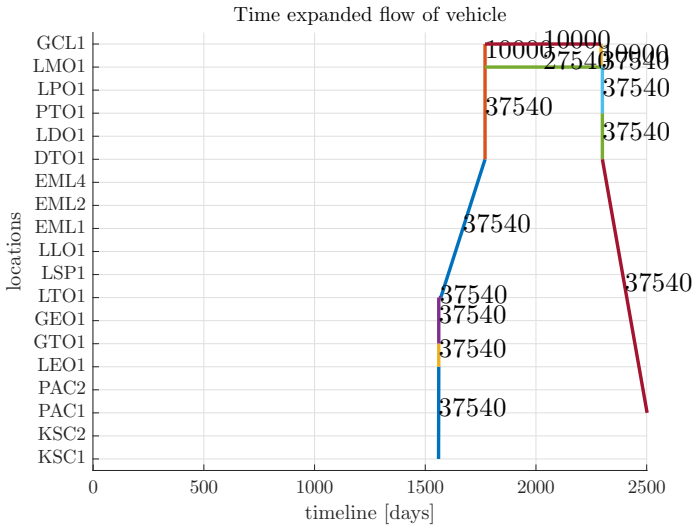



Figure 4-2: Vehicle flow of the Mars mission case study of [33]

**Payload** The payload mass flow in the reference problem is always 51′700 kg, except on Mars where is it momentarily separated into 35836.6 and 15863.4 kg before it is put back together. Because there is no physical reason for this separation to take place (these separations and recombinations of flow are common in linear problems, as microscopic flows are allowed and the objective function might be reduced by a insignificantly small amount) but it does increase the complexity of the mission. Thus it is decided to fix the unit value of the payload to 51′700 kg.

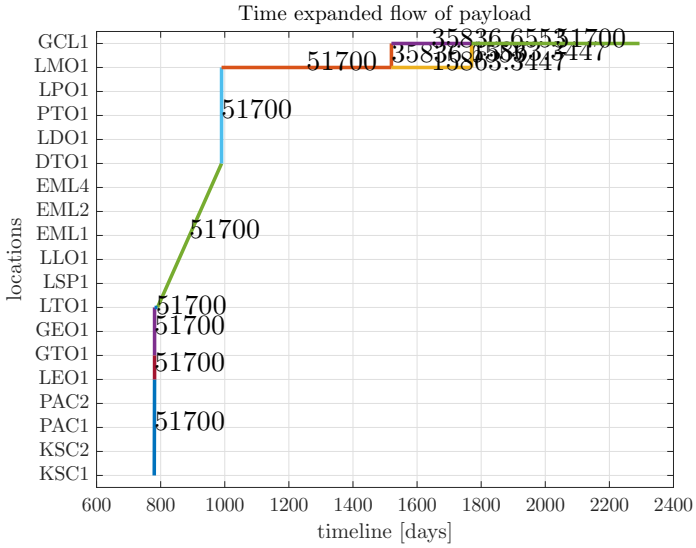



Figure 4-3: Payload flow of the Mars mission case study of [33]

**Crew** the crew must be constrained to be integer values, because a integer number of astronauts is sent out. Note that this is enforced in the original work by [33].

**Returning crew** Returning crew can only be created on the destination point by converting a "crew" element. However, the flow can be separated on the way down, which is observed when the constraint is not enforced. Note that because the crew and returning crew commodities are scarce throughout the network, adding the integer constraint affects the computation time only moderately.

**Sample** Because this commodity never changes its value within the mission, this commodity is modeled linearly.

**Hydrogen** Hydrogen is a liquid commodity that must be contained in tanks (constraint enforced by the GMCNF formulation), thus is is modeled as a linear variable.

**Oxygen** The modeling of oxygen is the same as for hydrogen (continuous variable).

**Methane** The modeling of methane is the same as for hydrogen (continuous variable).

**Water** The modeling of water is the same as for hydrogen (continuous variable).

**Food** The modeling of food is the same as for hydrogen (continuous variable).

**Waste** The modeling of waste is the same as for hydrogen (continuous variable), although for preliminary optimizations it is removed from the problem because its only contribution to the model is to add mass along an edge. In the present formulation it is modeled as being discarded as it is produced, however, it does have a minute effect on the objective function value at optimum. In the results presented in this work, it is included in the model as a continuous variable.

**InertLOX** Inert LOX stands for the empty mass of a oxygen-hydrogen rocket stage without its reservoir. This is a critical commodity because it has to be tracked throughout the network and it represents a piece of hardware. Thus the inert LOX commodity is modeled as an integer variable. Because it is a piece of hardware, the optimization gives good insight on the sizing of this piece of hardware, one of the major capabilities of the GMCNF formulation. The mass of this rocket unit provides an interesting research direction, as it allows the sizing of a rocket that would work across an entire

mission campaign. For this particular case, a unit rocket size is chosen based on existing upper stage and lunar rocket engines, such as the Apollo service modules propulsion system. Choosing a small engine will closely replicate the linear result, however choosing a too small size means that the integer range becomes large, making the optimization more difficult, and also creating potentially unrealistic designs with spaceships having hundreds of micro-rocket engines. A good trade off is to choose a propulsion system of an upper stage such as that of the Centaur upper stage: it uses two Pratt & Whitney RL 10A-3-3A cryogenic multiple start engines and has a total mass of 2.25 tons [16]. Because in this model the "inert rocket" includes not only the engines but also the other systems except the tank (i.e. gimbaled mechanism, pressurizing tanks etc.) the unit rocket engine mass is estimated to be 1000 kg. Choosing a unit engine size leads to a spacecraft looking similar to the concept shown in fig.4-4 where small rockets are strapped together to form a large rocket.

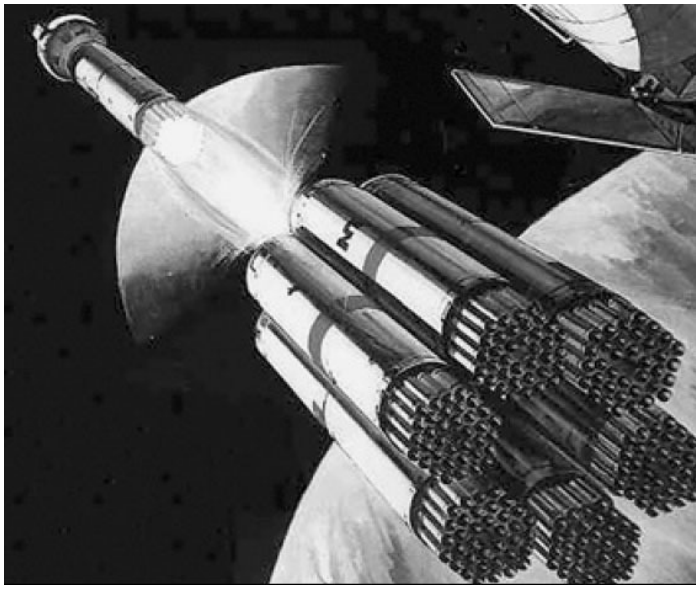


Figure 4-4: Example of a launch concept (OTRAG launcher, [4]) with multiple unit rockets strapped together to form a large rocket.

**InertNTR** inert Nuclear Thermal Rocket is also a piece of hardware, hence it has the same properties as the inert LOX commodity. The inert NTR has some limitations in size: unlike chemical rockets, nuclear rockets need a critical mass of fissil material to function. The smallest NTR the author found, having a mass of 390 [kg]. Note that one can be more restrictive by setting 2 to 3 different NTR rocket types (small,

medium and large), depending on current NTR technology or the results of a linear dynamic GMNCF. [8] indicates that the TRITON nuclear engine is a good choice for Mars transport, and that it is selected for this purpose. The engine itself weighs 821 kg [8]. Assuming that mass has to be added to account for fastening the engine to the spacecraft, including control systems, gimbal mechanisms etc. Thus the unit nuclear rocket engine mass is estimated at 2000 kg.

**InertLCH4** Although the linear continuous version of the problem says that no methane rocket should be used for space transportation (this does not include ascent from a planet like Mars), this commodity has the same properties as the LOX engines. Because it is also a chemical cryogenic engine, the same value of 1000 kg as the LOX engine is used.

**InertLander** Because a lander can have a bread range of sizes, the unit size is based on the results of the reference problem. Although 34 different masses of lander exist in the network, the highest number of occurrences occurs at a mass of 2717 kg, and the maximum value is 20970 kg. Thus the reference mass can be chosen to be 1000 kg in order to be close and above to the maximum occurring value.

**TankLH2** Using the reference problem: the maximum value is 1198 kg and the maximum occurring value is 551.1 kg. In this case, a mass of 200 kg can be chosen in order to give a good range of integers. [58]

**TankLOX** The same selection process as the hydrogen tanks is used for the oxygen tanks: because the maximum size is 2698 kg, a unit size of 400 kg is chosen. Note that the size of these tanks are much smaller than the propellant depots currently considered, such as [58] where empty mass is between 2 MT and 15 MT with a propellant mass fraction of ranging from 0.848 to 0.938.

**TankLCH4** The same selection process as the hydrogen tanks is used for the oxygen tanks: because the maximum size is 1000 kg, a unit size of 400 kg is chosen. The same comment concerning size is made here than for the LOX tank.

**TankH2O** The same selection process as the hydrogen tanks is used for the oxygen tanks: because the maximum size is 736 kg, a unit size of 100 kg is chosen. The same comment concerning size is made here than for the LOX tank.

**Aeroshell** Because the spectrum of masses of aeroshell or heat-shield used throughout the reference problem is so large (from 0 to 92′080 kg) and the integer range is to be kept lower than 30, a unit mass of 5000 kg is chosen.

**Plant O2** The maximum mass throughout the mission is 2365 kg, but there only 4 occurrences of this commodity. In order to give the optimizer room for adjustment, a unit mass of 1000 kg is given. This fits well with previous oxygen ISRU plant sizing calculations, which designed plants weighing from 400 kg to 1593 kg ([55], [56]).

**Plant H2O** The same process for sizing "plant O2" is applied to the water extraction plant, and the unit mass is 2000 kg.

**Plant CH4** The same process for sizing "plant O2" is applied to the water extraction plant, and the unit mass is 850 kg.

The bounds and unit masses of the problem are summarized in the table 4.1

The optimization runs are carried out on a high-performance desktop computer with 16 intel Zeon cores and 32 virtual cores using Gurobi and Matlab2016a.

| Commodity | ID | Unit mass [kg] | Variable type | lower bound | upper bound |
|---|---|---|---|---|---|
| vehicle | 1 | 10000 | int | 0 | 15 |
| payload | 2 | - | cont | 0 | 100'000 |
| crew | 3 | 100 | int | 0 | 6 |
| crewRe | 4 | 100 | int | 0 | 6 |
| sample | 5 | - | cont | 0 | 250 |
| hydrogen | 6 | - | cont | 0 | 100'000 |
| oxygen | 7 | - | cont | 0 | 100'000 |
| methane | 8 | - | cont | 0 | 100'000 |
| water | 9 | - | cont | 0 | 100'000 |
| food | 10 | - | cont | 0 | 100'000 |
| waste | 11 | - | cont | 0 | 100'000 |
| inertLOX | 12 | 1000 | int | 0 | 30 |
| inertNTR | 13 | 2000 | int | 0 | 30 |
| inertLCH4 | 14 | 1000 | int | 0 | 25 |
| inertLander | 15 | 1000 | int | 0 | 25 |
| tankLH2 | 16 | 200 | int | 0 | 20 |
| tankLOX | 17 | 400 | int | 0 | 20 |
| tankLCH4 | 18 | 400 | int | 0 | 25 |
| tankH2O | 19 | 100 | int | 0 | 20 |
| aeroshell | 20 | 5000 | int | 0 | 20 |
| plant o2 | 21 | 1000 | int | 0 | 10 |
| plant h2o | 22 | 2000 | int | 0 | 25 |
| plant methane | 23 | 850 | int | 0 | 15 |

Table 4.1: Summary of the parameters in the executable problem: the table shows the discretization of the different commodities. If the variable is chosen to be continuous, the upper bound is the maximum mass in kg, and if it is chosen to be discrete, the upper bound is the maximum number of intances along one edge.

## 4.2 Results of the case-study

Multiple runs were made in order to check for convergence. The final result was obtained by halting the optimizer after 5000 seconds. The full results of the case-study are shown in appendix **??**. The most relevant of these figures are shown below from fig.4-5 to fig.4-8. The first series of figures show how the commodities flow through the network, while the second series shows the mass distribution throughout the network, mass spectra.

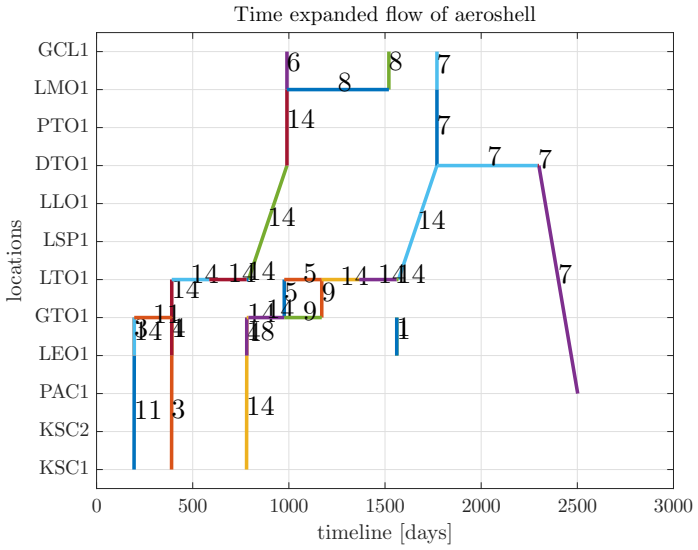



Figure 4-5: Mass flow of the aeroshell throughout the network. Although the general scheme is the same as the continuous problem shown in fig.2-28, there are slight differences in the topology locally, but mostly it becomes clear on how to track elements through the network.

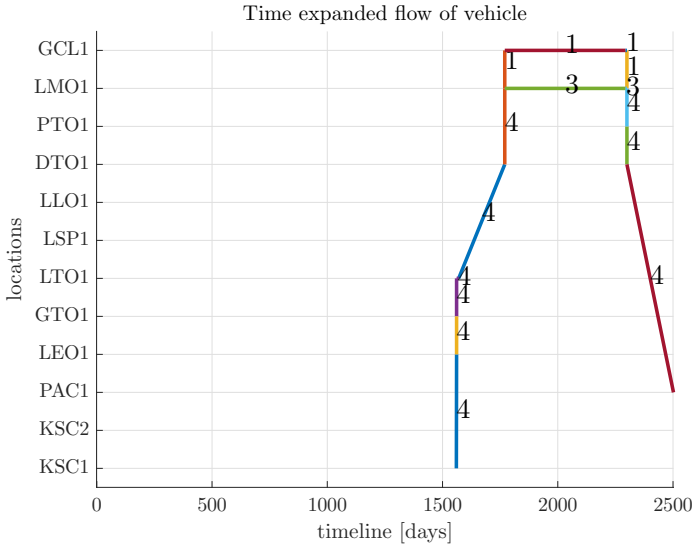



Figure 4-6: Mass flow of the vehicle throughout the network.

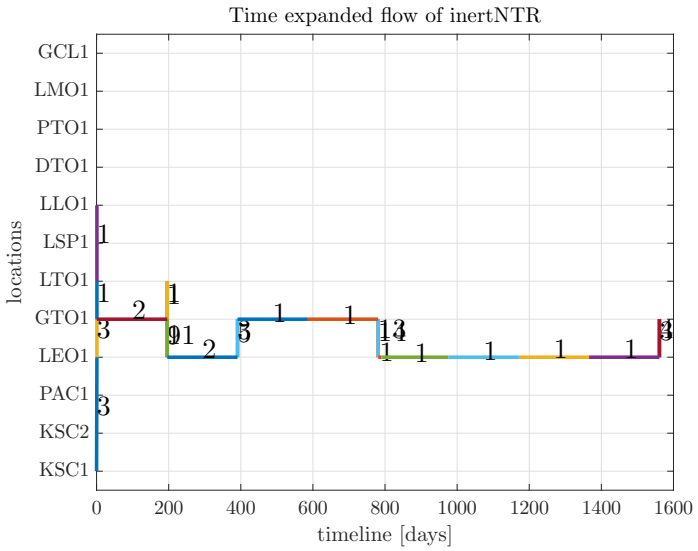



Figure 4-7: Mass flow of the NTRs throughout the network.

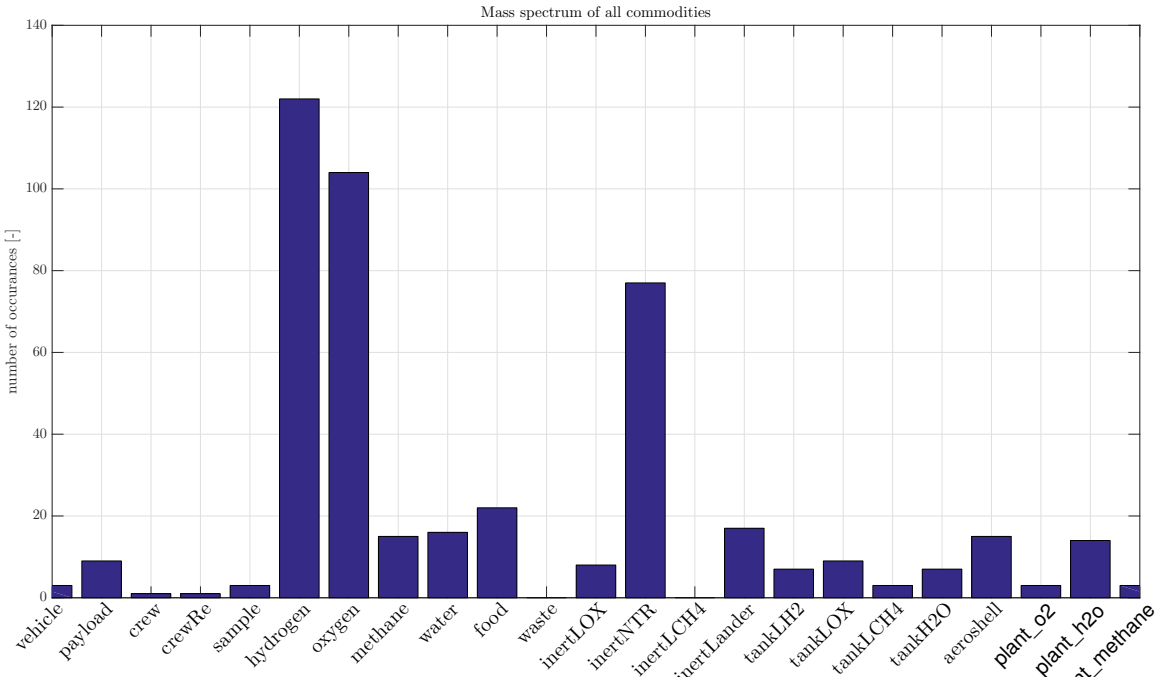



Figure 4-8: Number of different masses throughout the network for all commodities: because of the integer constraints, the number of different instances becomes manageable for all commodities that are discretized. Commodities like hydrogen and oxygen may vary continuously throughout the mission, so their high number of different masses is not an issue as hydrogen and oxygen does not have to be tracked through the network.

## 4.3 Discussion

In this work, 3 metrics of space mission architectures are discussed:

1. Initial mass in low earth orbit (IMLEO), which is essentially a metric of the direct cost of the mission

2. Number of launches: this metric is related to the reliability and feasibility of the mission: a high number of launches implies a higher mission failure probability, in case a launch fails. The feasibility aspect is related to the launch rate: currently, a typical launch rate is 15-25 rockets per year per rocket family. A launch rate that is higher than this will require an availability of rockets that might not be able to be met.

3. Number of manned dockings: again, this number represents risk. Note that undockings are far less difficult to execute, and are not included in this study.

Another important aspect in order to compare the mission architectures is to differentiate between the manned segment of the mission, and the unmanned segment of the mission: in a mission scenario where many operations are executed without any direct risk to humans, a direct comparison between number of dockings and operations is does not reflect the risk of the mission, because automated operations in order to prepare a space infrastructure can be repeated in case of failure. Thus the distinction between manned and automated segments of the mission are made throughout this section. The comparative study is made on four manned missions to Mars:

1. NASA's reference mission architecture DRA5.0

2. The dynamic GMCNF optimized mission to mars from [33]

3. The executable GMCNF optimized mission produced in this study

4. SpaceX's Interplanetary Transport System

These are briefly summarized below:

**DRA5.0** The mission concept of DRA5-0 is shown in fig.4-9. The idea is that martian ISRU is used to accumulate enough propellant to send the crew back to earth. For

this to happen, the mission is split into 2 phases:

The first is an infrastructure set-up phase where hardware is sent ahead of the crew to install the mining base, a start collecting propellant.

The second is the crewed mission leg, which involves less spacecraft, to send the crew, perfom the landing, exploration and come back.

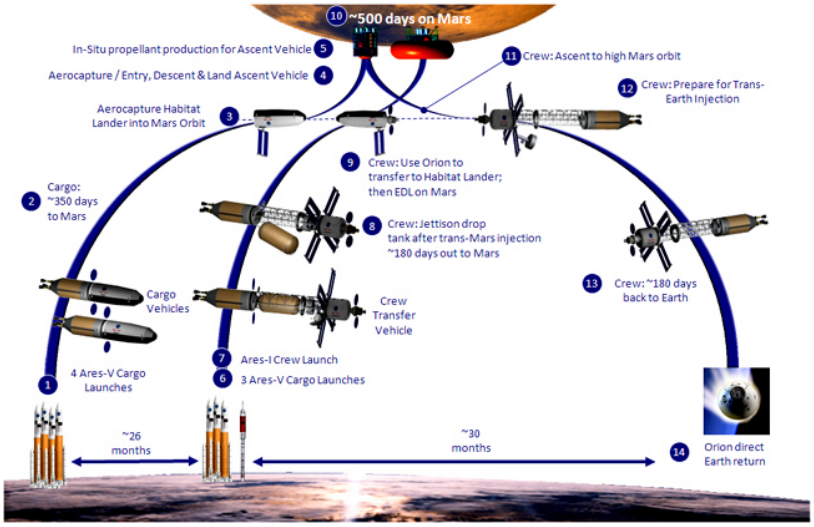



Figure 4-9: Mission concept of DRA5.0. Note the two mission segments: the first installs the space-based infrastructure, and the second sends the humans there and back.

**Dynamic GMCNF** The dynamic GMCNF (fig.4-10) also has two parts to the mission, namely the ISRU deployement and use phase, and the manned leg. Although this mission is infeasible for logistical reasons (non-traceable elements, as previously mentioned), it is included in this comparative study as a lower bound in terms of IMLEO, but also in order to provide a direct comparison with the executable GMCNF.

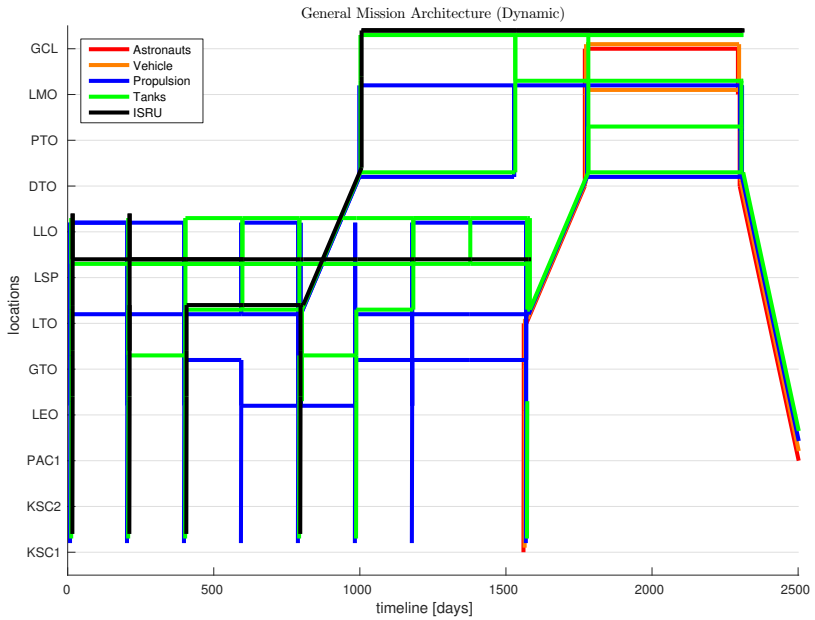



Figure 4-10: Mass flow diagram of the dynamic GMCNF mission architecture: the commodities are grouped into different categories for better visualization. All propulsion elements (LOX rockets, NTRs, LCH4) are grouped, as well as all tank types, vehicles, and astronauts. Note the two segment architecture, where the first segment lasts over a long time in order to accumulate the different propellants, and then the astronauts are sent.

**Executable GMCNF** The executable mission is the first one to be designed by GMCNF method to be logistically feasible. It is obtained with the problem setup described throughout this chapter. Again, it features a two segment architecture, where ISRU is sent ahead to gather material for the flight home. Note that it is similar to the dynamic GMCNF architecture, but it has less flows. This is due to having finite sized elements, that cannot be divided at will like in the dynamic case.

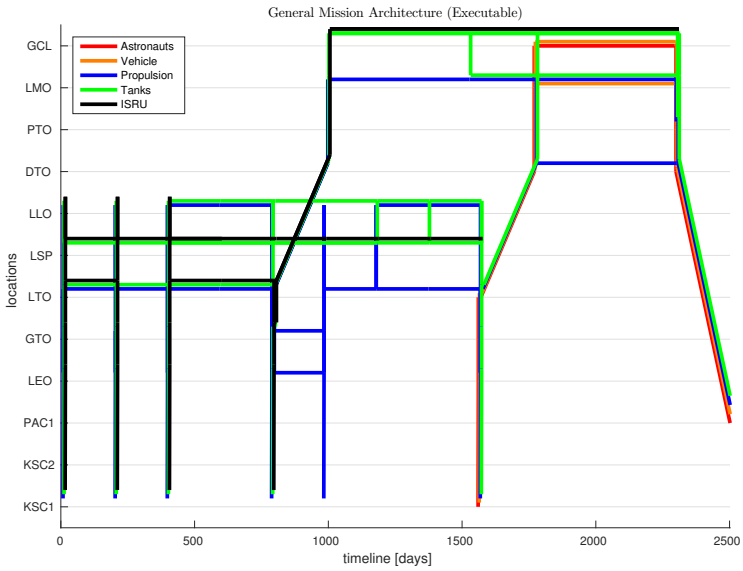



Figure 4-11: Executable mission architecture overview. Note that it has less flows than the dynamic GMCNG architecture.

**Interplanetary Transport System** The Interplanetary Transport System mission architecture [11] is presented here in order to afford comparison on the complexity of the mission. Mass-wise, it should not be compared to the other mission architectures because its purpose is to colonize Mars, and not explore it. Thus, IMLEO is far larger than the other missions. However, this mission architecture is remarkable because of its simplicity and alternative architectural approach to manned space missions.

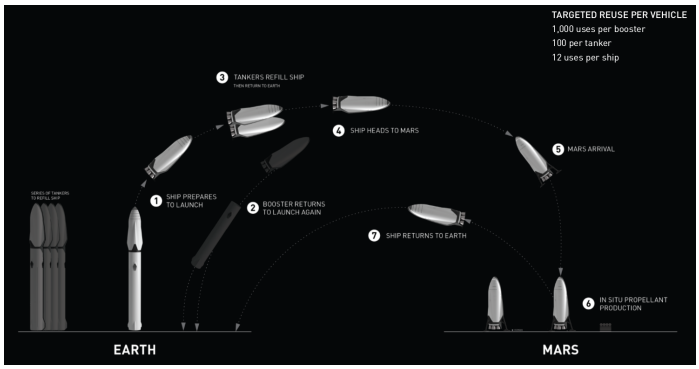



Figure 4-12: ITS mission architecture: note the relative simplicity of the mission. Instead of deploying a large in space infrastructure like the GMCNF cases or a complexe mission such as in DRA5.0, the mission architecture pays less attention to IMLEO and allows for larger spacecraft that do not need intricate in-space infrastructure to function. Image from [11]

### 4.3.1 IMLEO

The results are presented in fig.4-13

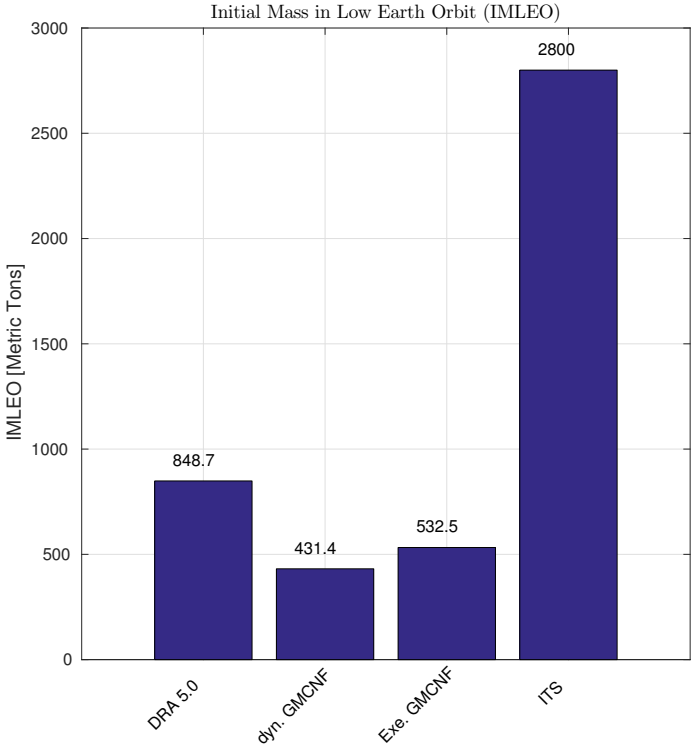



Figure 4-13: IMLEO for the four missions considered in this study: because the ITS mission is has a different purpose (colonize Mars instead of exploring it), it has a much larger launch mass, and should not be compared directly to the other 3 missions. These have different IMLEOs: assuming a launch cost of $10'000$ USD per kilogram, the dynamic GMCNF represents a cost reduction of $4,173$ billion USD (49% reduction in IMLEO), but unfortunately this architecture is not executable. Even though the executable GMCNF mission only has a 37% reduction is IMLEO, it still saves 3.165 billion USD with respect to the reference architecture.

The results show that GMCNF methods produce mission architectures that combine ISRU technologies, aerocapture, and NTRs in an optimal way from the IMLEO perspective. This optimum offers substantial savings with regards to the reference mission DRA 5.0. It should also be noted that executable missions require more IMLEO than non-executable scenarios, because of logistical constraints, for example an element cannot resize itself mid-flight.

Although IMLEO is a serious cost driver for a space mission, it has to be emphasized that it is by no means the only metric upon which a mission architecture should be evaluated. Other metrics, such as number of operations (launches, dockings, etc.) are quintessential,

but these are all concerned with the mission itself. They do not include development cost and time, which are some of the main drivers of a space mission. An example of reduction in development cost is to design and optimize not for a single mission segment, but for the entire campaign, cutting development costs because only a single type of hardware is built to satisfy the entire mission needs. This is discussed in the next chapter.

### 4.3.2 Number of launches

The number of launches is computed by counting the number of edges that have the "Kennedy Space Center" or KSC node as origin node. The results are shown in fig.4-14

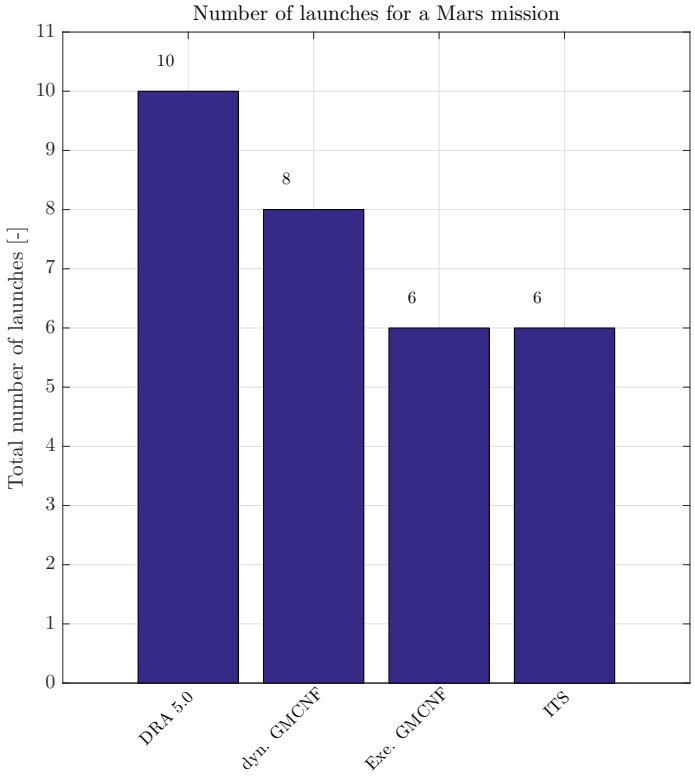



Figure 4-14: Number of launches for a single mission to Mars for all 4 missions: DRA5.0 requires 9 cargo launches, the dynamic GMCNF require 7 cargo launches, the executable one requires 6, and the ITS requires 5 cargo launches. All missions send astronauts once the cargo and/or infrastructure is in place, except for the ITS mission.

The number of launches required for a Mars mission is somewhat unexpected: the GMCNF is expected to have the highest number of launches, because it tends to balance optimally the flows throughout the network flow, leading to small flows to balance need and demand. On the other hand, DRA5.0 is expected to have few launches, because it is

designed with operational constraints in mind.

The following explanation can be made: Because the DRA5.0 is designed in a classic way, a mass to launch was obtained, and then the number of launches is calculated depending on the mass to launch. This sequential approach leads to a higher IMLEO, requesting more launches. The GMCNF on the other hand is optimized for IMLEO, thus fewer rockets are needed.

In conclusion, the GMCNF does not suffer from having too many launches, however future optimizations should contain constraints limiting launch masses to values that are capable of being launched. It also means that future missions will likely use a small number of large launch vehicles rather than a large number of small vehicles.

### 4.3.3 Number of dockings

Counting the number of dockings is not a easy as counting the number of burns or launches. However, they can be counted on the bat chart. These are shown for both dynamic and executable GMCNF in figs.4-15 and 4-16 respectively:

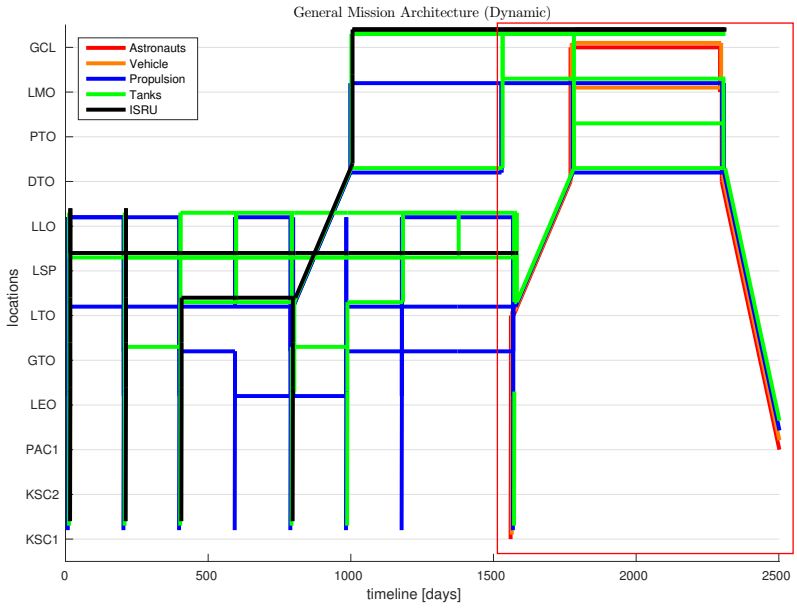



Figure 4-15: Mass flow diagram of the continuous GMCNF. The manned segment is contained in the red rectangle. 7 docking/assembly operations are counted around the manned vehicle.

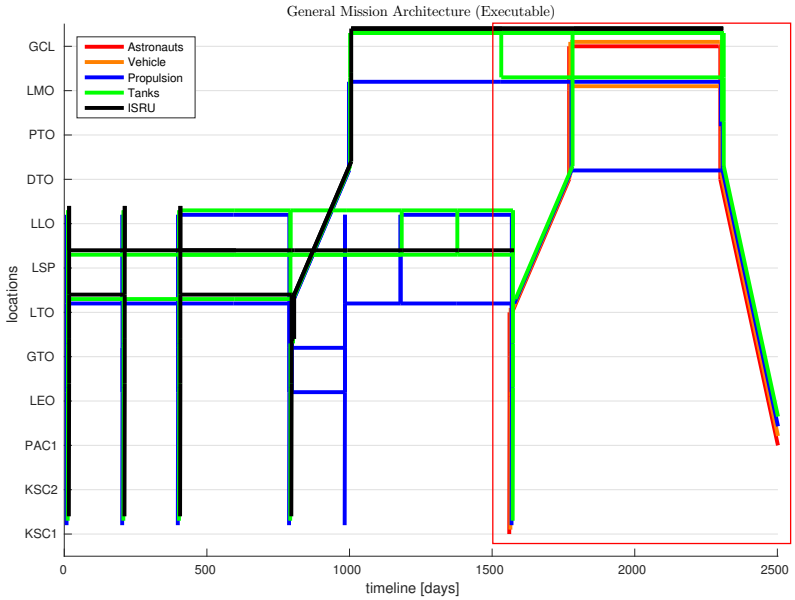



Figure 4-16: Mass flow diagram of the continuous GMCNF. The manned segment is contained in the red rectangle. 5 docking/assembly operations are counted around the manned vehicle. Note that this is 2 less (or a 28% relative reduction) than for the dynamic GMCNF.

The figures show that 7 docking/assembly operations are required for the dynamic GMCNF and 5 are required for the executable GMCNF.

As for the DRA5.0 case, based on fig.4-9, there are 3 dockings/assemblies for the crewed flight: one for joining the Mars transfer vehicle, one for joining the Habitat lander, and one to rejoin the Mars transfer vehicle for the way back.

For the ITS mission, there are 5 docking operations, that all happen in LEO when the refueling ship refuels that Mars transfer vehicle before it heads out to Mars. These results numbers are summerized in fig.4-17

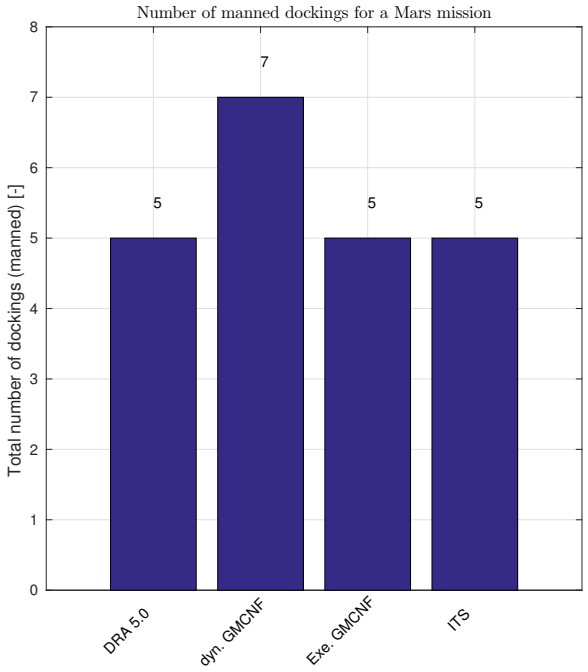



Figure 4-17: Number of dockings for the manned segments of each mission. Note that all are the same 5 dockings except for the dynamic GMCNF that needs a higher number because the mission is more complex

## 4.4 Conclusion

This mission architecture comparative study points out the following elements:

- The GMCNF mission architectures as well as the DRA5.0 missions all have 2 phases: an unmanned, ISRU deployment & collecting phase, and a manned phase. This solution is due to the Martian ISRU that substantially reduces IMLEO and Martian re-entry issues, as the alternative is to carry all propellant with oneself and landing it on the Martian surface.

- As expected, the complexity of the dynamic GMCNF mission is greater than the executable one. This complexity reduction comes at an IMLEO increase. This trend is verified with the ITS mission architecture which has low complexity, but high IMLEO. However, when only the manned portions of the missions are considered, all mission have similar architecture and number of operations, because of the two-segmented aspect of the missions: the manned portion is always a relatively simple flight to Mars and back. The difference in mission architecture is mostly located in the ISRU

portion of the mission. Hence, from a safety perspective, all missions have similar chances of keeping the crew safe, although the GMCNF missions have complexer ISRU segments that are more prone to failure, these only increase probability of loss of mission and not loss of crew.

Although these remarks make a case for executable GMCNF based space mission design, the main drawback is the complexity of the mission. The number of different elements in the network is presented in fig.4-18. This can be mitigated by increasing the size of the elements, however IMLEO is likely to increase as well. This opens new research questions concerning the optimization of hardware size (discretization of the problem) for a given mission campaign.

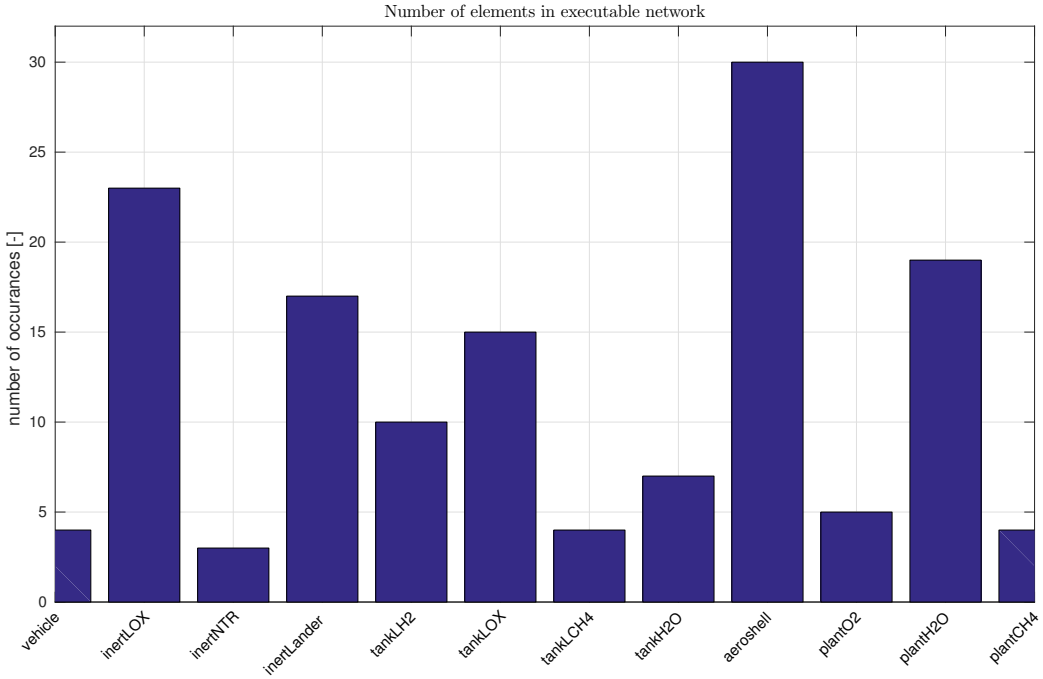



Figure 4-18: .

# Chapter 5

# Case study II: Impact of technological choices on a Mars mission architecture

One of the main added values of GMCNF is the ability to model new technologies such as propellant cooling, NTR, or ISRU, and optimize a complete mission with these integrated into it. This offers substantial help in deciding which technologies to invest in in order to execute a mission at minimum cost. In this chapter, the addition of three non-conventional technologies are investigated:

**Aerocapture:** aerocapture or aerobraking is a process by which $\Delta V$ can be transferred to a spacecraft by flying it through the atmosphere of a planet. This method was used on the Apollo capsules and robotic Mars missions such as *Global Surveyor* [52]. Aerobracking is an active research area because of the benefits it offers for reducing overall spacecraft mass [20].

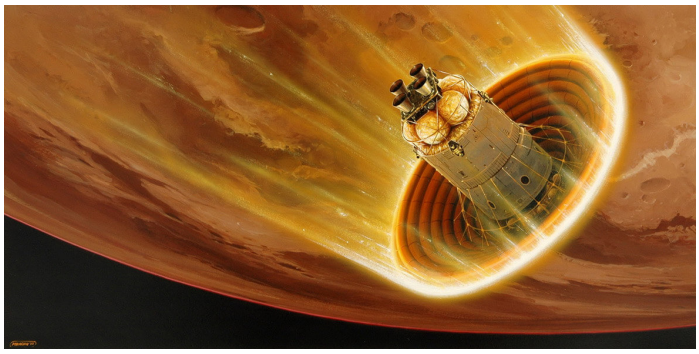


Figure 5-1: Artist rendering of an inflatable heat-shield executing aerocapture in the Martian atmosphere

**NTR:** Nuclear thermal rockets are seen as a promising technology to fly humans beyond the moon because of their high Isp (875-950 sec) [31]. However, these rockets represent a radiation hazard and so far have not flown. Another drawback is their low thrust to weight ratio ( 3:1) [37] compared to that of chemical rocket engines. Many interesting and promising NTR based mission concepts and studies have been proposed,[8],[19], [54] suggesting substancial advantages in using NTRs.

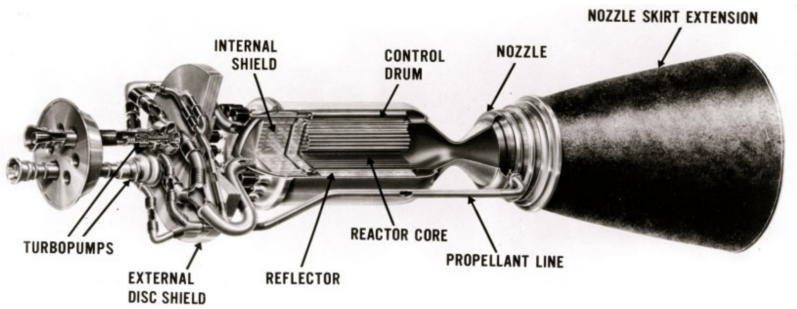



Figure 5-2: Schematic of an "Expander cycle" NTR Engine with dual $LH_2$ turbopumps [10]

A GMCNF optimization can provide a comparison between a mission architecture using NTR and one not using NTR, allowing for a broader study of mission architecture than [8],[19], [54] present.

**ISRU:** Evaluating whether ISRU should be used in one of the main goals of the GMCNF method: ISRU is the technology that gives the GMCNF mission architectures their IMLEO advantage: as can be seen in figs.4-11, the majority of the mission is dedicated in deploying space-based infrastructure that allows for an overall IMLEO that is lower

than that of other mission architectures. Many studies conclude that Mars based ISRU will largely benefit a Mars exploration mission [31], [9], [13] Currently, ISRU technology is not proven, but planned missions such as MOXIE [21] illustrate the importance of this technology in future long duration missions. However, although there is much effort devoted to ISRU on other bodies such as asteroids [53],[66] and particularly the Moon [55], [56], it is still unclear to what extent this ISRU may help in a Mars campaign. This questions is addressed in this thesis using executable GMCNF and varying the lunar ISRU production rate.

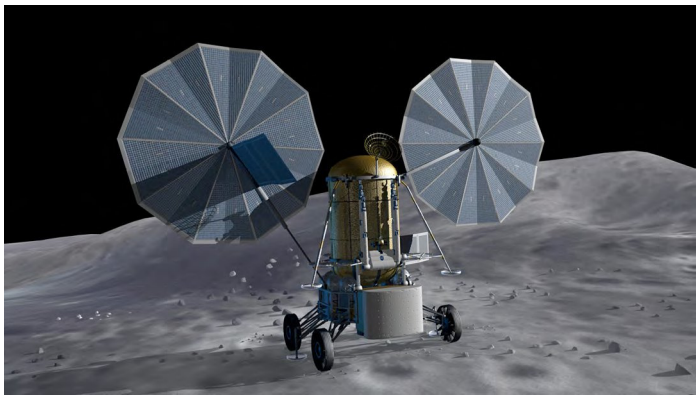


Figure 5-3: Artist rendering of ISRU water extraction unit

## 5.1 Aerocapture

Aerocapture can be used to give a spacecraft a negative $\Delta V$ using an atmosphere. Since aerobraking is an option, this is modeled in the GMCNF network as a parallel edge separately from an transfer edge without aerobraking. Four parallel edges exist between the same end nodes: $LOX/LH_2$ with/without aerobraking and NTR with/without aerobraking. Let $\theta$ denote the aeroshell mass fraction (i.e. if a spacecraft with a mass of 1 performs aerobraking, it must have an aeroshell with a mass of $\theta$). This yields:

$$\mathbf{x}_{ij}^{-} = \begin{bmatrix} \text{spacecraft} \\ \text{aeroshell} \end{bmatrix}_{ij}^{-} \qquad \mathbf{C}_{ij}^{-} = \begin{bmatrix} \theta & -1 \end{bmatrix}_{ij}^{-} \tag{5.1}$$

which is equivalent to

$$[\theta m_{\text{sc}}]_{ij}^{-} \leq [m_{\text{as}}]_{ij}^{-} \tag{5.2}$$

We assume that when performing aerobraking, an aeroshell has a mass of about 15% of the total mass being braked. This leads to the mission architecture shown in fig.5-4: note that although the general mission architecture is not changed, IMLEO goes from 532 MT in the executable GMCNF reference case to 710 MT in the case when aerobracking is not used. This illustrates a large sensitivity of IMLEO to the aerobracking technology.

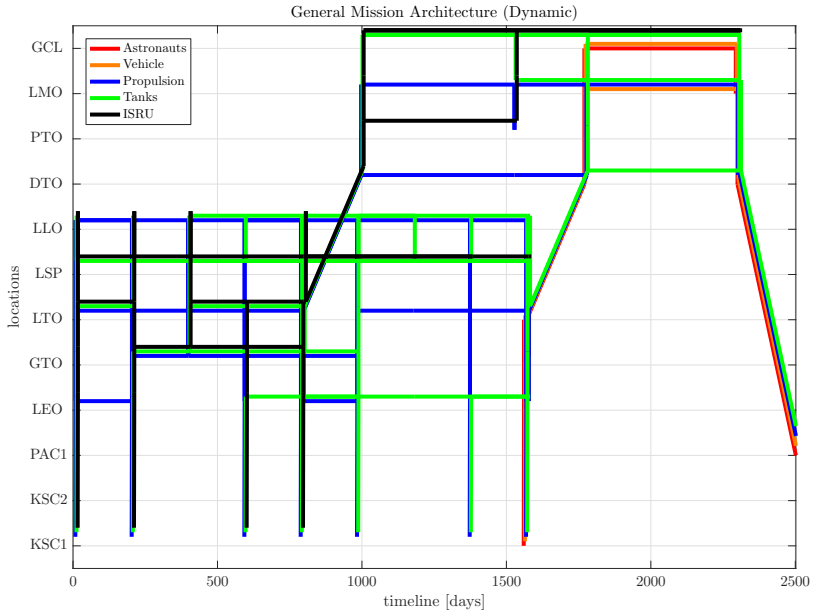



Figure 5-4: Mission architecture used when aerobracking is not used. Although the general architecture does not change, IMLEO increases by 33.0 % because propellant used to perform Martian capture must be carried along.

## 5.2 NTR

NTR propulsion is modeled the same way LOX and methane engines are modeled, however it has an inert mass fraction of 0.3 instead of 0.1 for LOX, and an Isp of 900 seconds instead of 450 seconds. When the NTR option is not considered, the mission shown in fig.5-5 is obtained. The same observation is made here than for the aerobracking removal: the general mission architecture does not change, but IMLEO increases by 27.5 % with respect to the executable GMCNF baseline.

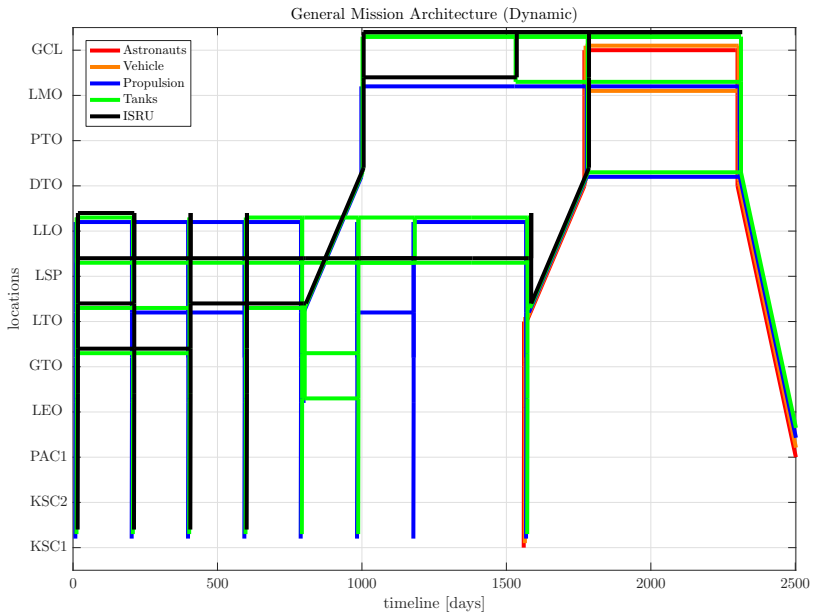



Figure 5-5: Mission architecture used when NTR is not used. Again, the general mission architecture does not change with respect to the base case, but IMLEO increases by 27.5 % with respect to the reference executable GMCNF scenario.

## 5.3 ISRU

The GMCNF missions have two distinct parts, namely a space-infrastructure deploying part and a human flight part. This architecture is radically different from that of the other mission concepts discussed here, where the is no ISRU on the Moon. Because it is widely accepted [13], [9],[21] that IMLEO is very sensitive to martian ISRU, it is the lunar ISRU that is investigated here: because the ISRU on the Moon changes the mission architecture from a direct one to one that flies by the Moon, the goal is to determine at what point the mission architecture changes from one that flies to and from the Moon to one that goes direct, in order to identify the critical lunar ISRU efficiency. This value is compared with those found in [33] and [37]

The method used here is to vary the ISRU productivity rate on the Moon from 5 (the reference value used in [33], [34]and to reduce this value to 0, print the different mission architecture plots and look where the transition occurs between a lunar ISRU architecture and a direct architecture.

The mission architecture using no lunar ISRU is shown in fig.5-6. Note that the architecture is identical to that of DRA5.0 where hardware is sent to Mars, in particular methane

gathering ISRU, and then the crewed mission is sent.

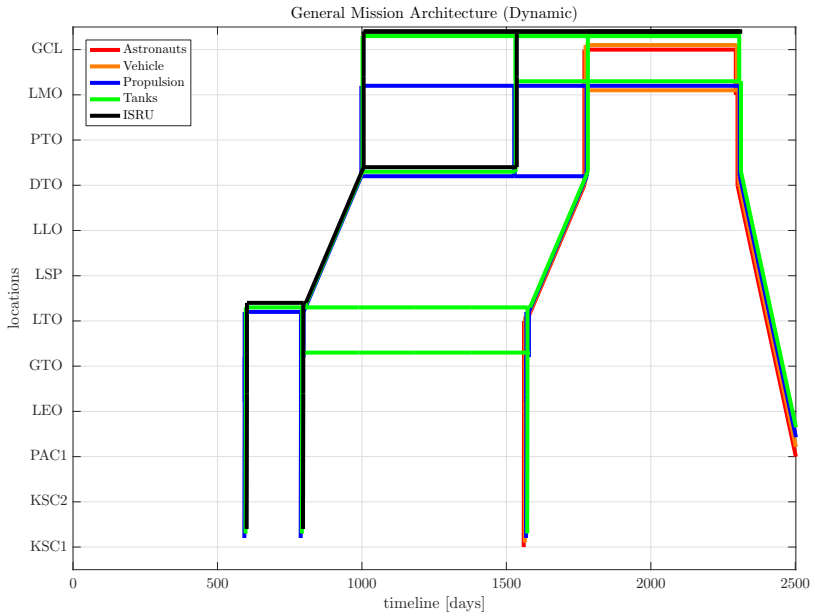



Figure 5-6: Mission architecture to Mars using no ISRU. Note that the architecture is the same as DRA5.0, which contributes to validating the executable GMCNF methodology.

The IMLEO vs ISRU rate plot is shown in fig.5-7, with the critical point where the network changes at a rate below 3 kg propellant/kg plant / year.

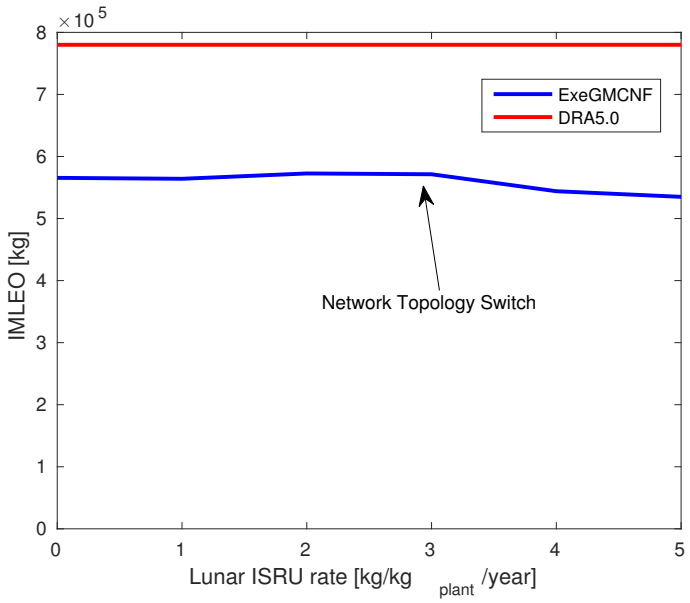



Figure 5-7: IMLEO versus Lunar ISRU rate: note that although lunar ISRU is overall beneficial to IMLEO, it only represents a 5% IMLEO gain at 5 kg/kg/year. An aspect to investigate is the discrepancy between the IMLEO of the dynamic GMCNF at no Lunar ISRU and the DRA5.0 IMLEO: these values should be the same, validating the GMCNF model. [33] uses the same model and falls within 10% of the IMLEO of DRA5.0. The discrepancy presented here is likely due to the Martian ISRU extraction rates that are too optimistic in GMCNF with respect to DRA5.0.

Two comments must be made:

- When lunar ISRU goes to zero, the IMLEO values of DRA5.0 and the executable GMCNF do not converge. This could be due to an overestimate on the ISRU capability on Mars. This verification is left for futur work.

- The difference between the use of lunar ISRU and neglecting it altogether represents only 5% of IMLEO. This results seriously questions the use of developing ISRU technology on the Moon. [**?**] shows a more significant difference between the two. This is illustrated in fig.5-8. Note that in the executable GMCNF, there is still the discretization, or choice of the element sizes, that could change this sensitivity. What is needed here is to do the same trade study, but using element sizes that are comparable to those that are planned to fly to Mars, such as the DRA5.0 vehicles. That study is left for future work.

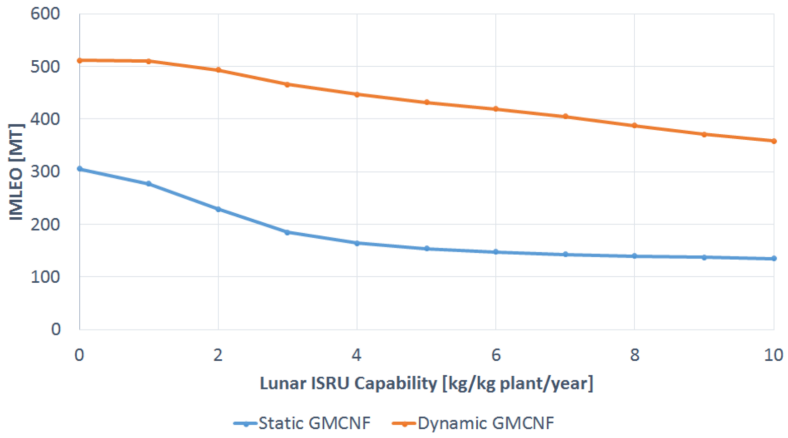



Figure 5-8: IMLEO vs lunar ISRU rate. Note the sharper decrease in IMLEO due to ISRU using static GMCNF. Image from [33].

## 5.4 Conclusion

This chapter investigated the effect of removing technological options from a manned exploration mission to Mars. In terms of IMLEO, it was found that the technologies that most affected IMLEO are aerocapture and nuclear propulsion.

Unlike other studies [33], [37], lunar ISRU is found to be an uneffective means to reduce IMLEO. The mission architecture switch between deploying lunar ISRU and going directly to Mars is found to occur below 3 kilogramm of propellant mined per kilogram of plant per year. The results are summarized in table 5.1.

| Case name | IMLEO [MT] | Difference to ref. case [%] |
|---|---|---|
| DRA5.0 | 848 | +58.8 |
| Exe. GMCNF reference (ref. case) | 532 | +0 |
| Exe. GMCNF (no Aerocapture) | 710 | +33.0 |
| Exe. GMCNF (no NTR) | 681 | +27.5 |
| Exe. GMCNF (no lunar ISRU) | 565 | +5.8 |

Table 5.1: Summary of results concerning technology switches:

# Chapter 6

# Conclusion

## 6.1 Summary of the thesis:

The original goal of this work is to connect two different ways of investigating space logistics: the low fidelity GMCNF and the high fidelity SpaceNet.
The investigation showed that linear GMCNF formulations did not produce executable mission architectures because the event-based simulation requires the tracking of elements throughout the logistics network, which is currently not possible with linear GMCNF results. Note that this characteristic is also what makes a mission executable, condition that must be fulfilled.
The solution to this problem is comprised of three parts:

- Constraining the hardware commodities to integer values, making the tracking throughout the network possible
- Cutoff the optimization process after a feasible solution is found that is close to the global optimum.
- Have a tracking algorithm work through the result in order to trace all elements through the transportation network.

These steps allow for the linking of GMCNF methods and SpaceNet. This task is left for future work.
The executable GMCNF produced mission architectures that were compared to DRA5.0, the previous GMCNF mission architectures and SpaceX's interplanetary transport system.

It was found that executable GMCNF methods produce mission architectures similar to DRA5.0, who has 2 parts, namely one for the ISRU deployement phase and one for the crewed mission segment. It was found that the crewed segments had similar features, such as number of dockings or the number of flight segments.
The second series of analyses concentrated on technological trade studies: it was found that aerocapture and NTR are critcial to IMLEO, but all else being equal do not change overall mission architecture. As for Moon based ISRU, it was found that this technology could only make a 5 % reduction in IMLEO, but with a large complexity burden.

## 6.2 Futur research directions and potential applications


Figure 6-1: Artist rendering of the Zee Aero personal air vehicle. A potential application for GMCNF related methods is the design of urban air mobility networks and vehicles.

This work and previous work on the GMCNF topic is a promising field. The following research directions are suggested:

**MDO for network elements**

Multidisciplinary optimization has gained substantial interest in the past decades [42] because it allows to take advantage of the growing computational power available to the engineer. However, classical MDO is typically concentrated around the design of a well defined systems (typically a aircraft wing) using small number of objective functions and a given set of constraints. However this approach encounters problems when the boundaries of the sys-

tem are pushed out: in the aircraft wing example, this expansion of boundaries could include the entire aircraft but also an entire fleet of aircraft in a given air-transportation network. This problem is discussed in [63], and applications have also been made for space logistics [60],[61],[62]. GMCNF methods could potentially improve the heuristic based methods used in the references cited above. Another potential improvement resides in the multi-commodity aspect of GMCNF, meaning that elements could be sized but also chosen using piece-wise linear approximation.

One direct application of this research proposal is to size/design ISRU equipment by creating models based on other sizing work (such as [58], [59], [45], [26]) for space infrastructure such as propellant depots or ISRU equipment for a manned Mars mission with hardware elements that are already in the testing phase, in order to see to what extent the addition of these technologies can increase the capability of existing spacecraft and hardware.

**Technology deployment and infusion**

Because executable GMCNF optimizes dynamic networks, and can be used for network deployment optimization, technology infusion seems a promising research direction. An example of this are electric cars: because the network of recharging stations is not as widespread as gas stations, there is a network deployment problem which could be solved using executable GMCNF methods. Other technologies, in particular in the renewable energy domain, share this problem. Technology deployment and infusion have many common challenges, and GMCNF methods could complement others such as shown in [23]. Complex product development is also an application for time-expanded networks as shown in [57], an could constitute a potential application as well.

**Transportation optimization**

Transportation optimization seems to be one of the most direct applications of GMCNF, in particular if the transportation has a multi-commodity related constraints. Suggested applications include air-transportation, shipping and rail transportation. Potential applications also surface mission planning and exploring, where network based methods have already been used [41].

**GMCNF multi-objective optimization**

As seen in chapters 4, having IMLEO as objective function produces missions that have many operations and become exceedingly complex. True mission design is a product of many trade offs between competing objectives. An suggested idea is to apply ideas of iso-

performance or multi-objective optimization [22] to these problems, such as cutting the feasible search space at a suboptimal value, and gathering all solutions gathered in that space to do iso-performance of multi-objective optimization on IMLEO, but also other functions (namely ilities, such as complexity).

**Linking GMCNF to other models**

This research direction proposal is mainly focused on linking GMCNF methods to SpaceNet. The challenges here are to automate the event generation using the GMCNF output. However, an architecture of an interface is proposed in appendix B. The author suggests that before these research projects are started, a GMCNF tool be written in order to have proper oversight and insight in the problem being solved, as well as having better visualization tools that would make any further work on GMCNF methods substantially faster than reusing the current code that is targeted specifically to manned Mars campaign design.

# Appendix A

# Intermediate Problem

In a concern of guaranteeing correct results, a smaller analytical problem is set-up in order to gain better understanding of GMCNF modeling and to test the software written for this purpose. The goal is to have a simple problem where a tradeoff can be shown between transforming commodities locally or sending them through a longer path, but with a more efficient conversion rate (this is an analogy to the space logistics problem where the most mass efficient mission does not fly directly to it's destination, but through ISRU re-supply points). The network is constituted of 2 nodes and 4 edges (see fig.**??**). Node $A$ can create commodity $C_1$, but needs $C_2$. $C_2$ can be obtained either by sending $C_1$ along transformation edge 1, or transformation edge 3. The cost of transit on edge 1 is constant, but the transit cost on 2 and 4 are $\phi$, which we vary. Note that edge 3 transforms $C_1$ to $C_2$ according to $C_2 = \chi C_1$.

## A.1 Problem setup:

Node A has a demand of $C_2$ (see caption of fig. **??**) and produces at most 5 counts of $C_1$. If $x_1$ is the amount of $C_1$ sent out from A, we have the following equation if the flow goes entirely through arc 1:

$$\begin{aligned} 2 &= 2x_1 \\ x_1 &= 1 \end{aligned}$$

This holds because the inflow to A of $C_2$ is 2, and the cost is 1 for $C_1$ and $C_2$. Thus this cost is constant to $c = 1 + 1 = 2$.

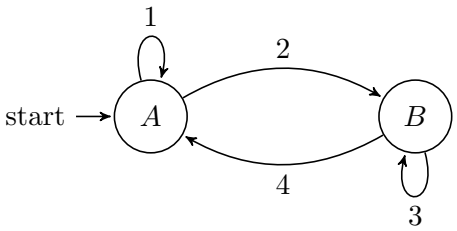



Figure A-1: Graph representation of a smaller test problem. Here we only have 2 commodities, namely a commodity $C_1$ (such as hardware) and a commodity $C_2$ (such as fuel). Here, node $A$ can provide up to 5 counts of $C_1$, but consumes 2 counts of $C_2$. This experiment aims at observing the trade-off between obtaining $C_2$ by transforming it from A (arc 1), or sending it through the network to B, where it can be multiplied at a rate $\chi$. There is a second variable, which is the cost of transportation from node A to B and back, which is $\phi$. This problem represents a reduced version of the reference problem, and can be solved by hand. The transportation cost of all commodities is 1 on arcs 1, $\phi$ on arc 2 and 3, and 0 on arc 4.

If the flow goes through B, we have the following:

$$\begin{aligned} 2 &= \chi x_1 \\ x_1 &= \tfrac{2}{\chi} \end{aligned}$$

The cost is the sum of the flow going along arc 2, $\phi \cdot x_1$, then that going through arc 3, $(x_1 + x_2)$ (because along arc 3 the cost is one) and the returning arc 4, $\phi x_2$. Because $x_2 = \chi x_1$, this yields a total cost of:

$$c_2 = \phi x_1 + x_1 + \chi x_2 + \phi\chi x_1 = x_1\phi\,(1 + \chi) \tag{A.1}$$

Thus, the network topology changes when the costs $c_1$ and $c_2$ equate, giving a relationship between $\chi$ and $\phi$ when the network topology switches. Substituting $x_1$ for $\frac{2}{\chi}$:

$$c_2 = 2\phi\left(\frac{1}{\chi} + 1\right) = c_1 = 2 \tag{A.2}$$

$$\phi = \frac{\chi}{1+\chi} \tag{A.3}$$

or

$$\chi = \frac{\phi}{1-\phi} \tag{A.4}$$

Hence, is the transport cost is too high ($\phi > \frac{\chi}{1+\chi}$) the commodities flow through arc 1, and if the transformation rate is good enough ($\phi < \frac{\chi}{1+\chi}$), the flow goes the other way. Note

that although the optimization problem is linear, *the relationship between the parameters is non-linear*. One can verify this result by letting $\phi \to 1$: this means the transportation cost goes to 1 on arc 4, meaning that no matter how big $\chi$ is, (or how little material we send along arc 2) it will still not be as advantageous as sending the flow through arc 1, where the cost is 1 by definition.

This problem is solved for a variety of $\phi$ and $\chi$. The results are presented in the form of a $\phi$ versus $\chi$ graph (see fig.A-2. The network topology switch can be seen along the line $\chi = \frac{\phi}{1-\phi}$. All edges are represented in order to verify that no other unexpected phenomena occur.

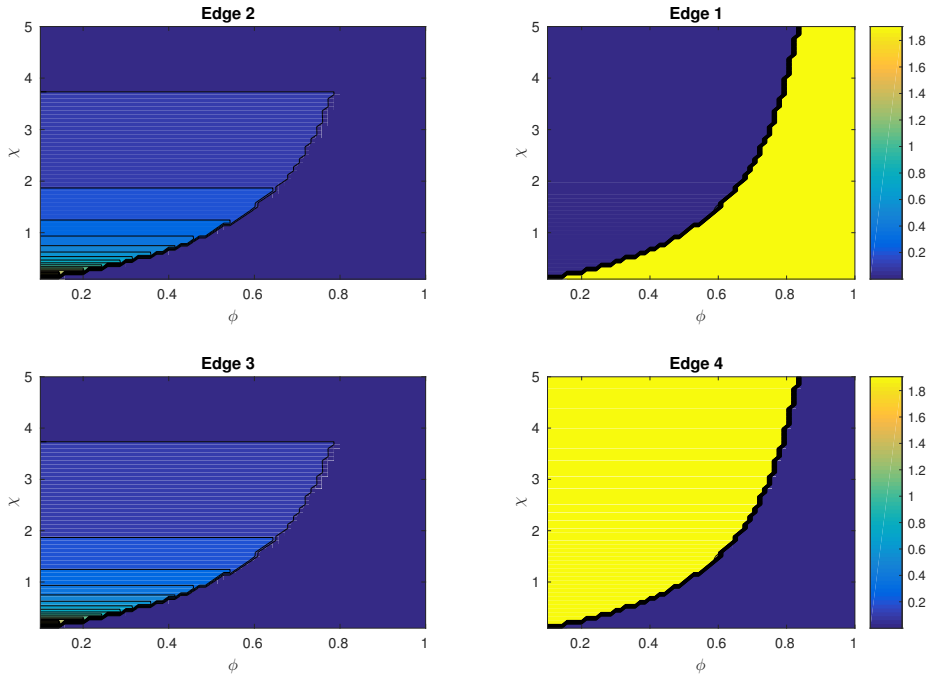



Figure A-2: The result of the study of the intermediate problem: these contour maps represent the total (sum of all commodities) optimized flows through the indicated edge (1,2,3 and 4) for a network problem with the parameters $\phi$ and $\chi$. Note the sharp contrast on edges 1 and 4: this comes from the optimal network topology switch occurring along the analytically calculated line $\chi = \frac{\phi}{1-\phi}$. For the other two edges this is less obvious, as the $\chi$ parameter varies the amount of flow that is needed to be sent from node A, e.g. a higher conversion rate lowers the amount of flow needed to be transported out to B for a given amount of flow being sent back to A.

# Appendix B

# Interface between SpaceNet and GMCNF optimization

## B.1 Requirements for a SpaceNet-INFINITeX interface

The goal of this research is to palliate the GMCNF shortcomings by using SpaceNet in order to check for the detailed feasibility of a given mission architecture.
The following points should be noted:

- Although the static and dynamic GMCNF frameworks can be applied to a large variety of problems, the current softwares (**INFINIT** and its dynamic version) specifically target the optimization of manned missions to Mars.

- The SpaceNet software is designed to allow a user to build a mission with a GUI. When a mission can be simulated successfully, the software can produce certain measures of effectiveness regarding the mission as well a some graphs regarding consumption and production of a variety of quantities such as any forms of supplies or waste. The inputs are cannot be parameterized, and in order to obtain a feasible mission, user input is heavily relied upon.

Because the softwares are so different in nature, it is difficult to link them together. The missions produced by a GMCNF optimization process tend to be large and complexe missions, which are extremely tedious to put into SpaceNet. The following approach is adopted:

- A loose one-way coupling between GMCNF and SpaceNet is provided. In this case, a

tight coupling would require rewriting at least the SpaceNet software, as it currently relies on heavy user input to function properly, which cannot be done if it is supposed to be interfaced with an optimization process.

- The coupling is simple in a programming sense, and the input-output files are written in a human readable format. This allows for easy verification of the data. This approach will require user input to run the different softwares, but this is unavoidable with the low level of automation of SpaceNet. It is rather the SpaceNet software that is expected to be ran more often (possibly in an automated fashion), in order to obtain sensivities and to provide a solution that is feasible.

  The main concern here is to avoid a software that makes too many simplifying assumptions and whos results cannot easily be verified (e.g. a black-box type).

## B.2 Software architecture

The general software architecture to link both programs is depicted in fig.B-1

This architecture is chosen for the following reasons:

**No new type of software:** all files and codes are wirtten either in Matlab (the language of GMCNF) or in Java (language of SpaceNet). All the information is passed using input-output (io) files, in formats used by both SpaceNet and GMCNF: excel files. The .xml file is used specifically by SpaceNet.

**Low coupling** it is decided to add separate code modules instead of changing the legacy code, in order to reduce the risk of errors, and being able to test the new code modules properly.

**Clear data transfer** by passing the data between the codes using excel files, one can correct and check the output/input data. This does not work for the .xml files, but xml format is needed in order to communicate with the SpaceNet software.

The major drawback of this layout is the relatively large number of files. However, this approach allows modularity and easier debugging which has a greater value in this particular application. The next paragraphs present the different components of the link between SpaceNet and **INFINITeX**:

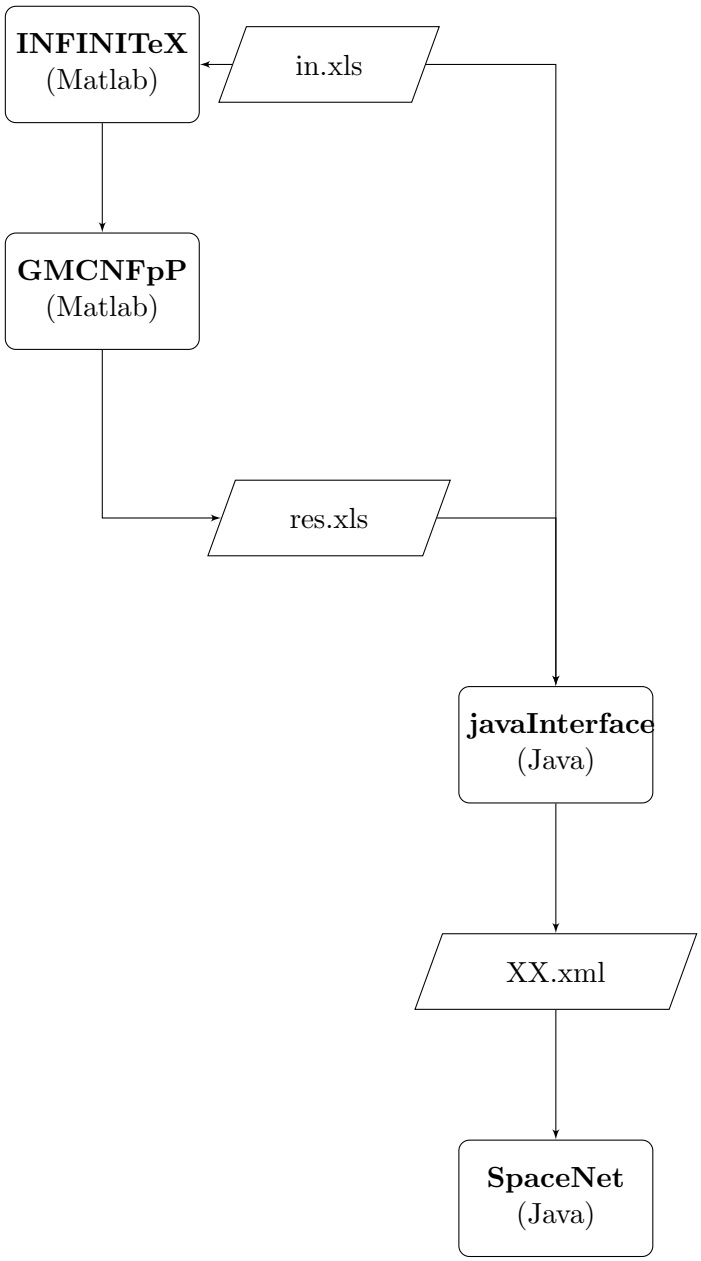



Figure B-1: At first there is the matlab module that produces the optimized transportation network, **INFINITeX**, which corresponds to Ho's work [33]. This feeds **GMCNFpP.m**, which post-processes the restults into an excel file that will then be read by the **javaInterface**, which makes use of SpaceNet's classes to write an .xml file which is then read by **SpaceNet**.

### B.2.1 GMCNF post-processing (GMCNFpP)

The integer GMCNF optimization produces a network in which the elements can be traced. The tracing and automatic definition of the events such as transfer of propellant from a full tank to an empty tank is done by a matlab function that produces an an excel spreadsheet ( res.xls in fig.B-1

### B.2.2 Input file (in.xls)

The input file is the input file of **INFINITeX** with some additional data concerning the nodes and the vehicles in order to construct a SpaceNet mission.

### B.2.3 Post-processed GMCNF file (res.xls)

The post-processed file is an excel file containing the elements and events in order to build the SpaceNet mission.

### B.2.4 Java interface

The Java interface is a simple class that inherits all the SpaceNet methods and is capable of reading excel files. The elements and events are read from the excel file xx.xls, saved as Java variables and the saved into an XML file that can be read by SpaceNet.

## B.3 General remarks

Note that this architecture serves only as a suggestion for futur implementation, none of the modules presented here has been coded yet. This is left for futur work.

# Bibliography


[1] Cost of mass in leo. http://space.stackexchange.com/questions/1989/what-is-the-current-cost-per-pound-to-send-something-into-leo. Accessed: 2016-07-01.

[2] CPLEX website. http://egon.cheme.cmu.edu/ewo/docs/rlima_cplex_ewo_dec2010.pdf.

[3] Gurobi Website. url/http://www.gurobi.com/resources/getting-started/mip-basics.

[4] Otrag company article. http://www.astronautix.com/o/otrag.html. Accessed: 2016-10-02.

[5] Project HARP. http://www.astronautix.com/graphics/m/mart3gun.jpg.

[6] SSME specifications. http://www.rocket.com/space-shuttle-main-engine.

[7] *Nuclear Pulse Propulsion Orion and Beyond*, 2000.

[8] *TRITON: A TRImodal capable, Thrust Optimized, Nuclear Propulsion and Power System for Advanced Space Missions*, 2004.

[9] *In-Situ Resource Utilization on Mars - Update from DRA 5.0 Study*, 2010.

[10] *NUCLEAR THERMAL ROCKET (NTR) PROPULSION: A PROVEN GAME-CHANGING TECHNOLOGY FOR FUTURE HUMAN EXPLORATION MISSIONS*, 2012.

[11] *Making humans a multiplanet species*, 2016.

[12] L. DeLucas A. McPherson. Microgravity protein crystallization. *Nature*, 2015.

[13] E. Santiago-Maldonado A. Muscatello. Mars In Situ Resource Utilization Technology Evaluation. *50TH AIAA Aerospace Sciences Meeting; 9-12 Jan. 2012; Nashville, TN; United States*, 2012.

[14] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows: theory, algorithms and applications*, volume 1. 1993.

[15] Dale C Arney and Alan W Wilhite. Modeling Space System Architectures with Graph Theory. *Journal of Spacecraft and Rockets*, 51(5):1413–1429, 2014.

[16] David Baker. *Jane's Space Directory*. 2002.

[17] W. Berry and H. Grallert. Performance and technical feasibility comparison of reusable launch systems: A synthesis of the ESA winged launcher studies. *Acta Astronautica*, 38(4-8):333–347, 1996.

[18] Dp Bertsekas. *Network Optimization: Continuous and Discrete Models*, volume C. 1998.

[19] S. Borowski. "bimodal" nuclear thermal rocket (bntr) propulsion for future human mars exploration missions. "BIMODAL" NUCLEAR THERMAL ROCKET (BNTR) PROPULSION FOR FUTURE HUMAN MARS EXPLORATION MISSIONS (Technical presentation). Technical presentation at 2003 NASA Seal / Secondary Air System Workshop.

[20] P. Esposito D.Johnston W. Willcockson D. Lyons, J. Beerer. Mars Global Surveyor: Aerobraking Mission Overview. *Journal of Spacecraft and Rockets*, 36(3), 1999.

[21] F. Meyen D. Rapp, J. Hoffman. The Mars Oxygen ISRU Experiment (MOXIE) on the Mars 2020 Rover. *SPACE Conferences and Exposition 31 Aug-2 Sep 2015, Pasadena, California AIAA SPACE 2015 Conference and Exposition*, 2015.

[22] Olivier L. de Weck and Marshall B. Jones. Isoperformance: Analysis and Design of Complex Systems with Desired Outcomes. *Systems Engineering*, 9:pp. 45–61, 2008.

[23] Kenneth J. Mihalyov Olivier de Weck Eun Suk Suh, Michael R. Furst. Technology Infusion for Complex Systems: A Framework and Case Study. *Wiley InterScience*, 2008.

[24] Andy Evans, Olivier de Weck, Deanna Laufer, and Sarah Shull. Logistics Lessons Learned in NASA Space Flight. (May), 2006.

[25] William A Andy Evans and United Space Alliance. Logistics Lessons Learned in NASA Space Flight Interplanetary Supply Chain Management & Logistics Architectures. 2007.

[26] Jonathan A. Goff, Bernard F. Kutter, Frank Zegler, Dallas Bienhoff, Frank Chandler, and Jeffrey Marchetta. Realistic Near-Term Propellant Depots: Implementation of a Critical Spacefaring Capability. *American Institute of Aeronautics and Astronautics*, pages 1–18, 2009.

[27] P Grogan. *A FLEXIBLE, MODULAR APPROACH TO INTEGRATED SPACE EXPLORATION CAMPAIGN LOGISTICS MODELING, SIMULATION, AND ANALYSIS*. PhD thesis, MIT, 2010.

[28] Paul Grogan. SpaceNet 2.5r2 Quick Start Tutorial.

[29] Paul Grogan. SpaceNet 2 . 5r2 User ' s Guide, 2010.

[30] Paul Grogan, Afreen Siddiqi, and Olivier de Weck. Matrix Methods for Optimal Manifesting of Multi-Node Space Exploration Systems. *AIAA SPACE 2010 Conference & Exposition*, 48(4):1–32, 2010.

[31] Mars Architecture Steering Group. Human Exploration of Mars Design Reference Architecture 5.0 , 2009.

[32] Greg J. Gstattenbauer. Cost Comparison Of Expendable, Hybrid and Reusable Launch Vehicles. (September):1–11, 2006.

[33] Koki Ho. *Dynamic Network Modeling for Spaceflight Logistics with Time-Expanded Networks*. PhD thesis, 2015.

[34] Koki Ho, Olivier De Weck, Jeffrey Hoffman, and Robert Shishko. Dynamic network modeling for spaceflight logistics. *AIAA Modeling and Simulation Technologies Conference 2014*, 2014.

[35] Koki Ho, Olivier L. De Weck, Jeffrey A. Hoffman, and Robert Shishko. Campaign-level dynamic network modelling for spaceflight logistics for the flexible path concept. *Acta Astronautica*, 123:51–61, 2016.

[36] Abdelkrim Doufene Takuto Ishimatsu Daniel Krob Hugo G. Chale-Gongora, Olivier de Weck. Planning an Itinerary for an Electric Vehicle. 2014.

[37] T Ishimatsu. *Generalized multi-commodity network flows: case studies in space logistics and complex infrastructure systems*. PhD thesis, MIT, 2013.

[38] Takuto Ishimatsu. What is Humanity ' s " Best " Path to Mars ? Introduction : Takuto Ishimatsu. 2015.

[39] Takuto Ishimatsu, Olivier de Weck, Jeffrey Hoffman, Yoshiaki Ohkami, and Robert Shishko. A generalized multi-commodity network flow model for space exploration logistics. *AIAA SPACE 2013 Conference and Exposition*, 2013.

[40] Takuto Ishimatsu, Olivier L. de Weck, Jeffrey A. Hoffman, Yoshiaki Ohkami, and Robert Shishko. Generalized Multicommodity Network Flow Model for the Earth–Moon–Mars Logistics System. *Journal of Spacecraft and Rockets*, 53(1):25–38, 2015.

[41] OLIVIER DE WECK JAEMYUNG AHN and JEFFREY HOFFMAN. AN OPTIMIZATION FRAMEWORK FOR GLOBAL PLANETARY SURFACE EXPLORATION CAMPAIGNS. *JBIS*, 61:pp. 487–489, 2008.

[42] Jaroslaw Sobieszczanski-Sobieski Paul Arendsen Alan Morris Martin Spieck Jeremy Agte, Olivier de Weck. MDO: assessment and direction for advancement—an opinion of one international group. *Struct Multidisc Optim*, 2009.

[43] Kumiko Kiuchi. Nuclear Pulse Propulsion - orion and beyond. *Journal of Beckett Studies*, 18(July):72–87, 2008.

[44] Dietrich E. Koelle. Economics of small fully reusable launch systems (SSTO vs.TSTO). *Acta Astronautica*, 40(2-8):535–544, 1997.

[45] Nasa. Propellant Depot Requirements Study Status Report. *HAT technical interchange meeting*, 58(3), 2011.

[46] Robert Shishko Olivier L. de Weck, David Simchi-Levi. SpaceNet 1.3 User ' s Guide, 2007.

[47] Olivier L. de Weck Paul T. Grogan, Chaiwoo Lee. Comparative Usability Study of Two Space Logistics Analysis Tools. *AIAA SPACE 2011 Conference & Exposition 2011*, 2011.

[48] Olivier L. de Weck Paul T. Grogan, Howard Yue. Space Logistics Modeling and Simulation Analysis using SpaceNet: Four Application Cases. *AIAA SPACE 2011 Conference & Exposition 2011*, 2011.

[49] Olivier L. de Weck Paul T. Grogan, Howard Yue. Logistical Analysis of a Flexible Human-and-Robotic Mars Exploration Campaign. *JOURNAL OF SPACECRAFT AND ROCKETS*, 48(51), 2014.

[50] Sebastian Raggl, Judith Fechter, and Andreas Beham. A dynamic multicommodity network flow problem for logistics networks. *27th European Modeling and Simulation Symposium, EMSS 2015*, pages 301–306, 2015.

[51] Steffen Rebennack. Ellipsoid Method. *Encyclopedia of Optimization*, pages 890–899, 2008.

[52] Robert D. Braun Reuben R. Rohrschneider. Survey of Ballute Technology for Aerocapture. *Journal of Spacecraft and Rockets*, 44(1), 2007.

[53] Shane D Ross. Near-Earth Asteroid Mining. *Space Industry Report*, (November):1–24, 2001.

[54] D. McCurdy T. Packard S. Borowski. Conventional and Bimodal Nuclear Thermal Rocket (NTR)Artificial Gravity Mars Transfer Vehicle Concepts. *AIAA 50 th Joint Propulsion Conference and Exhibit, Cleveland, OH, July 28 - 30, 2014*, 2014.

[55] Samuel S. Schreiner, Jeffrey A. Hoffman, Gerald B. Sanders, and Kristopher A. Lee. Integrated modeling and optimization of lunar In-Situ Resource Utilization systems. *IEEE Aerospace Conference Proceedings*, 2015-June:1–11, 2015.

[56] Samuel Steven Schreiner. *Molten Regolith Electrolysis Reactor Modeling and Optimization of Insitu Resource Utilization Systems.* PhD thesis, MIT, 2015.

[57] Matthew R. Silver and Olivier L. de Weck. Time-Expanded Decision Networks: A Framework for Designing Evolvable Complex Systems. *Wiley InterScience*, 2006.

[58] David Street and Alan Wilhite. 1 American Institute of Aeronautics and Astronautics. pages 1–13, 2006.

[59] Christopher Tanner, James Young, Robert Thompson, and Alan Wilhite. On-Orbit Propellant Resupply Options for Mars Exploration Architectures. *Earth*, pages 1–15.

[60] C Taylor, M Song, and D Klabjan. A mathematical model for interplanetary logistics. *Logistics Spectrum*, page 20, 2007.

[61] Christine Taylor, Miao Song, Diego Klabjan, Olivier L De Weck, and David Simchilevi. Modeling Interplanetary Logistics : A Mathematical Model for Mission Planning. *Elements*, pages 1–15, 2004.

[62] Christine Taylor and Olivier L De Weck. Case Study of Earth-Moon Supply Chain Logistics. *Transportation*, (April):1–13, 2005.

[63] Christine Taylor and Olivier L De Weck. Integrated Transportation Network Design Optimization. (May):1–16, 2006.

[64] David R. Tobergte and Shirley Curtis. Empirical complexity analysis of a MILP-approach for optimization of hybrid systems. *Journal of Chemical Information and Modeling*, 53(9):1689–1699, 2013.

[65] Dashen Xue, Zhaohui Li, and Nan Xue. Multi-commodity logistics network design based on heuristic algorithm. *Lecture Notes in Electrical Engineering*, 139 LNEE:75–80, 2012.

[66] Kris Zacny, Phil Chu, Jack Craft, Marc M. Cohen, Moffett Field, Warren W. James, and Brent Hilscher. Asteroid Mining. *American Institute of Aeronautics and Astronautics*, pages 1–16, 2013.