A fast, robust genetic algorithm for producing families of constrained multi-burn orbit transfers

Eric George
The Aerospace Corporation

ABSTRACT

"Pork Chop" contour plots of ΔV as a function of departure and arrival times, initially used for planning interplanetary trajectories, are becoming an important tool for the visualization and analysis of geocentric satellite maneuver envelopes.

The prototypical pork chop plot is based on single burn Lambert transfers for each departure-arrival time pair. However, these transfers are often very inefficient for scenarios involving a significant plane change and have little ability to satisfy additional constraints.

This paper presents a novel Genetic Algorithm approach for producing optimal multi-burn, non-coplanar trajectories for all departure-arrival time pairs in a single analysis using a distributed population with localized selection and replacement operations. The algorithm determines, for each departure-arrival time pair, both number of maneuvers and the timing, direction, and magnitude for each maneuver, with the foundational objective of minimizing the overall Delta-V.

This technique has also proven adept at minimizing Delta-V, subject to additional constraints such as lighting conditions and relative velocity. The sample scenario presented improves upon the Lambert solution by a factor of up to eight and illustrates the ability of the approach to open large areas of the solution space previously excluded by a lighting constraint.

1. INTRODUCTION

"Pork Chop" plots, illustrating contours of orbital energy versus mission departure and arrival time, have been used since the 1970s to plan the Voyager missions – the Grand Tour of the outer planets [1] as well as Mars missions [2]. They are becoming an increasingly useful tool for visualization and analysis in the geocentric context as well.

Fig. *1* illustrates a typical pork chop plot for a single burn Lambert transfer between a pair of geosynchronous objects with an inclination difference. Note the low ΔV regions, corresponding to the times when the non-maneuvering (secondary) object crosses the orbital plane of the maneuvering (primary) object.

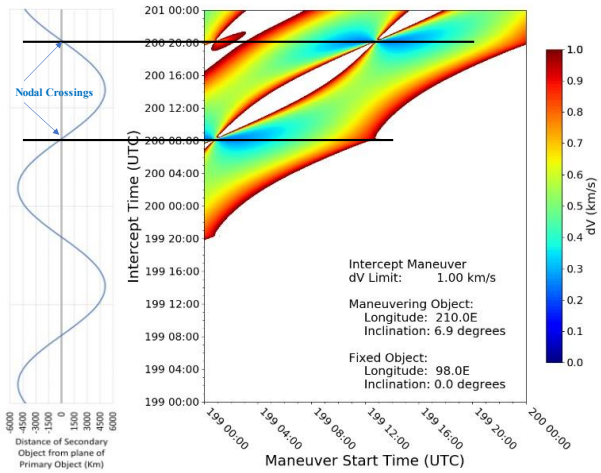

Fig. 1 Heatmap of intercept transfer between two geocentric objects. Single burn, impulsive Lambert solutions.

Fig. 2 illustrates the orbits represented by the pork chop plot. Each point in the heatmap represents an independent solution with unique departure and arrival times.

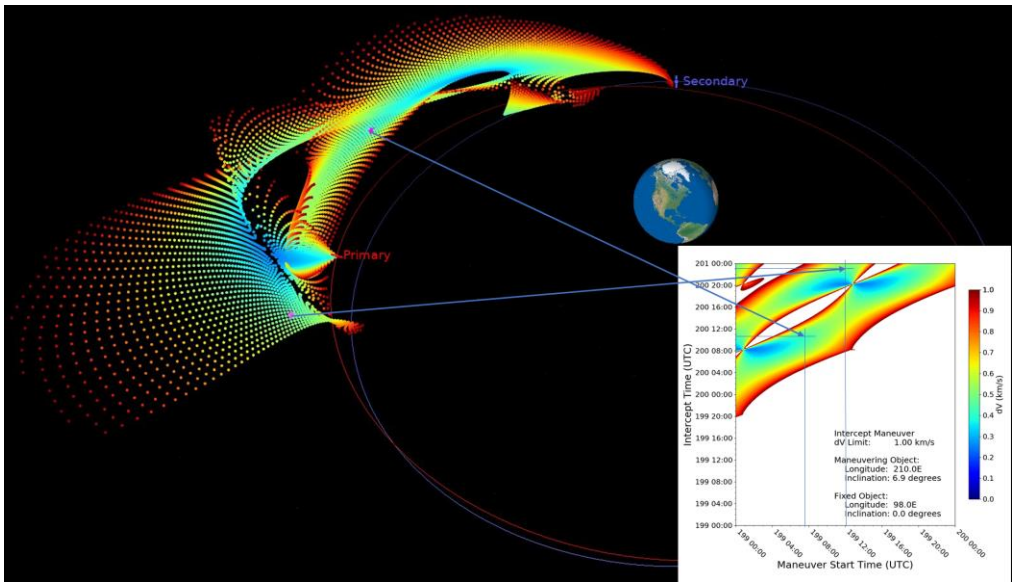

Fig. 2 Correlation of 3D orbit visualization with ΔV heatmap for single burn, impulsive Lambert solutions

These plots can be extended in several ways. One of the most interesting is visualization of a constraint on the solar phase angle (φ) as the primary object approaches the secondary object. Fig. *3* illustrates a phase angle constraint of 45º applied 1000 km from the secondary object. The grey contours indicate regions where a solution meeting the ΔV limit is possible, but which doesn't meet the solar phase angle constraint.

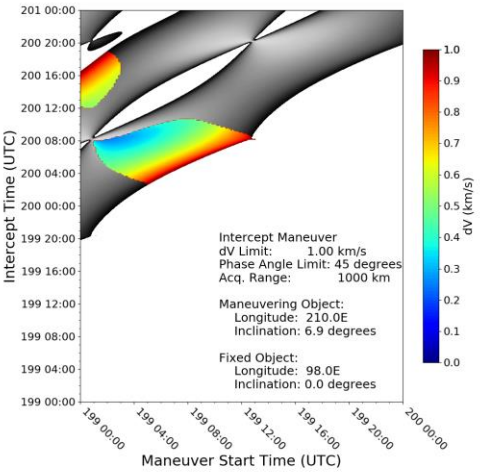

Fig. 3 Heatmap of intercept transfer between two geocentric objects, with 45º solar phase angle constraint. Single burn, impulsive Lambert solutions.

2. MULTI-BURN SOLUTIONS

While plots based on single Lambert transfers are simple and fast to compute, for non-coplanar cases they suffer from high ΔV requirements in areas away from the nodal crossing of the primary orbit (Fig. *1*). Further, these solutions can't readily be adapted to additional constraints, such as the solar phase angle constraint illustrated by Fig. *3*.

To address these issues, a Simple Genetic Algorithm (SGA) [3] was initially developed, which solved each point in the contour plot independently. Maneuvers are represented as a direction, given by the spherical coordinates Φ, Θ in a local coordinate frame, the ΔV magnitude, and a time, given as a fraction of the time remaining to the required intercept/rendezvous time. The number of maneuvers performed, out of some maximum number *N*, is part of the solution space and controlled by a single bit, ε, for each maneuver in the representation.

Table 1 Maneuver Representation

Variable	Size (bits)	Range	Description
ε	1	0, 1	On/Off toggle to determine whether this burn is expressed in the solution.
Φ	2-32	0º - 360º	ΔV direction spherical coordinate. In-plane, measured from the velocity vector, positive toward zenith.
Θ	2-32	0º - 180º	ΔV direction spherical coordinate, 0º in the direction of the angular momentum vector.
ΔV	2-32	0 - ΔV_{lim} km/s	ΔV magnitude
Δt	2-32	0.0 – 1.0	Drift time to next event, as a fraction of the time remaining until intercept/rendezvous.

Each member of the Genetic Algorithm population, termed an entity, consists of a bit vector consisting of the representation from Table *1*, repeated for each of the *N* maneuvers.

Table 2 Organization of the entity representation

Maneuver 1					**Maneuver 2**					…	**Maneuver *N***				
ε_1	Φ_1	Θ_1	ΔV_1	Δt_1	ε_2	Φ_2	Θ_2	ΔV_2	Δt_2	…	ε_N	Φ_N	Θ_N	ΔV_N	Δt_N

Upon completion of the drift Δt following the final expressed maneuver, a Lambert transfer is calculated to the targeted final intercept/rendezvous point. The total ΔV for an entity is given by Eq. 1

Eq. 1
$$\Delta V_{tot} = \sum_{i=1}^{N} \varepsilon_i \Delta V_i + \Delta V_{lambert}$$

The representation described above can readily produce invalid trajectories. The failures fall in two categories – ΔV failures where the maneuver results in a hyperbolic or re-entering trajectory, and Lambert failures where no viable Lambert transfer exists for the final maneuver. Hyperbolic and re-entering trajectories are presently excluded so that a failure to execute an intermediate maneuver doesn't result in a potentially irrecoverable trajectory.

Failure handling has two components – estimation of a maneuver magnitude which allows the evaluation of the remainder of the entity and incrementing a failure tracking variable *F* by a value proportional to the severity of the failure.

Eq. 2
$$F = \sum_{i=1}^{N} f_{dv,i} + f_{lambert}$$

ΔV failures are handled by adjusting the magnitude of the offending maneuver to the largest magnitude which doesn't cause the failure, ΔV_{adj}, found via a bisection iteration. The difference between this value and the original value is the contribution to the failure metric (Eq. 3).

Eq. 3
$$f_{dv} = \Delta V - \Delta V_{adj}$$

The ΔV contribution from failed Lambert solution is presently set to 0, while contribution to the fail metric is calculated via Eq. 4

Eq. 4
$$f_{lambert} = \frac{|\bar{P}_l - \bar{P}_t|}{T_t - T_l}$$

Where T_l, $\bar{P}_l$ are the time, position, and velocity of the primary object at the time of the required Lambert burn and T_t, $\bar{P}_t$ are the time and position of the intercept or rendezvous. This is not in any way intended to estimate ΔV required to make this transfer. The only requirement is to be proportional to how badly the Lambert transfer fails. As most of these failures are due to attempting to cross large distances in very short time periods, this generally works, though if ΔV failures are common, a scaling factor to keep $f_{lambert}$ proportional to f_{dv} is useful.

Failed entities are handled via a dynamic penalty function. Care must be taken not to drive failed solutions out of the population too early or the diversity of your population may suffer, leading to premature convergence to a local minimum. This is especially true when large fractions of the initial randomly generated population fail. The score, *S*, of an entity, which is used for selection, is calculated as

Eq. 5
$$S = \Delta V + D f_{lambert}$$

Where D is a scale factor calculated such that the score $S_f = \Delta V_f + F_f D$ of the best failed entity in a population ranks at some targeted percentile *G* of the range between the best (S_b) and worst (S_w) non-failed entities in the population.

Eq. 6
$$D = \frac{S_b + G(S_w - S_b) - \Delta V_f}{F_f}$$

The value of G starts relatively close to 1, allowing failed entities to compete with successful entities. But over the course of the run, G is gradually decreased, so that failed runs are less successful at competing with successful entities and are thus driven from the population. The rate and character of this decrease is a matter of ongoing work.

Fig. *4* shows the result from the SGA. This approach was very successful, generating solutions with ΔV levels up to eight times lower than the comparable Lambert based contour plot (Fig. *1*).

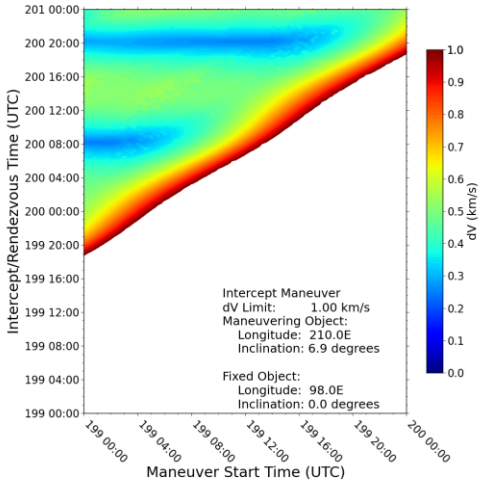

Fig. 4 Heatmap of intercept transfer between two geocentric objects. Simple Genetic Algorithm, N=3

While successful in terms of the trajectories developed, very large populations and manual tuning were required to reliably produce acceptable solutions. Scaled to the full contour plot, the computational costs were untenable – over 2,800 CPU core-hours were required to produce the data for Fig. *4*.

However, examination of the solutions from the SGA led to the insight that the solution parameters were relatively smooth and continuous across large swaths of the plot. This eventually led to the current solution strategy of a Shared Population Genetic Algorithm (SPGA).

3. SHARED POPULATION GENETIC ALGORITHM

In the SPGA, each solution point – a unique combination of departure and arrival times – has a small local genetic algorithm population P_{ij}, as shown in Fig. *5*. The size of these sub-populations is 2-3 orders of magnitude smaller than that used in the SGA. Sub-population sizes as small as 25 entities have been used successfully. The spacing between the solution points is an input parameter. Most of the figures shown in this paper used 10-minute intervals in both directions which, combined with a minimum time of flight of 16 hours, results in 17,545 sub-populations.

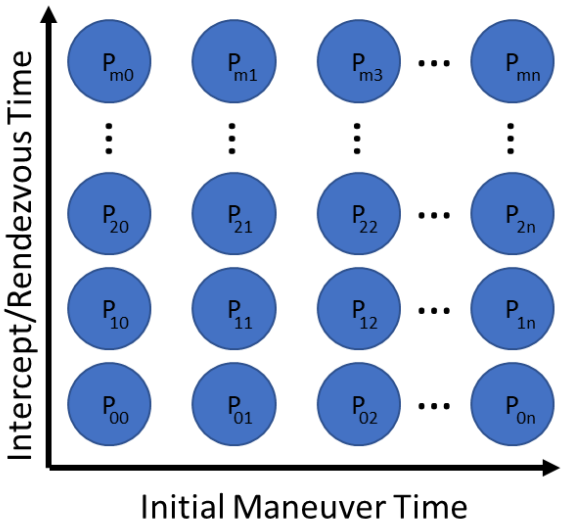

Fig. 5 SPGA Sub-Populations

There are three primary operations involved in creating new entities in a Genetic Algorithm.

1. Selection of parent entities based on solution fitness
2. Crossover re-combination of parent entities to create child entities
3. Mutation to maintain genetic diversity

In the SGA solution, each solution point, such as P_{00} in Fig. *5*, was solved independently, and therefore the selection step was simple, involving only the population for that solution point.

The algorithms described in this paper all use a Tournament selection algorithm. In Tournament Selection, a pool of entities is randomly selected from the population, and the parent entity is the most fit of that tournament pool.

The principal innovation of the SPGA is that the selection operation can span neighboring populations (Fig. *6*).

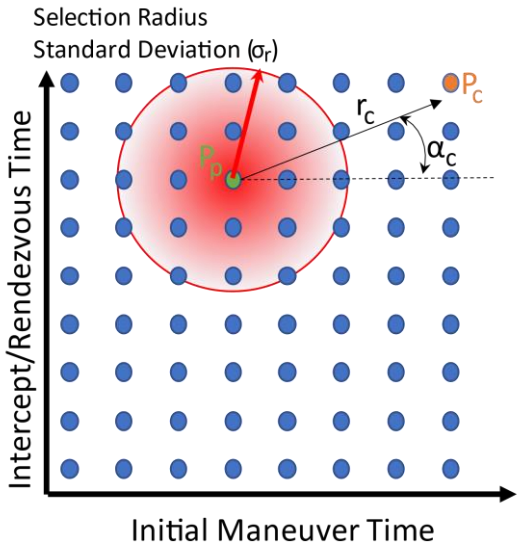

Fig. 6 SPGA Selection Operation

The selection process for the SPGA is as follows:

1. Randomly select the primary solution point, P_p (green point in Fig. *6*)
2. Determine the radius, r_c, from P_p to the selection point by means of a random draw from a normal distribution with standard deviation σ_r (input parameter).
3. Determine the direction, α_c, to the selection point by means of a uniform random draw in the range 0^o to 180^o
4. Determine the nearest sub-population, P_c (orange point in Fig. *6*), to the selection point given by the radial coordinates (r_c, α_c)
    a. The selection point may be outside of the bounds of the analysis, in which case, the process is repeated from step 2 until a valid selection point position is found.
5. Randomly select an entity for the tournament pool from contributing sub-population P_c, and evaluate it according to the maneuver start time and intercept/rendezvous time *for the primary solution point* P_p.
6. Repeat steps 2-5 until the tournament pool for each parent entity is full.

The most fit entities from each tournament pool are chosen as the parent entities, which undergo Crossover and Mutation to produce two new child entities.

The SPGA is a Steady State Genetic Algorithm [4], rather than the Generational Genetic Algorithm described by Holland [3]. As such, another pair of tournaments is conducted, with the criteria reversed (i.e. selecting for the least fit entity in the tournament), to determine the entities to be replaced by the new child entities. These tournaments, and therefore entity replacement, only occur in the Primary population P_p. While this is a steady state algorithm, the term generation is useful and here means the number of child entities produced divided by the population size.

Convergence is assessed individually for each sub-population, according to the following criteria:

1. At least 85% of the entities of the population must have no failures.
2. The best un-failed entity in the population must remain unchanged for C generations, where C is an input parameter.
3. Populations where the best entity remains unchanged for $1.5C$ are considered to be converged, despite failures.

The overall run concludes when all sub-populations have converged. Once a sub-population is marked as converged, it still participates in the algorithm as both Primary and Secondary population, but the odds of a converged population being selected as a Primary population are 5X lower than the unconverged populations in order to focus the processing resources there.

While the SPGA model bears some similarities to typical Island Model Genetic Algorithms [5], there are important differences. The primary difference is that each sub-population in the SPGA model is solving a different problem, while in Island Model algorithms, each sub-population is solving the same problem. Migration in Island Model algorithms is also generally much more restricted than the selection process described above.

4. SHARED POPULATION GENETIC ALGORITHM RESULTS

The performance of the SPGA algorithm has been exceptional. Solutions comparable to the original SGA data set can be obtained in as little as 2-4 CPU core-hours, or 2-4 minutes of run time on high end systems. The process is also very robust – failure of solution points (sub-populations) is very rare.

To assess convergence characteristics, a high quality reference data set was generated for the same geosynchronous transfer problem shown in Fig. *1* and Fig. *4* using a large population size (750 entities per sub-population) and enhanced convergence criteria, requiring the best entity in a population to remain unchanged for 12 generations. The maximum number of maneuvers, N, was four. Fig. *7* presents the reference analysis results, with the Lambert result (Fig. *1*) repeated for context as Fig. *8*. While qualitatively similar to the SGA results (Fig. *4*), note the improved smoothness and uniformity of the contours, indicating more uniform convergence with respect to neighboring sub-populations.

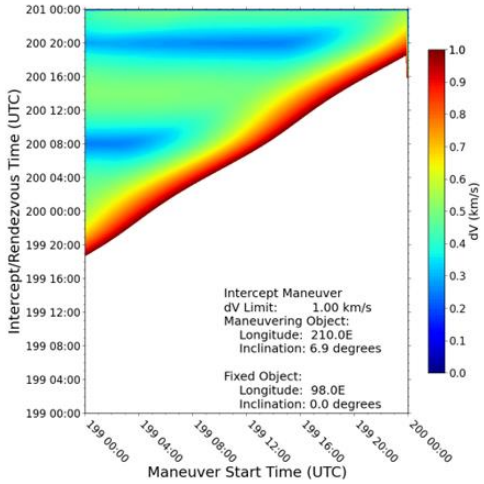

Fig. 7 Heatmap of intercept transfer between two geocentric objects, SPGA unconstrained reference analysis, N=4

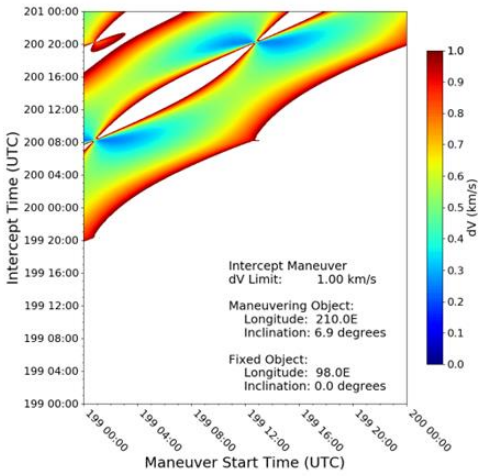

Fig. 8 Heatmap of intercept transfer between two geocentric objects. Single burn, impulsive Lambert solutions

Relative to the Lambert solution (Fig. *8*), the Genetic Algorithm solution increases the size of the accessible solution space by 50% and matches or exceeds the performance of the Lambert solver for all of the common solution points. The GA solution outperforms the Lambert solution by an average of 138 m/s (20.8%) and a max of 705 m/s (72.1%) for solution points where both techniques have valid solutions.

To assess the impact of sub-population size on run time and solution quality, five additional analysis runs were performed with sub-populations sizes of 25, 50, 100, 200, and 400 entities, but otherwise identical parameters. The convergence criteria for these runs were set to eight generations.

Fig. *9* shows the average difference in ΔV across all sub-populations for varying sub-population size. Sub-population sizes as low as 50 entities will converge to within 1% of the reference data set. Fig. *10* shows the same data, but with respect to the analysis run time, rather than generation, on a system with 64 2.1GHz CPU cores.

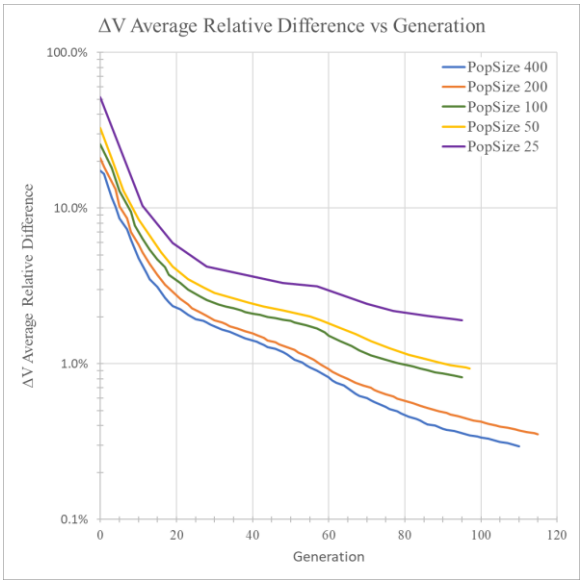

Fig. 9 Convergence vs Generation varying sub-population size, assessed against the unconstrained SPGA reference data set

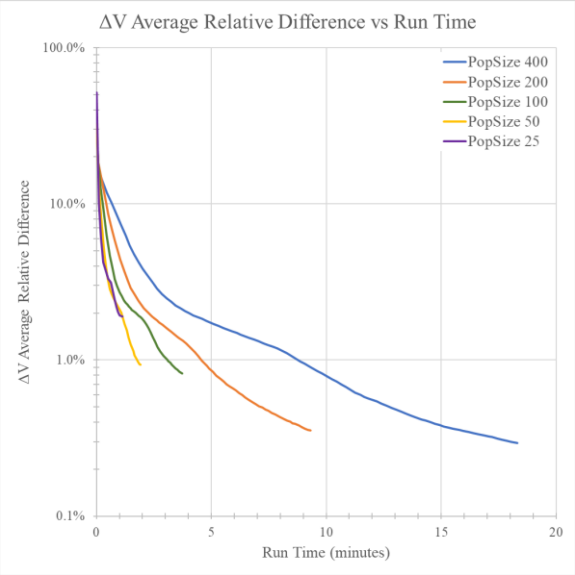

Fig. 10 Convergence vs Run Time varying sub-population size, assessed against the unconstrained SPGA reference data set

Fig. *11* looks at the results in more detail for the analysis with a sub-population size of 200 by considering the minimum, maximum and average differences in the final solutions across the 17,545 sub-populations. Note that some sub-populations generated better solutions than the reference data set. The spikes in the Max Relative Distance data are presently unexplained pending improved instrumentation and data collection of the program.

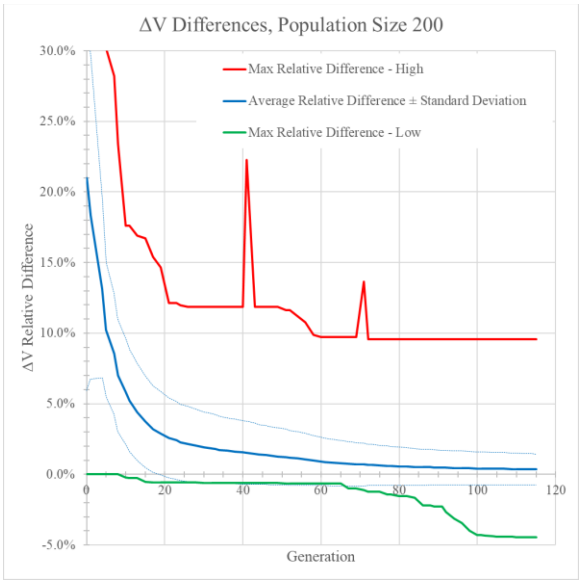

Fig. 11 Convergence details, sub-population size 200, with respect to the unconstrained SPGA reference data set.

The following figures explore the nature of the solutions, using the unconstrained SPGA reference data set.

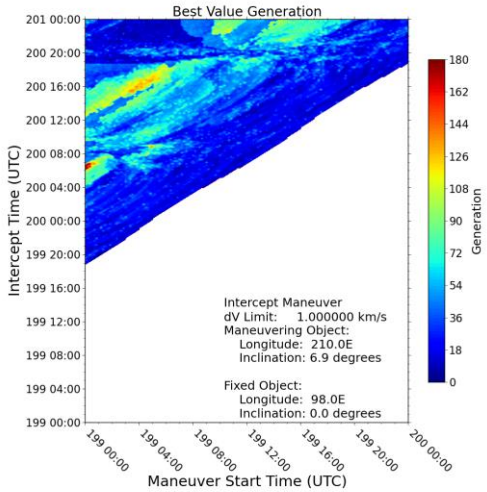

Fig. 12 Sub-population generation at convergence, unconstrained SPGA reference data set

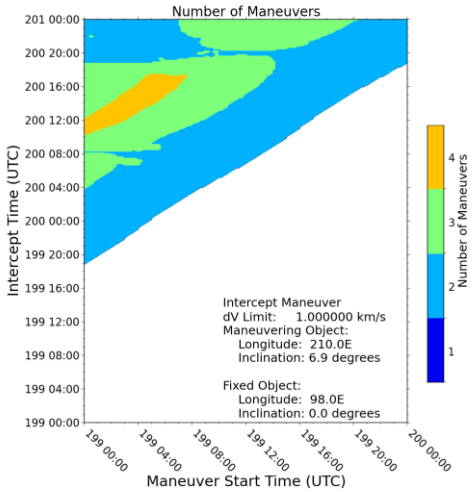

Fig. 13 Number of maneuvers, unconstrained SPGA reference data set

Note that much of the solution space converges very rapidly (Fig. *12*), while only a small fraction of the solution points are still improving late in the analysis. This suggests that there might be an opportunity to improve performance by varying the sub-population size according to convergence rate.

Recall that the algorithm also selects the number of maneuvers. Fig. *13* shows the total number of maneuvers used across the solution space. Although there is no direct selection pressure for fewer maneuvers, the algorithm does favor this. Areas where convergence is prolonged correspond to regions where more maneuvers were used.

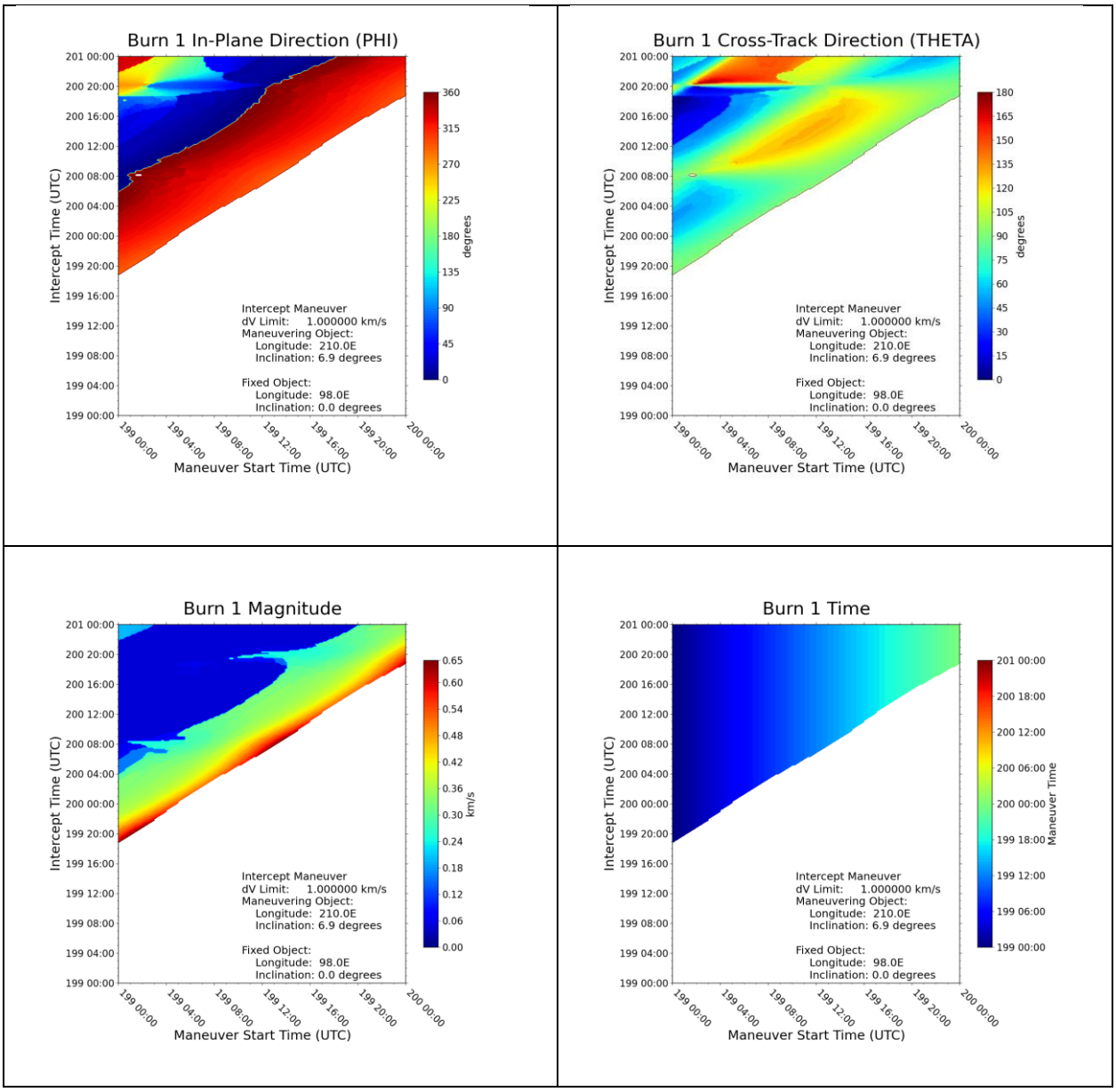

Fig. 14 Maneuver 1, unconstrained reference data set, N=4

Note that there is a minimum ΔV of 50 m/s for the first maneuver. This is to ensure that the maneuvering vehicle departs the starting position at the specified time. Without this minimum, the algorithm would often set the ΔV for the initial maneuver to 0, waiting for a more advantageous starting time. The lower left plot in Fig. *14* shows that a significant fraction of the initial burns at or near this lower limit.

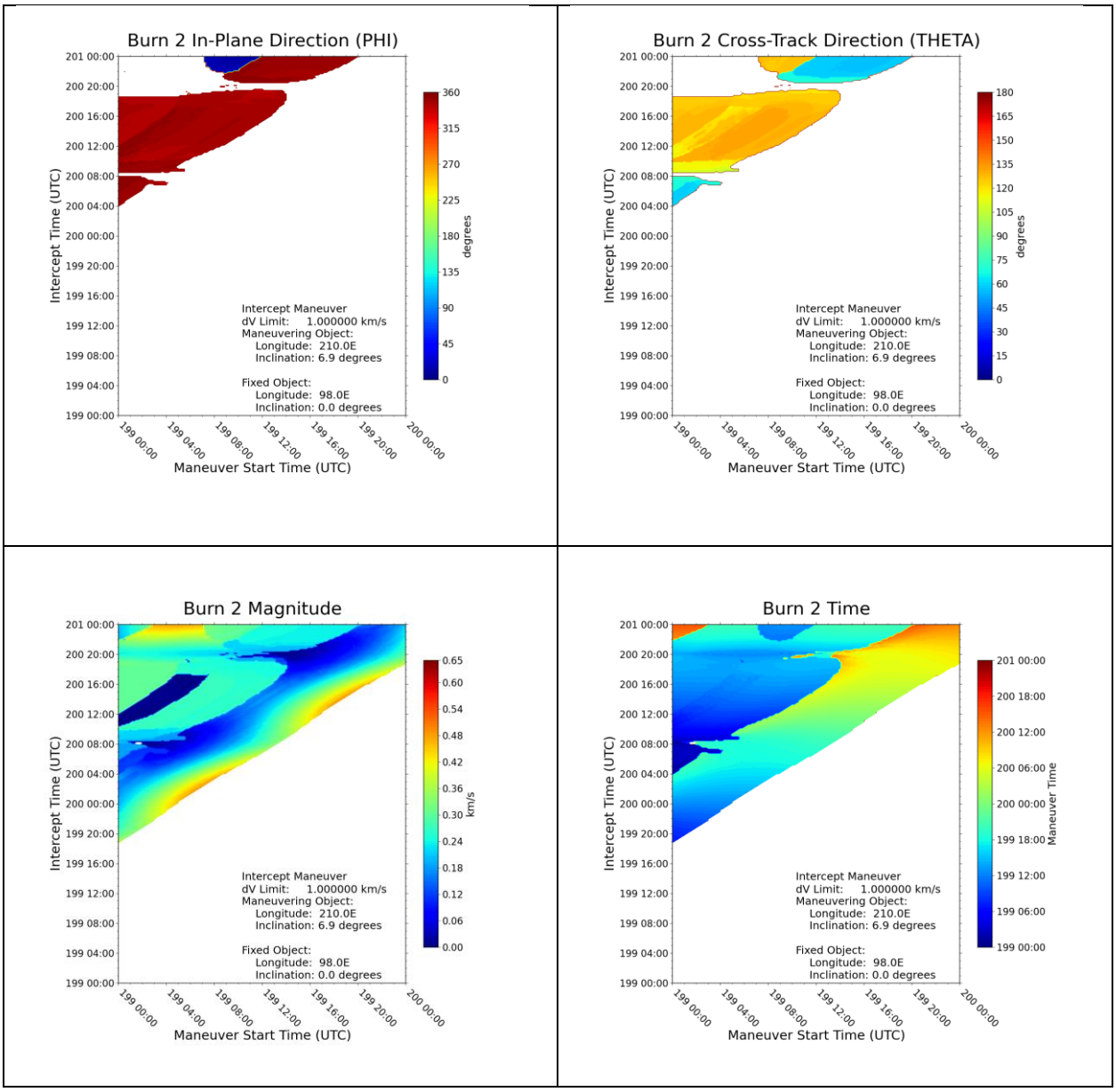

Fig. 15 Maneuver 2, unconstrained reference data set, N=4

Note that the ΔV magnitude sub-plot (lower left) in Fig. *15* includes both the prescribed maneuvers for areas of the solution space which use more than two maneuvers (areas shown in the Φ and Θ sub-plots) as well as the ΔV magnitude of the final Lambert solution for areas of the solution space which only use two maneuvers.

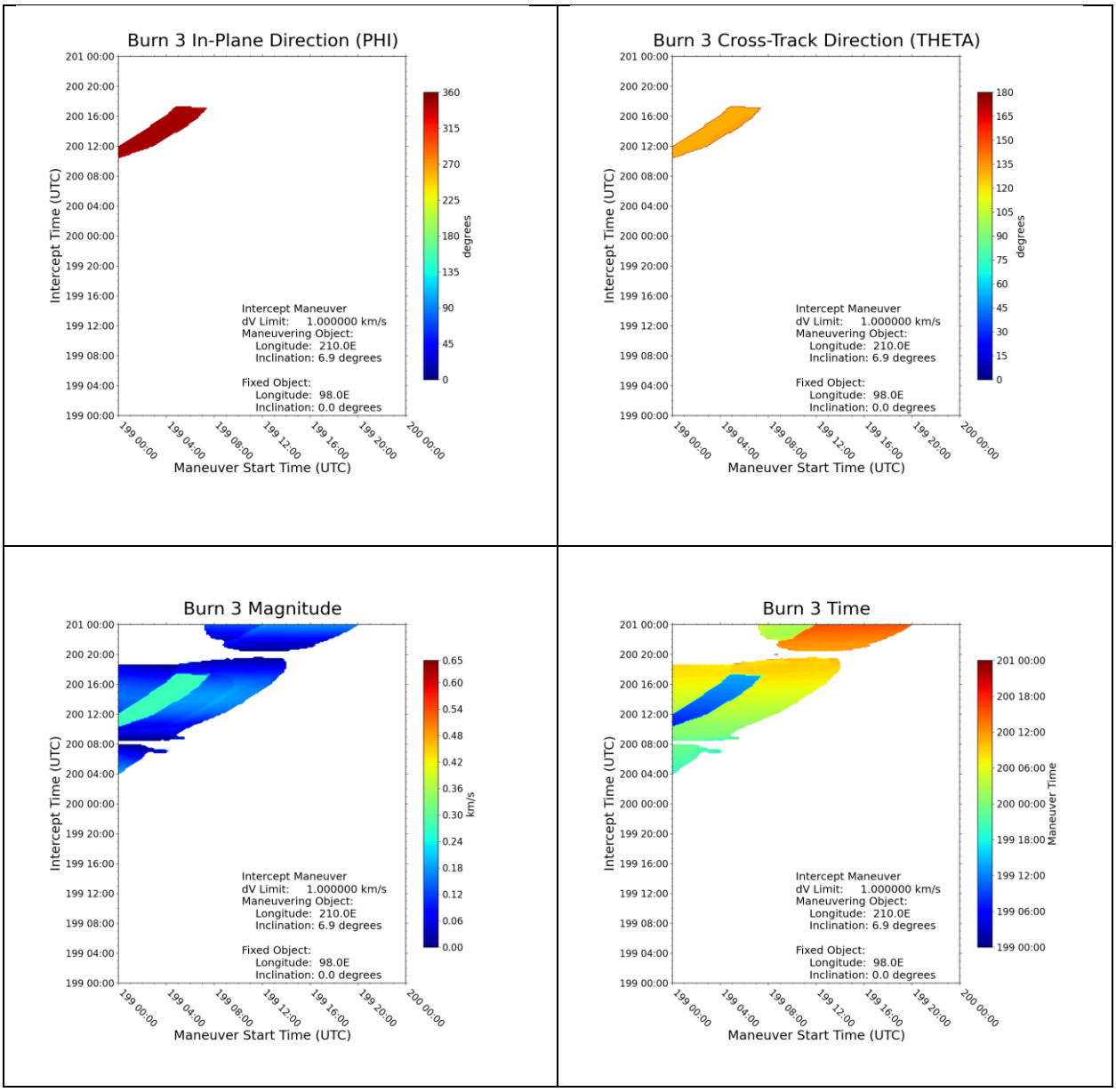

Fig. 16 Maneuver 3, unconstrained reference data set, N=4

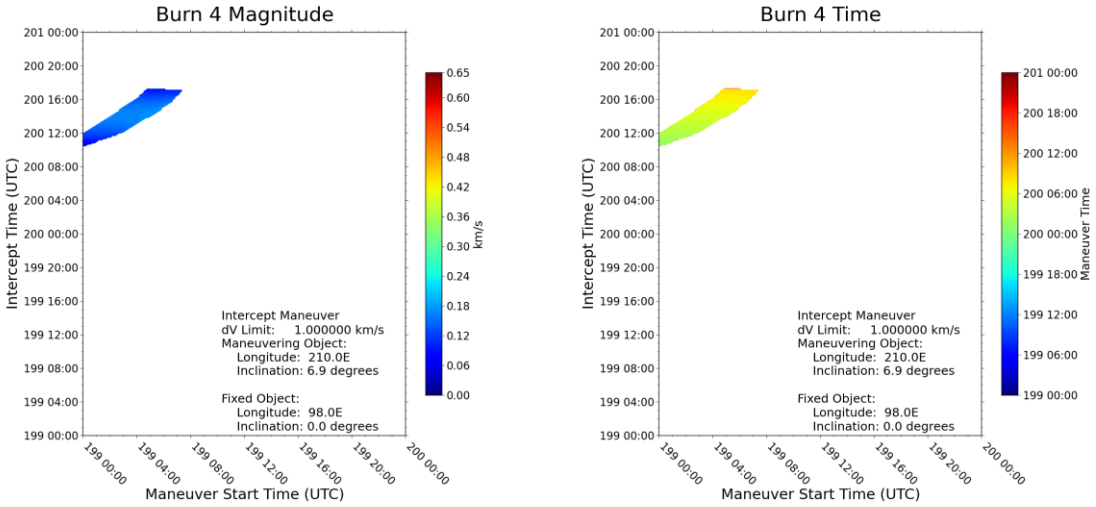

Fig. 17 Maneuver 4 magnitude and time, unconstrained reference data set, N=4

Fig. *17* represents the ΔV magnitude for the final Lambert solution for sub-populations using 4 maneuvers. The Φ and Θ sub-plots represent the ΔV direction only for maneuvers developed by the genetic algorithm and are therefore not present for the final Lambert maneuver.

The ΔV magnitude sub-plots (lower left) of Figures Fig. *14* through Fig. *16* and Fig. *17* indicate that while up to four maneuvers were chosen by the algorithm, many of those maneuvers are very small. Additional analysis runs were performed with the maneuver limits of two and three. The results (Figures Fig. *18*, Fig. *19*) indicate that allowing the fourth maneuver does have a small impact on the solution quality, but at a considerable cost in convergence and run time.

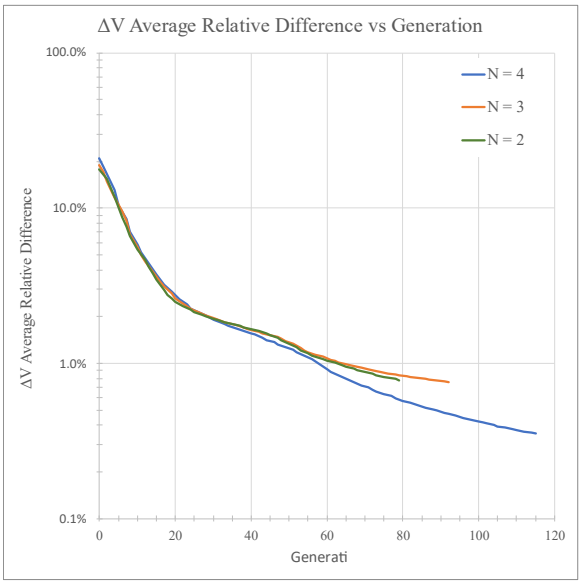

Fig. 18 Convergence vs Generation varying maneuver limit, assessed against the unconstrained SPGA reference data set

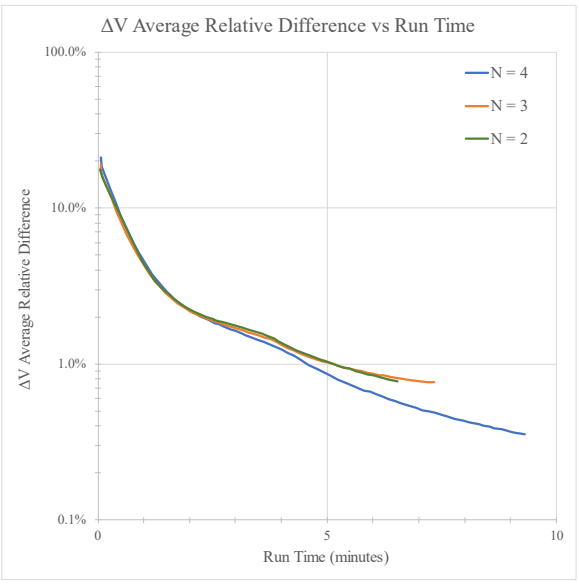

Fig. 19 Convergence vs Run Time varying maneuver limit, assessed against the unconstrained SPGA reference data set

5. Solar Phase Angle Constraint

The solar phase angle constraint (φ_{max}) is implemented by the addition of another term to the score equation (Eq. 5)

Eq. 7
$$S = \Delta V + D f_{lambert} + E f_{phi}$$

The φ failure value f_{phi} is given by Eq. 8 where the value of φ is assessed using the final trajectory at the specified range before intercept/rendezvous with the secondary object.

Eq. 8
$$f_{phi} = \begin{cases} 0 & \varphi \leq \varphi_{max} \\ \varphi - \varphi_{max} & \varphi > \varphi_{max} \end{cases}$$

The scaling value *E* is presently set to an initial value of 0.02 km/s, increasing at a rate of 0.75% every ¼ generation.

A reference data set (Fig. *20*) was created for φ_{max} = 45° for the same geosynchronous scenario used in the previous section and analogous to the Lambert solution shown in Fig. *3* (repeated as Fig. *21* for ease of comparison).

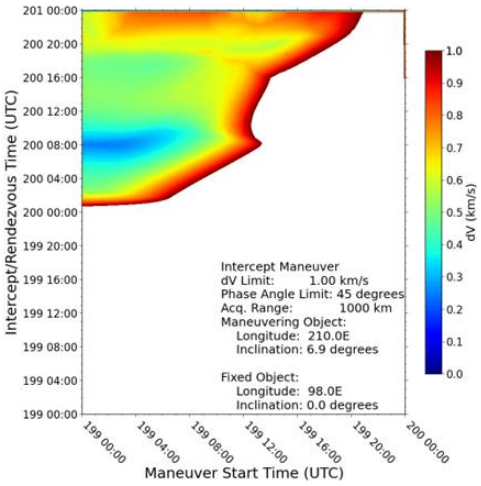

Fig. 20 Heatmap of intercept transfer between two geocentric objects, constrained SPGA reference analysis, φ_{max} = 45°, N=4

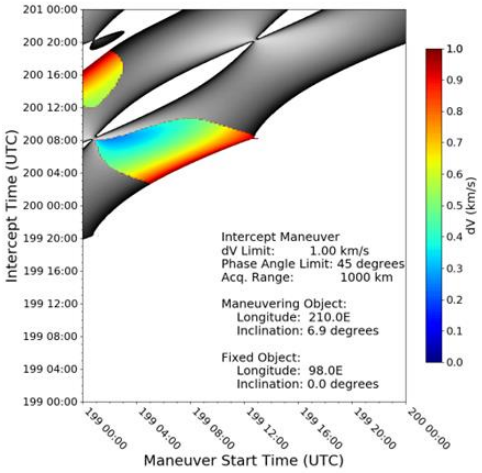

Fig. 21 Heatmap of intercept transfer between two geocentric objects. Single burn, impulsive Lambert solutions, φ_{max} = 45°

Relative to the Lambert solution (Fig. *21*), the Genetic Algorithm solution increases the size of the accessible solution space by nearly a factor of seven and matches or exceeds the performance of the Lambert solver for 99.92% of the common solution points. The GA solution outperforms the Lambert solution by an average of 100 m/s (15%) and a max of 507 m/s (51%) for solution points where both techniques have valid solutions.

A solar phase angle constraint of 45° is very challenging, as evidenced by the regions of failed sub-populations in the contour plots shown in Fig. *22* for reduced population sizes.

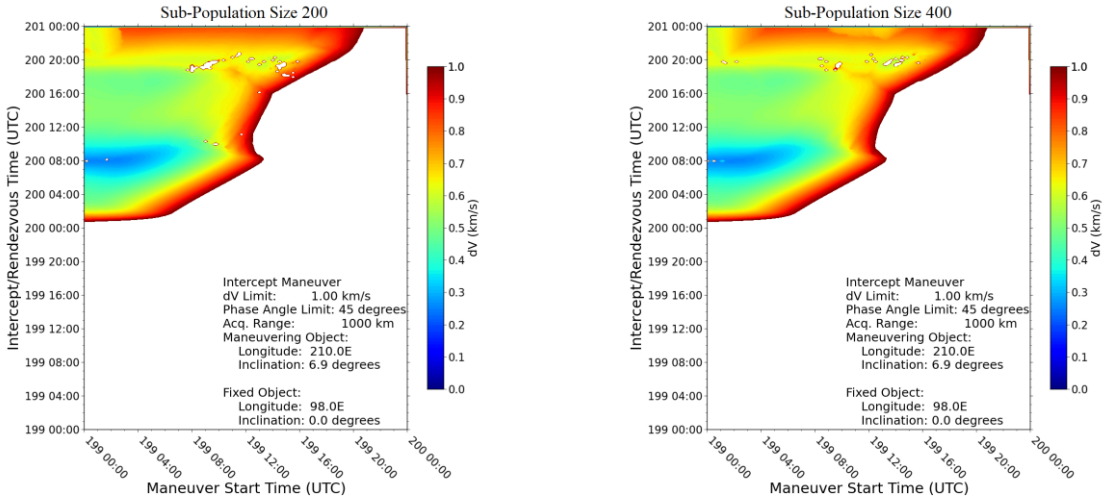

Fig. 22 Heatmaps for sub-populations of size 200 and 400, φ_{max} = 45°, N=4

Fig. *23* summarizes the percent of failed sub-populations with respect to the reference data set. The problem can be quite significant for the smallest populations.

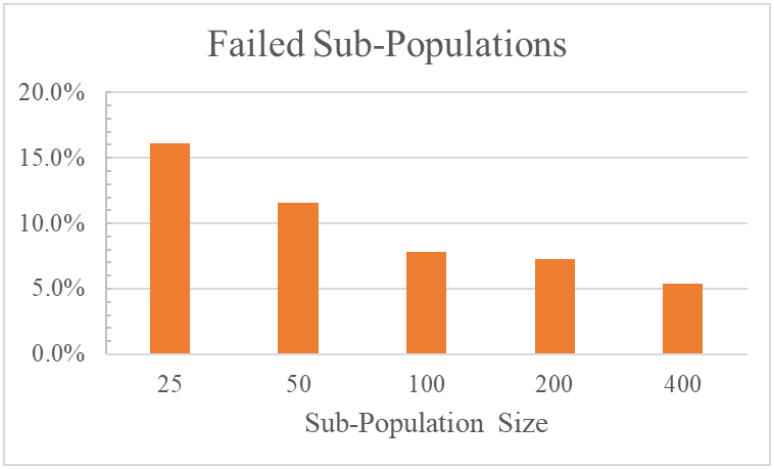

Fig. 23 Fraction of failed solution points with respect to constrained reference data set, φ_{max} = 45°, N=4

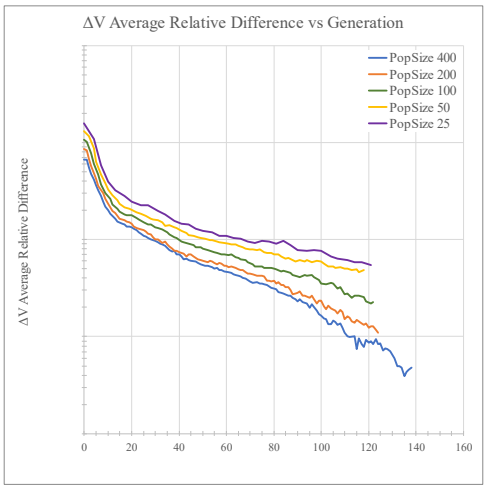

Fig. 24 Convergence vs Generation varying sub-population size, assessed against constrained SPGA reference data set, φ_{max} = 45°, N=4

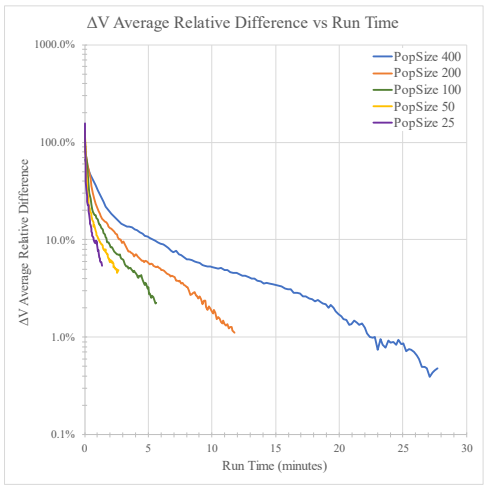

Fig. 25 Convergence vs Run Time varying sub-population size, assessed against constrained SPGA reference data set, φ_{max} = 45°, N=4

Note that the fourth maneuver was rarely used in the constrained problem (Fig. *27*) compared to the unconstrained problem (Fig. *13*). Initial investigations did not show any significant advantage in convergence or run time in restricting the problem to 3 maneuvers.

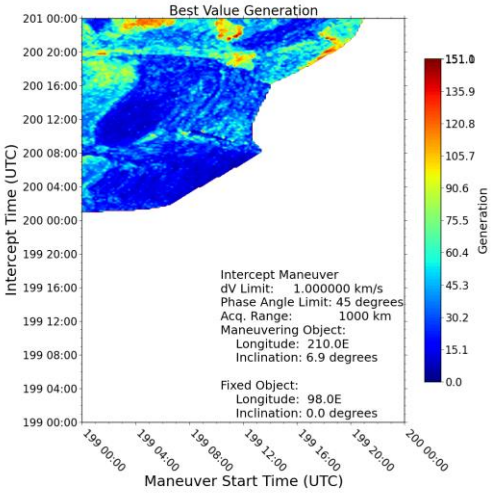

Fig. 26 Sub-population generation at convergence, constrained SPGA reference data set, φ_{max} = 45°, N=4

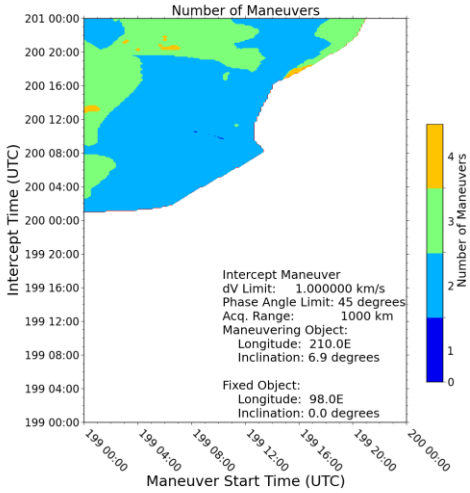

Fig. 27 Number of maneuvers, constrained SPGA reference data set, φ_{max} = 45°, N=4

Fig. *28* describes the initial maneuver for the constrained problem using the reference data set. The plots show significantly more complexity with respect to the unconstrained plots (Fig. *14*), including a switch from prograde to retrograde burn direction in the upper-right of the in-plane burn direction plot (upper-left sub-plot). This corresponds to the sub-GEO trajectories in Fig. 30, which illustrates the trajectories developed in a 3D orbit visualization.

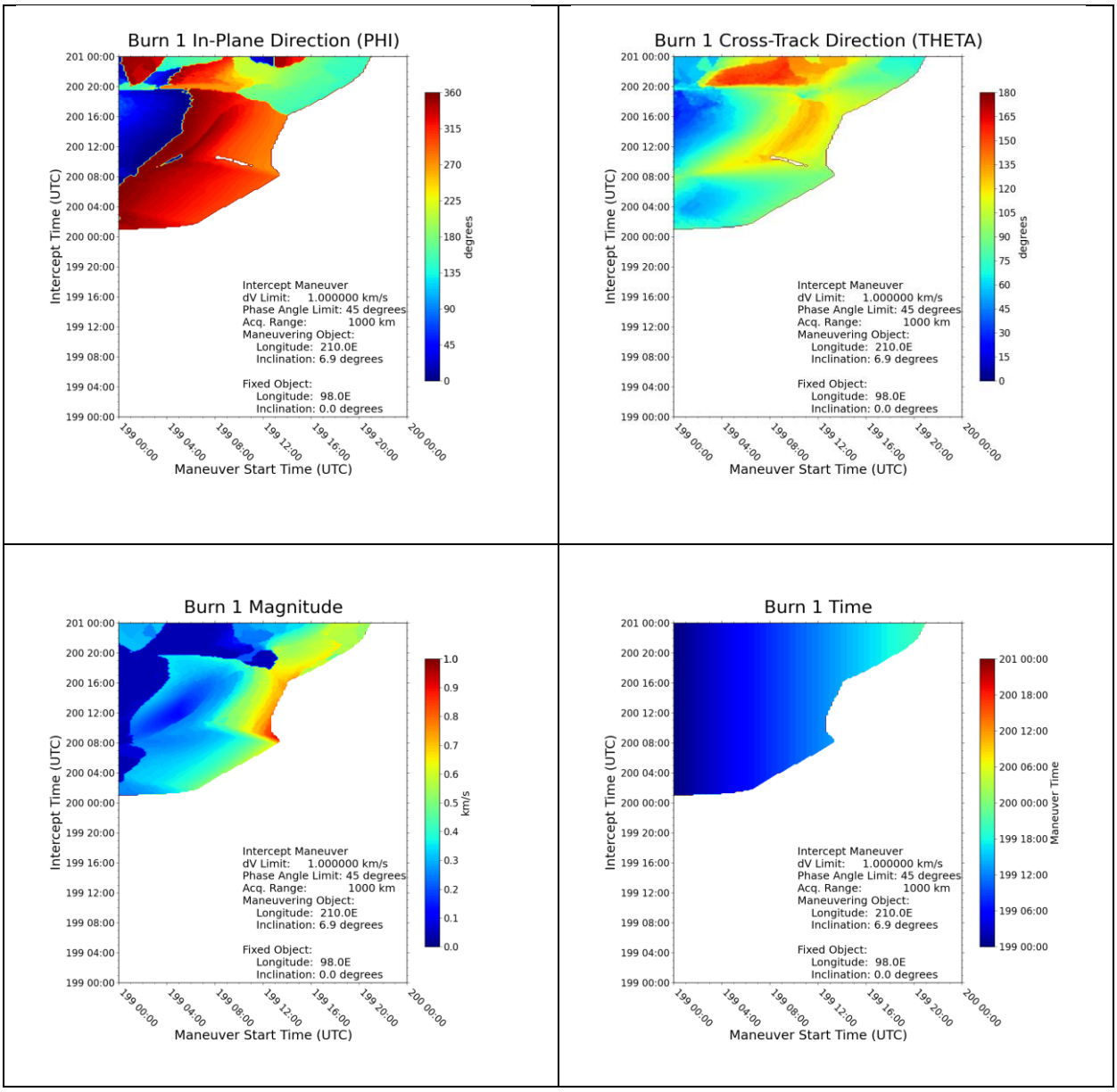

Fig. 28 Maneuver 1, constrained SPGA reference data set, φ_{max} = 45°, N=4

Fig. 29 continues the trend of increased complexity, highlighting the often sharp divisions between adjacent solutions.

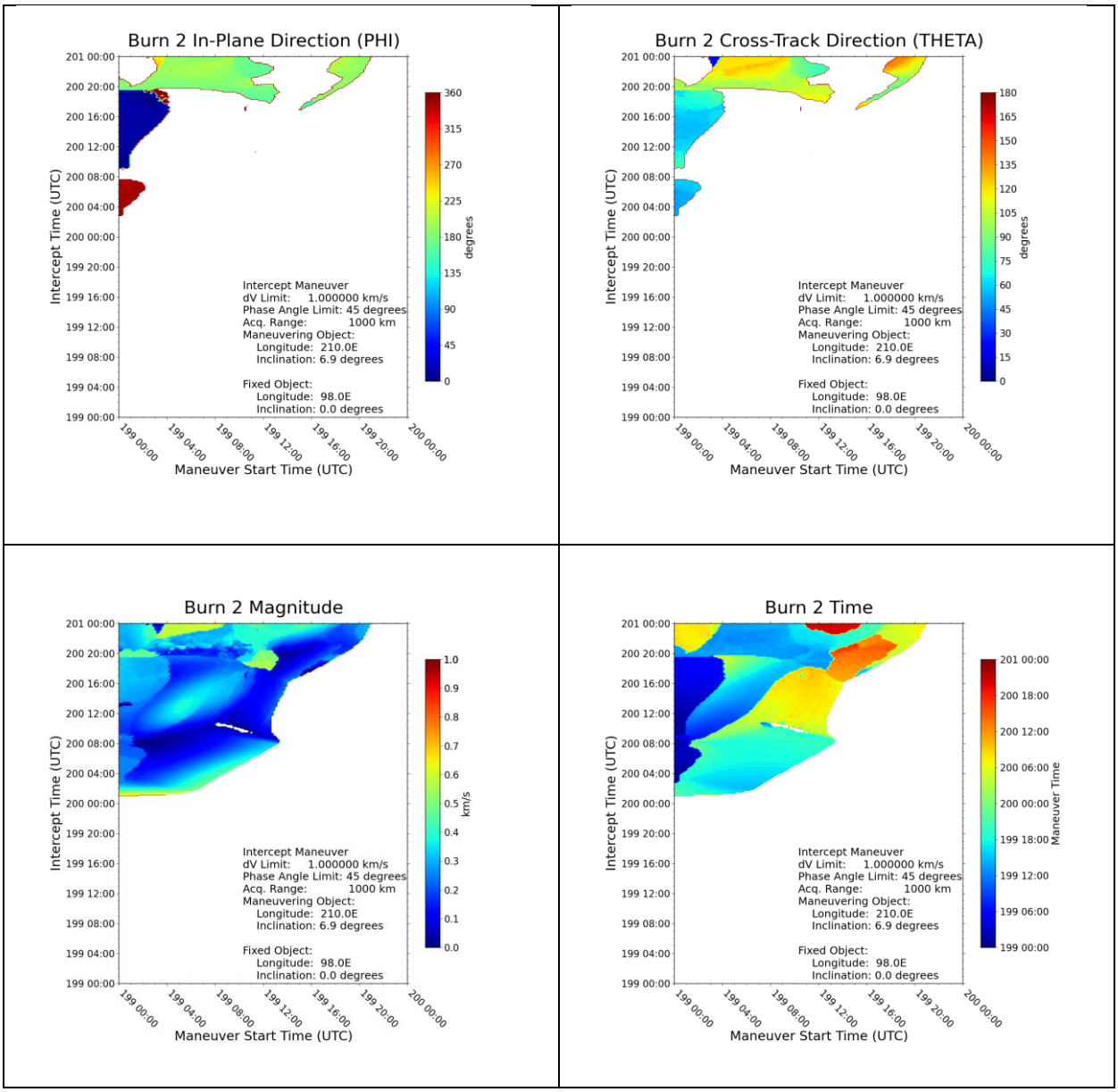

Fig. 29 Maneuver 2, constrained SPGA reference data set, φ_{max} = 45°, N=4

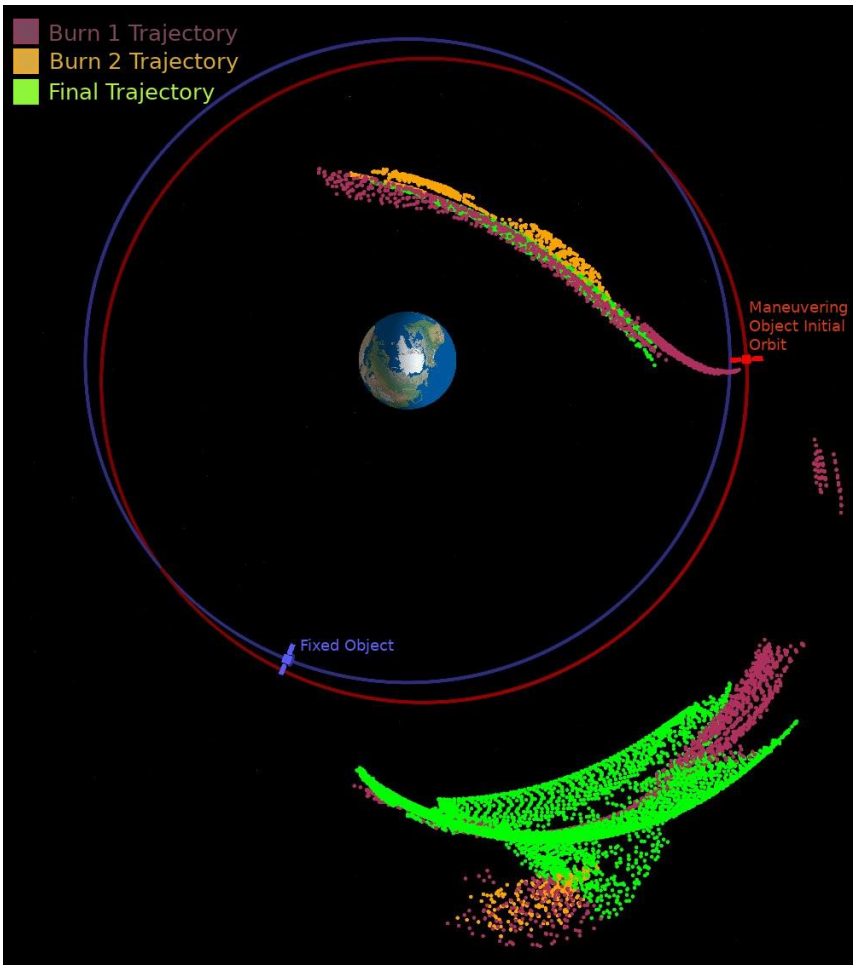

Fig. 30 Orbit visualization for constrained reference data set

6. Conclusions

The Shared Population Genetic Algorithm has demonstrated the ability to generate large quantities of multi-burn trajectories for constrained and unconstrained orbital intercept and rendezvous problems with reasonable computational effort. Ongoing work includes refinements to the selection radius and sub-population size according to the local convergence environment and well as migration to GPU processing to facilitate complex constraints and more realistic maneuver models, including finite burn modelling.

Acknowledgements

The author thanks D. Bass, M. Britton and G. Vazquez of The Aerospace Corporation for technical and editorial review of this paper. This work was performed as internal research and development by The Aerospace Corporation in support of the Space Warfighter Initiative.

[1] Westwick, Peter J. (2007), *Into the Black: JPL and the American Space Program, 1976-2004*. Yale University Press. ISBN 978-0-300-11075-3, Part 2, pg 20

[2] *"Porkchop" is the First Menu Item on a Trip to Mars*, (n.d.), NASA's Mars Exploration Program, https://mars.nasa.gov/spotlight/porkchopAll.html

[3] Holland, John H. (1992), *Adaptation in Natural and Artificial Systems*. MIT Press. ISBN 0-262-58111-6

[4] Agapie, Alexandru and Wright, Alden H., "Theoretical Analysis of Steady State Genetic Algorithms" (2014). Computer Science Faculty Publications. 28., https://scholarworks.umt.edu/cs_pubs/28

[5] Whitley, Darrell & Rana, Soraya & Heckendorn, Robert. (1998). The Island Model Genetic Algorithm: On Separability, Population Size and Convergence. Journal of Computing and Information Technology. 7.