**Part B)**

**B1) Simulation Completion**

*ISRU Efficiency:* We determined that the ISRU efficiency will not have a significant effect on IMLEO required for each resupply mission due to our analysis of the steady-state system. The main effect of ISRU efficiency on steady state IMLEO (and resulting science value) is the spares mass which accounts for approximately 0.02% of the total IMLEO for any given resupply. Calculating total cost and science utility with the initial emplacement is important and will be part of the future work of this project. However, for this section of the analysis we removed it to focus on development costs associated with future propulsion systems which drive IMLEO for each resupply.

*Launch Cost:* In order to maximize the science return for the mission, we calculated a launch cost based on a constant dollars per mass rate for IMLEO. We chose use the conservative estimate of $10k/kg based on historical systems (Jones, “Estimating the Life Cycle Cost of Space Systems.” 2015). However, the results are expected to be sensitive to this parameter and its sensitivity should be investigated.

*Development Cost:* The development cost necessary for the new propulsion systems was also calculated using the Advanced Missions Cost Model (Jones, 2015). In this model, cost is a function of objective variables like system mass and time to development, and more subjective ones such as ‘difficulty’. These more subjective parameters can also be investigated as future work. For this analysis, we used the values recommended in the reference.

*Utility:* We calculated the utility of each architecture based on the amount of scientific value that the surface crew could provide based on the number of crew, their available hours and the scientific interest of landing site.

*Remaining Issues:* Finally, the model has so far been looking at the transit spacecraft as a single-use vehicle. Although this is a fair assumption considering historic comparables and the lack of demonstrated on-orbit maintenance, re-usable transit vehicles could significantly change the results (especially in the case of NTR reactors), and we hope to look into this further if time permits. Additionally, in order to wrap the Launch Cost together with the Development Cost we have looked at two methods. First off, and for this assignment, we assumed that this will be a campaign over 10 resupply missions, and have thus set the total cost at Dev\_Cost + 10\*Launch\_Cost. We have also looked into using a form of net present value discounting considering an infinitely repeated (or repeated for a long but unknown duration) future expense as such: CostNow + RepeatedCost/r, where r is the discount rate. While r could be baselined at inflation, determining the correct use of r, including factoring in the risk of cancelling the campaign (and thus no longer making use of the development in this context) is a complicated decision involving a lot of assumptions. As such, we have decided to continue with the 10 launch scheme until we can come up with justifiable assumptions to estimate the size of r. Note, however, that an ‘interest’ rate of 1% (0.01) is equivalent to 10 repeated missions, and as such, these two formulations are quite similar (or similarly, pegging the ‘interest’ rate to inflation at 2.9% is the same as calculating the costs over 29 missions.) Thus we may end up using either one, with preference to the formulation with the most justifiable numbers.

**B2) Heuristic Optimization**

1. We chose to use genetic algorithms for our Heuristic Optimization. Our simulation takes a little over 0.5 seconds for each run, and GA provides the ability to run many of these simulations in parallel, since it investigates an entire generation at the same time. The code had previously been configured to run in parallel, and we postulated that the time savings of 4X-8X faster would be valuable.

Additionally, our group felt that the advantage of design space exploration provided by GA was an interesting aspect and we wanted to investigate this further.

1. We chose to focus on the overall cost for this problem, and intend to look at Utility when we add multi-variable optimization to the problem. Since utility involves mostly correlated aspects, there are usually simple decisions with few tradeoffs in order to maximize it, which makes the optimization problem relatively trivial. Contrariwise, the total cost contains conflicting aspects, namely, increases in development cost are applied to reductions in ongoing launch costs. As such, focusing on Total Cost for the single objective optimization allows for non-trivial use of the optimization algorithm.

The design variables for this exercise were constrained to the increase in Isp (with its related development costs), and the percentage of food grown on Mars. Additionally, for this algorithm, we decided to test with net present value scheme for future launch costs, using the current inflation rate of 2.9%, which is mathematically equivalent to adding in the launch costs from 29 resupply missions. Finally, since Isp and Food % (both continuous) were the only design variables, the architectural selections were fixed, and thus NTR was not considered in this optimization problem.

For the first run, we used the mid-range parameters of Pop\_Size = 30, Crossover\_Frac = 75%, Elite\_Count = 1, and Termination\_Tol = 1x10^-6, and the Genetic Algorithm terminated after 5 generations with the solution of Isp = 465 and Food % = 0.75. The following table shows the results as we alter the tuning parameters.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Pop Size | Crossover | Elite Count | Tolerance | # Gen | #Func | Isp | Food Grown  On Mars |
| 30 | 75% | 1 | 1e-6 | 5 | 6180 | 448 | 75% |
| 20 | 75% | 1 | 1e-6 | 5 | 4120 | 448 | 75% |
| 20 | 85% | 2 | 1e-6 | 5 | 4120 | 448 | 75% |

More runs were originally planned, but it quickly became clear that the optimal solution is to minimize investment when considering only LH2 with the design variables of Isp and Food %, while only looking at the objective of total cost.

1. Our confidence that this is the best way to optimize for *cost alone* in relatively high. The algorithm has been validated, and even with the exploration of GA tuning parameters as shown in the above table, the results have remained largely the same. We think this has to do more with the convolution of the cost objectives; essentially, many of the architectural choices trade off increased cost with increased utility, and by ignoring the utility, we simply settle on the lowest cost solution.

However, we highly doubt the optimality of this solution when considering the multi-objective problem of balancing the cost with the utility.

Since the single-objective optimization results here have made it clear that investment is not worth it at a rate of return (r) equal to inflation at 2.9%, we decided to run a small sensitivity analysis on r, using the tuning parameters in run 2 above. The increases in rate of return could be correlated to the opportunity cost of missing out on the utility due to *not* running the exploration activities.

As can be expected by the fact that Food Grown is largely a trade off recurring cost and utility, this single-objective algorithm optimized *just cost* and kept the food % at 75% grown on mars. However, as the weight of the recurring launch costs increased (via the increasing rate of return percentages, or weight given to the *importance of future value*), from 2.9% to 5%, the algorithm found the optimal investment in Isp improvements to be increasing as well, approaching the maximum of 480.

We can learn from this that our confidence in the algorithm is based on two primary factors. Firstly, the increased interactions from running the **full** design space through **all** of the multiple objectives will be very important in arriving at an optimal, or near-optimal, solution. Secondly, the weighting given to the importance of future value plays an important role in the confidence of the solution, and should be considered when deciding whether to select a pre-weighted scheme, or to take an a-posteriori perspective when selecting multi-objective algorithms.

**B3) Local Derivative-Free Search**

**i) Algorithm Selection**

Although the design space for this problem consists of discrete feasible points, we attempted to use a local derivative-free search algorithm to find the optimum set of variables to maximize the science value per dollar of the missions. We tested and modified a standard Matlab derivative-free search method first before deeming it necessary to develop a custom algorithm to best search our design space.

*Nelder-Mead Method:* The Nelder-Mead method provides local search in multidimensional space through an interesting heuristic method using a morphing simplex polytope. We used Matlab’s fminsearch function to implement that algorithm for our problem. Integrating our code with Matlab’s function required converting our discrete design space into a continuous one for the ND algorithm to search. We completed this by rounding values to the nearest integer and including a penalty function when evaluating designs outside of the bounds. This made the resulting behavior of the algorithm very sensitive to the step size, especially for variables with very few alternatives (eg power: solar PV or nuclear). We were able to adjust the step size within the fminsearch function, but were not able to find a proper step size to scale with each variable in such a small, discrete space using the rounding conversion. This led us to try an alternative method.

*Coordinate Search Variation:* We developed a custom gradient-free local search method that we felt would work well search the unique design space. This allowed us to restrict the search to discrete variables while still searching in the local region around the current solution. The pseudocode for our algorithm is:

1. Choose initial guess X0
2. For each variable
   1. Evaluate the objective at two neighboring (forward and back) variable alternatives for each design variable while holding all others constant.
3. If a better objective is found
   1. Move to X1 corresponding to the best objective found.
   2. Repeat from step 2.
4. If no better objective is found
   1. Terminate search and save final X and objective value.
5. End

The algorithm works well for our problem because it is able to efficiently search the space using knowledge of the ‘distance’ between each non-dimensional, discrete alternatives and the bounding of the problem. No other prior knowledge is necessary. Due to the local search nature of this algorithm, it does not guarantee to find a global minimum, but rather may become trapped in local minimums.

*Full Factorial Evaluation:* We performed a full factorial evaluation of our design space as it is relatively small in its current form with 1050 feasible design points. Using the initial default parameters ($10k/kg launch cost, 10 resupply missions), the optimum design was:

X\* =

Propulsion – LH2, Isp 452

Surface Power – Solar

Location – Gusev

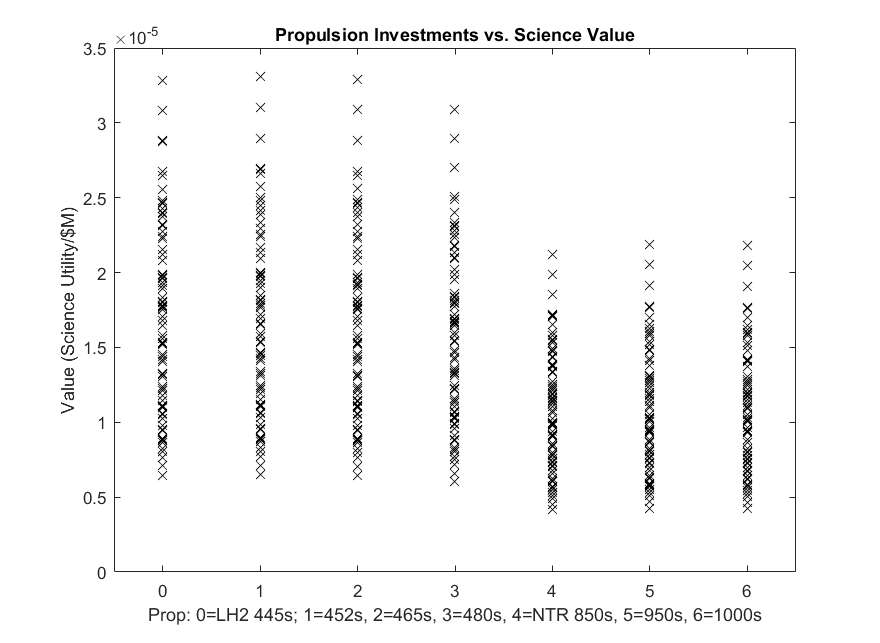
Food – 100% Earth (max)

Surface Crew – 24 (max)

Value(X\*) =

3.31e-5 ScienceUtility/$M

This allows us to compare the results of our local and global search methods with the correct global optimum and confirm that it may be used to properly interrogate the design space. The figure below shows all of the architectures with their scientific value plotted against propulsion options (technology and Isp). Here we can see the global optimum (LH2 452s) and a local optimum using NTR with Isp of 950s, circled in green and orange, respectivley.



**ii) Single Objective Optimization**

Having developed the coordinate search algorithm and developed the full factorial design space to test the results against, the optimization technique was executed for a variety of starting points with the objective of maximizing the science value per million dollars spent. The results of these optimizations are recorded in the table below.

The starting points (initial guesses) cover the bounds of the design space as well as sample the interior. As can be seen through the color coded optimal solution column, four different maximum values were found in the design space. Three of these, indicated in orange below, were determined to be local maximums buy comparing their “value” metrics. The optimizer also identified the global minimum from three of the starting points attempted. We were quite confident that the optimizer had found the global minimum due to the spread of starting points we began with, however we confirmed that this was indeed the global optimum by checking with the DOE.

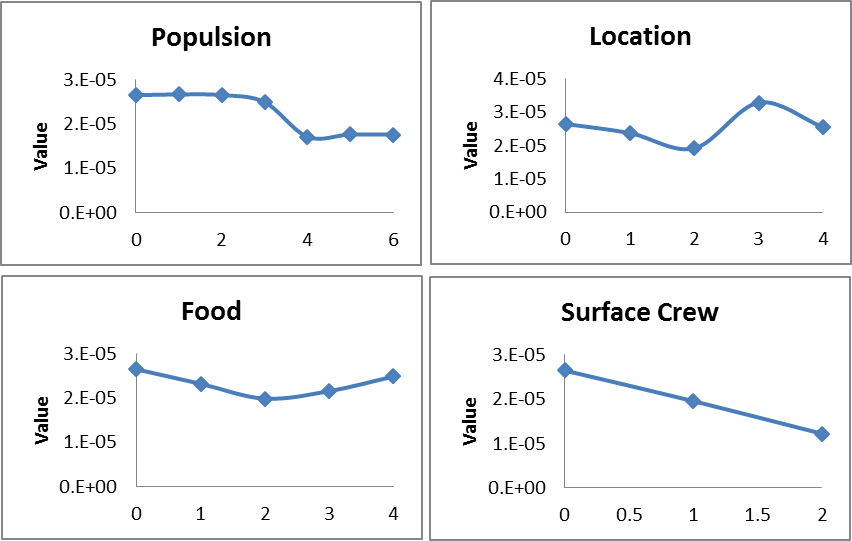


Despite using 10 starting points to characterize the design space, the solver still used less than half the function calls necessary to create the DOE and did a good job of characterizing the design space. The function typically converged quickly within 50 function evaluations, and sometimes many fewer. Furthermore, it was immediately recognized that the current convergence criteria (which checks the same point 3 times to ensure there are no other options) is unnecessary and could be cut to one convergence check to reduce the computation needs.

This bodes well to use this method as a quick way to roughly find the global maximum and local maximum points in this design space, especially as it expands in size. We therefore explored other mechanisms by which to make the search algorithm faster. The table below represents a “heat map” of the six design variables where the darker green represent higher performing (more value/dollar) points within the design variable (holding all other design variables constant) and the lighter green are design options that deliver less value.



This table is useful as it communicates a wide variety of information about the design space. We can see, for example, that depending upon where the initial value is within propulsion, the non-derivative based optimizer will settle either on index 1 (LH2-452), or index 5 (NTR-950), where the nuclear engine represents a local minimum. Similarly, the Location and Food design variables also have a bimodal distribution that creates local minimums in the design space. To better understand the sensitivity of the objective to each design variable. The following plots were created that show more detail of the variance in each design variable, holding all others equal. The bimodal maximum points can easily be seen in the Propulsion, Location, and Food plots.



In order to increase the efficiency of our algorithm and reduce the chance the algorithm would become stuck in a local maximum point, we reordered the indices of each dependent variable to best order the variables from increasing to decreasing value for our design point, X\*, and remove local maximums. We note, of course, that the relative value of these design variables is likely to change due to interaction effects between the design variables. However, as we were interested in solving the solution specifically around X\*, reordering based upon the general value provided at X\* can provide significant time savings and help remove local maximums in this section of the design space. The table below displays the reordered design variables and displays how the opportunity for local maximums has been removed, except in the Propulsion category where the original order was deemed necessary to maintain.

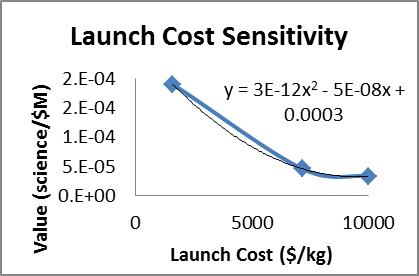


The new design variable order was implemented (although the convergence criteria was kept the same to understand the impact of the reordering alone), and the following tests were conducted. These three starting points tested the bounds of the design space and the midpoint of the Propulsion design variable to ascertain that with only three runs of the optimizer, the global maximum and single local maximum of the design space could be found. It should be noted that in general each run of the optimizer tool longer as it was traversing a greater portion of the design space without encountering a local maximum, but the total computational time requirements were reduce by over 75% compared to our first attempt.



**iii) Sensitivity Analysis**

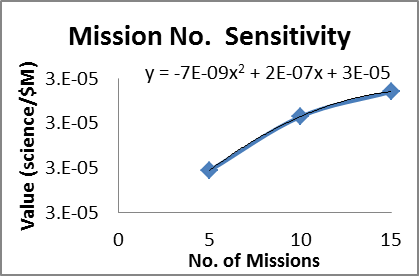
The above section discussed sensitivity analyses conducted for the design variables about the point X\* as part of the process of improving the optimization algorithm and refining the result. Please see above for a discussion of the design variable sensitivities. However, in addition to the design variables, this section b1 suggested that the launch cost from earth as well as the number of missions conducted for the 2040 Mars campaign would be parameters that the value of the mission may be highly dependent upon. Therefore, we chose to vary these two parameters as part of the sensitivity studies.

The first sensitivity study used our refined optimizer to determine the sensitivity of the value per million dollar mission cost to the launch costs to IMLEO. Three launch costs were utilized, the $10,000/kg value assumed as the baseline for today’s capabilities. A $7,140/kg estimate for the SLS, and a $1,600/kg estimate for the ultimate costs of a SpaceX Falcon 9 Heavy launch. The table below displays the global maximum findings from our algorithm and the accompanying table shows the sensitivity may be described by a quadratic trendline.



The findings of the sensitivity study meet the intuition of our team, albeit to a greater degree. We expected a reduction in launch costs to improve the objective, however we did not expect just how strongly it improved the objective value, and the non-linear nature of the sensitivity. Furthermore, it was a surprise to learn that with extremely reduced launch cost values such as those proposed by SpaceX, the relatively minor investment for even marginally higher Isp values does not pay off and the missions may choose to use only existing technology and not invest in advanced chemical propulsion.

Finally, the second sensitivity study conducted was to determine the impact of changing the parameter that controls the number of missions flown during the Mars 2040 campaign. The table and figure below present the optimal solutions and objective values optimal campaign architecture considering the different number of missions.



The findings of the mission number sensitivity study were directly in line with those anticipated by our team. As the number of missions in the campaign increases, the high initial NRE development costs are spread out over a greater number of missions, and the benefits (especially of reduced propellant required due to higher Isp’s) make higher upfront technology investment more optimal. This can be seen when with a 15 mission campaign the optimal engine type flips to the LH2 engine with a 465s Isp.