THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF AEROSPACE ENGINEERING

Performance Analysis of DE-Sigma Optimization Method

JOHN QUINN
SPRING 2021

A thesis
submitted in partial fulfillment
of the requirements
for a degree
in Aerospace Engineering
with honors in Aerospace Engineering

Reviewed and approved* by the following:

Dr. Robert G. Melton
Professor of Aerospace Engineering
Thesis Supervisor

Dr. Mark Maughmer
Professor of Aerospace Engineering
Honors Adviser

* Electronic approvals are on file.

ABSTRACT

The modified heuristic method Differential Evolution Sigma (DE-Σ) could be a powerful tool for mission optimization but requires more research in algorithm parameter permutations and problem applications. This paper will analyze DE-Sigma's performance in two problems: Finding a Lyapunov orbit in the Earth Moon System (Lyapunov Problem) and finite thrust between coplanar circular orbits (FTX Problem). Within these problems, the parameters of the algorithm (weighting function exponent 'm', trial population fraction 'phi', and crossover value 'CR') are varied to see how the final solution is affected. Particle Swarm Optimization (PSO) is used as a control value for comparison to DE-Sigma. Performance is measured by successful convergence.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I'd like to thank my family for supporting my academics. Without them, I wouldn't be here. I would like to thank Dr. Robert G. Melton for his knowledge and patience on this topic. Without his guidance, none of this research would be possible. Finally, I'd like to thank Sarah Johns. Thank you for being my editor and cheerleader. Your support carried me through every step of this paper.

Chapter 1

Introduction

Optimal trajectories for space missions can be determined analytically or numerically. Some programs make use of heuristic methods to find the optimal trajectory for a given problem. These programs operate by creating a search space out of unknown parameters. An objective function is constructed to take these random parameters and introduces penalty functions relevant to the problem. Some algorithms model natural phenomena like the movement of swarms of bugs or birds in Particle Swarm Optimization (PSO).

Particle Swarm Optimization

The PSO method creates particles that act as members of the swarm. Each particle is an individual solution that contains elements which are values of unknown parameters to be entered into the equations of motion of a problem. These elements are initial values for integration to produce parameter values at the end of the trajectory. A particle has a position in the search space from its parameter elements but at the end of on iteration a velocity updates the parameters to move the swarm particle to a new place in the solution space. The final values of the unknown parameters from integration are inserted into an objective function.

This objective function incentivizes the swarm to find the absolute minimum through the penalty functions. If unknown parameter values at the end are outside of acceptable ranges as defined by a given problem (i.e. A particle's trajectory takes too long or is too far from a target destination) a penalty is applied to increase the value returned by the objective function. Each particle is evaluated by the objective function to 'inform' the other members of the swarm. The global best position, that is the

position with the smallest objective function value returned, from all previous iterations is saved in addition to the best position visited in the current iteration. The velocity of a particle is a combination of its own random inertia as well as movement towards the recent best and global best position. Over iterative cycles, the swarm eventually finds a minimum of the objective function. Failings of PSO are that the minimum may not be absolute but local and iterative stagnation. This phenomenon is called false convergence. False convergence has reportedly led to a 40% failure by the PSO algorithm to acquire a valid solution to a given problem [1].

DE Algorithm

The original Differential Evolution algorithm models the natural phenomena of evolution. Particles made of elements that are unknown parameter values are used like in PSO. The same integration method is used along with an objective function for fitness evaluation. Instead of sharing intelligence like a smart swarm, the particles are compared with a differential particle. This differential particle is the result of "particle evolution". Three random particles are taken from the population. These are the parent particles. Elements of the particles are added and multiplied by tuning parameters. This combination and multiplication results in a child particle and the process is called the crossover process. The child particle's fitness is compared with a member of the original population of particles. If the child particle has elements that fall out of the search space, those out of bounds elements are set to the bounding values. If the fitness of the child is greater than the parent function via the objective function, the child replaces the parent. If not, the child particle is discarded and iteration continues.

This method runs into the problem of too quickly contracting the search space. Because the particles do not behave like a smart swarm with individual minimums being communicated, all particles converge towards a minimum. This minimum may not be a global minimum.

DE-Σ Algorithm

Modified Differential Evolution Algorithm (DE-Σ) attempts to avoid the problem of false convergence using multiple populations and a differential sum particle. Rather than modeling a "smart swarm" like in PSO, this method uses a more mathematical approach. Particles are still employed that have elements made up of unknown parameter values. Integration still takes place as well as the insertion of the final values of integration into an objective function for evaluation. Position and velocity are not employed. DE-Σ divides the population of particles into two separate populations. This is the modification to the original Differential Evolution Algorithm. Particle elements are updated by comparing the objective function evaluation of particles to a differential particle, the so-called offspring particle. This differential particle is generated by a combination of semi-random values and three other parent particles randomly selected from the population. If the differential particle returned a smaller value from the objective function, the original particle was replaced by the differential particle. This is the titular evolution. DE-Σ splits the entire population into two smaller subsets. This prevents all of the particles converging towards one minimum like Particle Swarm Optimization. The solution space is kept from contracting too quickly, and encourages the possibility of the algorithm finding an absolute minimum [2].

While this method does address the problem of false convergence found in Particle Swarm Optimization, the method still needs refining. Parameters in the algorithm could be altered to improve accuracy, success rate, or run-time. These parameters are the weighting exponent m, the crossover value CR, and the population fraction ϕ. The weighting exponent governs how the weighting factor, one of the tuning parameters involved in DE- Σ, slowly approaches 0 from 1 as the iterations proceed. The effect this has is that the differential particle is smaller over iterative cycles, allowing the algorithm to home in on minimums gradually rather than overshooting. The crossover value is subtracted from a value that is randomly selected from a uniform distribution of 0 to 1. As the crossover value increases, the differential particle is more likely to be rejected by the algorithm. This process is detailed further in the analysis of the actual algorithm later in this paper. Finally, the population fraction determines the size of the two

subsets of particles. The division is not equal but both subsets added together would equal the original population. As ϕ increases or decreases from 0.5, the subsets become more unequal. Perhaps there are optimal subsets that are not equal. Maybe there is some use to having a larger subset and a small subset interact with the search space.

Chapter 2

Problem Introduction

The following problems were used to test what combination of the variable parameters worked best in the DE-Σ algorithm with PSO acting as a control. These problems closely follow the ones in Ref 1.

Lyapunov Problem

This problem involves finding an initial position and period such that the third body of the circular restricted three-body problem enters a stable orbit around the interior collinear Lagrange point of the Earth-Moon system.

Beginning with problem assumptions, the circular restricted three body problem assumptions can be carried over. The third body is assumed to be a satellite with negligent mass compared to the Earth and Moon. The larger bodies follow a counterclockwise circular orbit around the system's center of mass. Canonical units are also employed. The distance between the Moon and Earth is constant and 1 distance unit (1 DU = 384,400 km), and a time unit is such that the larger bodies both complete an orbit in 2π time units (TU = 375190 s). By these definitions, $\mu_{earth} + \mu_{moon} = 1\ DU^3/TU^2$. A gravitational parameter is introduced:

$$\mu \triangleq \frac{\mu_{moon}}{(\mu_{moon} + \mu_{Earth})} = 0.01215510\ \frac{DU^3}{TU^2} \qquad (1)$$

The coordinate frame involving the Earth, Moon, and satellite are given by the following synodic reference frame. Velocity, Radius and Gamma are all defined within it:

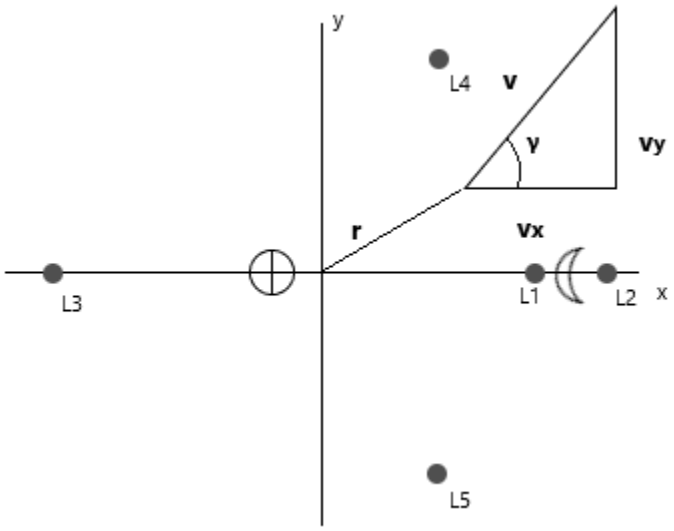

Figure 1. Synodic Reference Frame

The equations of motion can start being defined:

$$\dot{x} = v_x \tag{2}$$

$$\dot{y} = v_y \tag{3}$$

$$\dot{v_x} = \frac{\delta\Omega}{\delta x} + 2v_y \tag{4}$$

$$\dot{v_y} = \frac{\delta\Omega}{\delta y} - 2v_x \tag{5}$$

where

$$\Omega = \frac{x^2+y^2}{2} + \frac{1-\mu}{\sqrt{(x+\mu)^2+y^2}} + \frac{\mu}{\sqrt{(x+\mu+1)^2+y^2}} \tag{6}$$

The Jacobi integral is used for this dynamical system, whose value is the Jacobi Constant, C:

$$C = 2\Omega - (v_x^2 + v_y^2) \tag{7}$$

The Jacobi Constant represents the energy of the system and remains constant. Its value determines the geometric shape of zero-velocity curves, or where the positions where the velocity of the satellite is 0. These curves constitute the boundary of the third body's motion. After introducing the Jacobi Constant, the angle formed by the velocity vector and the x axis of the system is defined as γ. The velocity components can be rewritten as:

$$v_x = \sqrt{2\Omega - C}\cos\gamma \tag{7}$$

$$v_y = \sqrt{2\Omega - C}\sin\gamma \tag{8}$$

$$\tan\gamma = \frac{v_y}{v_x} \tag{9}$$

The time derivation of γ is:

$$\frac{\dot{\gamma}}{\cos^2\gamma} = \frac{\dot{v_y}v_x - \dot{v_x}v_y}{v_x^2} \qquad \dot{\gamma} = \frac{\Omega_y\cos\gamma - \Omega_x\cos\gamma}{\sqrt{2\Omega - C}} \tag{10}$$

where Ω_x and Ω_y is the partial derivative of Ω with respect to x and y respectively. Equations (2), (3), and (10) are the state equations of the problem's system. Returning to the objective of the problem, an orbit is considered stable if at the end of the period, T, the three unknown parameters {x, y, γ} should match their respective initial values:

$$x(T) = x(0) \quad y(T) = y(0) \quad \gamma(T) = \gamma(0) + 2\rho\pi \quad (\rho \in \mathbf{Z}) \tag{11}$$

These conditions are easily translated into an objective function for optimization:

$$J = |x(T) - x(0)| + |y(T) - y(0)| + \min\{mod[|\gamma(T) - \gamma(0)|, 2\pi], mod[-|\gamma(T) - \gamma(0)|, 2\pi]\} \tag{12}$$

The last term of J is a piecewise function of γ that has a maximum value of π and a minimum value of 0. The function should be equal to 0, provided a least one periodic orbit exists. The problem now consists of trying to find initial conditions for all the unknown parameters, however, the problem can offer further simplifications. The Lyapunov orbit would be a clockwise trajectory about the L_1 point. L_1 has coordinates (0.839863,0) in canonical units. The orbit would be symmetrical about the x-axis. The problem could be further limited so that if x(0) is less than the distance to L_1, y(0) can be assumed to be zero since L_1 lies on the x-axis. Therefore:

$$x_{min} \le x(0) \le x_{L1} \quad y(0) = 0 \quad \gamma(0) = \frac{\pi}{2} \tag{13}$$

where x_{min} is set to 0.75 DU. This further reduces the objective function to:

$$J = |x(T) - x(0)| + |y(T)| + \min\{mod\left[\left|\gamma(T) - \frac{\pi}{2}\right|, 2\pi\right], mod\left[-\left|\gamma(T) - \frac{\pi}{2}\right|, 2\pi\right]\} \tag{14}$$

Finite Thrust Problem

A common assumption of thrust maneuvers is that of impulsive thrust. While useful for hand calculations, vehicles actually use finite thrust, a subject of optimization. Optimal finite thrust should approach impulsive thrust performance. For this reason, a Hohmann transfer between the two orbits of this problem is used as the measure of success.

The first burn of this problem begins at $t_0 = 0$ with radius R_1 and ends at t_1. After the first burn, the vehicle coasts under the influence of its own inertia and gravity until time t_2, whereupon the second burn begins. This burn ends at time t_f with radius R_2. The equations of motion are in polar coordinates:

$$\dot{v_r} = \frac{\mu - r v_\theta^2}{r^2} + \frac{T}{m}\sin\delta \tag{15}$$

$$\dot{v_\theta} = -\frac{v_r v_\theta}{r} + \frac{T}{m}\cos\delta \tag{16}$$

$$\dot{r} = v_r \tag{17}$$

$$\dot{\xi} = \frac{v_\theta}{r} \tag{18}$$

where μ is the gravitational parameter of the center body, v_r is the radial speed, v_ϑ the transverse speed, T the thrust magnitude, δ the thrust direction (the angle from the local horizontal plane to the thrust vector), and *m* is the spacecraft mass. Assuming no throttling and maximum thrust is used during each burn, the thrust-to-mass ratio is given by the following piecewise function:

$$\frac{T}{m} = \begin{cases} \frac{T}{m_0 - \frac{T}{e}t} = \frac{en_0}{e - n_0 t} & if\ 0 \le t \le t_1 \\ 0 & if\ t_1 \le t \le t_2 \\ \frac{T}{m_0 - \frac{T}{e}(t_1 + t - t_2)} = \frac{en_0}{e - n_0(t_1 + t - t_2)} & if\ t_2 \le t \le t_3 \end{cases} \tag{19}$$

where *c* is the thruster's exhaust velocity and n_0 the initial value of the thrust-to-mass ratio. The thrust steering angle is represented as a third order polynomial in time:

$$\delta = \begin{cases} a_0 + a_1 t + a_2 t^2 + a_3 t^3 & if\ 0 \le t \le t_1 \\ b_0 + b_1(t - t_2) + b_2(t - t_2)^2 + b_3(t - t_2)^3 & if\ t_2 \le t \le t_f \end{cases} \tag{20}$$

with the coefficients a_i and b_i to be determined by the optimizer. For the first thrust arc, the following initial conditions are:

$$v_r(t_0) = 0 \;\; v_\theta(t_0) = \sqrt{\frac{\mu}{R_1}}$$

$$r(t_0) = R_1 \;\; \xi(t_0) = 0 \tag{21}$$

The initial values are selected by the optimizer and the equations of motion are integrated up to t_1. The coast arc is considered a Keplerian trajectory. The trajectory is assumed to be elliptical like the transfer arc of a Hohmann transfer. Using the final values of integration, the state values at t_2 can be found. First, the vis-viva integral is used to find the semi-major axis of the coast trajectory:

$$-\frac{\mu}{2a} = \frac{v_{r,1}^2 + v_{\theta,1}^2}{2} - \frac{\mu}{r_1} \tag{22}$$

The angular momentum and semi-latus rectum can then be found and used to obtain the eccentricity:

$$h = \sqrt{\mu p} = r v_\theta^2 \tag{23}$$

$$p = a(1 - e^2) \tag{24}$$

$$e = \sqrt{1 - \frac{r_1^2 v_{\theta,1}^2}{\mu a}} \tag{25}$$

The orbit equation can be used with equation (24) to find the following relationships:

$$r = \frac{p}{1 + e\cos f} \tag{26}$$

$$v_r = \dot{r} = \sqrt{\frac{\mu}{p}} e \sin f \tag{27}$$

$$v_\theta = \sqrt{\frac{\mu}{p}}(1 + e\cos f) \tag{28}$$

With this information, the true anomaly f_1 at time t_1 is then related to the velocity components by:

$$\sin f_1 = \frac{v_{r,1}}{e}\sqrt{\frac{a(1-e^2)}{\mu}} \quad \cos f_1 = \frac{v_{\theta 1}}{e}\sqrt{\frac{a(1-e^2}{\mu}} - \frac{1}{e} \tag{29}$$

which means that:

$$f_1 = \begin{cases} \text{Tan}^{-1}\frac{\sin f_1}{\cos f_1} & if\ \cos f_1 > 0 \\ \text{Tan}^{-1}\frac{\sin f_1}{\cos f_1} + \pi & if\ \cos f_1 < 0 \end{cases} \quad (30)$$

where $\text{Tan}^{-1}(x)$ is the principal value of $\tan^{-1}(x)$. Eccentric anomaly can now be found at t_1. This value is then used with ΔE, a value picked by the optimizer, to find E_2.

$$\tan\frac{E_1}{2} = \sqrt{\frac{1-e}{1+1}}\tan\frac{f_1}{2} \quad (31)$$

$$E_2 = E_1 + \Delta E \quad (32)$$

With the eccentric anomaly at time t_2, the true anomaly at t_2 can be found using Equation (31). With both anomalies found, the state values at time t_2 can be found:

$$v_{r,2} = \sqrt{\frac{\mu}{p}}e\sin f_2 + 2 \quad (33)$$

$$v_{\theta,2} = \sqrt{\frac{\mu}{p}}(1 + e\cos f_2) \quad (34)$$

$$r_2 = \frac{a(1-e^2)}{1+e\cos f_2} \quad (35)$$

$$\xi_2 = \xi_1 + (f_2 - f_1) \quad (36)$$

These values are used as the initial values for the second integration of the equations of motion (15)-(18). The results of the second burn arc should be:

$$v_r(t_f) = 0 \quad v_\theta(t_f) = \sqrt{\frac{\mu}{R_2}} \quad r(t_f) = R_2 \quad (37)$$

Therefore, the problem now operates with particles with the following elements:

$$\boldsymbol{p} = [a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3, \Delta t_1, \Delta E, \Delta t_2] \quad (38)$$

where $\Delta t_1 = t_1 - t_0$ and $\Delta t_2 = t_f - t_1$. R_1 is equal to 1 LU with a circular period orbit of 2π TU. This means the gravitational parameter is set to unity. The elements of the particle are bounded as follows:

$$0 \le \Delta t_1 \le 3\ TU, 0 \le \Delta t_2 \le 3\ TU, 0 \le \Delta E \le 2\pi$$

$$-1 \le a_i, b_i \le 1 \quad i = (0,1,2,3) \quad (39)$$

For the cases in this paper, c = 0.5 LU/TU and n_0 = 0.2 LU/TU^2. Equation (37) is incorporated into the objective function in addition to the thrust time.

$$J = \Delta t_1 + \Delta t_2 + \sum_{i=1}^{3} w_i |d_i| \tag{40}$$

The values of the weight functions w_i and penalty functions d_i are defined as the following:

$$d_1 = v_r(t_f) \quad d_2 = v_\theta(t_f) \quad d_3 = r(t_f) - R_2 \tag{41}$$

$$w_i = \begin{cases} 100 & if\ |d_i| > 10^{-3} \\ 0 & if\ |d_i| \leq 10^{-3} \end{cases} \ (i = 1,2,3) \tag{42}$$

Chapter 3

Results and Discussion

Inserting the equations of motion from the Lyapunov problem into the Differential Evolution and Particle Swarm algorithms allows for solution analysis. For this analysis, the population fraction, Phi, varied from 0.1 to 0.9. The convergence factor, CR, varied from 0.2 to 0.9. Lastly, the weighting exponent, m, had values of 0.5, 1.0, and 1.5. For each permutation of these parameters in differential evolution and for the control trial in particle swarm, 20 executions of the algorithm were used. Success was measured by the objective function reaching a certain accuracy. The percentage of successful runs within a permutation's 20 runs was noted as well as the lowest objective function value reached in permutations with a successful run.

In addition, the average success of individual parameters in the entire data set of a problem was measured. The top 5 most successful combinations of parameters were saved. Finally, time execution of both the control PSO and DE- Σ algorithm was qualitatively recorded. Quantitative data would be hard to track accurately with threading, machine differences and background programs.

Lyapunov Problem

A successful solution of the Lyapunov Problem matches closely with the solutions in Ref 1. The following figures are from the employed DE-Σ algorithm. They show the Lyapunov Orbit produced by the final parameter values and the history of the J value over 500 iterations.

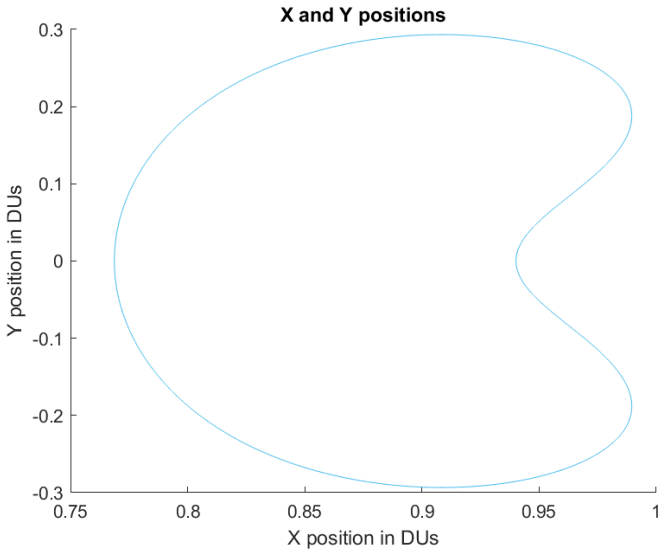

Figure 2. Lyapunov Orbit (m = 1.5, CR = 0.4, Phi = 0.3)

This seemingly single orbit is actually 20 orbits overlayed on top of each other. The algorithm at least shows high levels of precision in its solutions. The individual J histories of each solution can be seen:

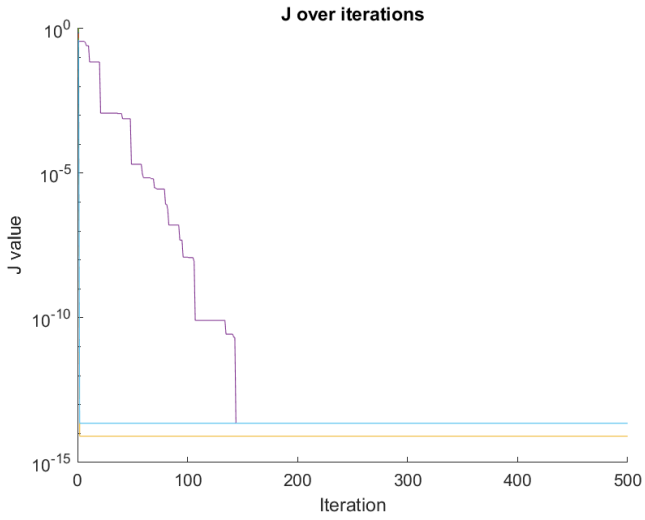

Figure 3 Time history evolution of objective value

In this particular set of permutations, the convergence to a final value happens rather quickly. Other parameters take all 500 iterations to reach a lowest value.

Across all 4320 runs for the Lyapunov problem, some patterns were noted. The following figures represent the average success rate across all permutations while only focusing on individual parameters. The control value of PSO's success rate was found to be 25% in 20 runs.

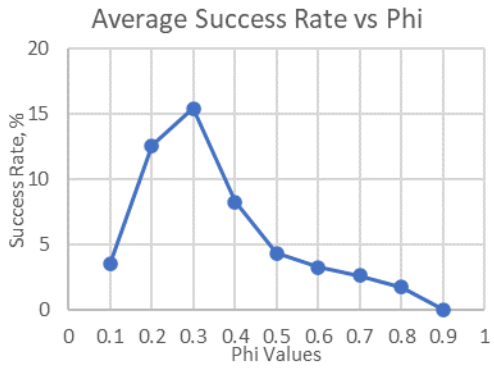

Figure 4. Average of all success rates arranged by associated Phi values

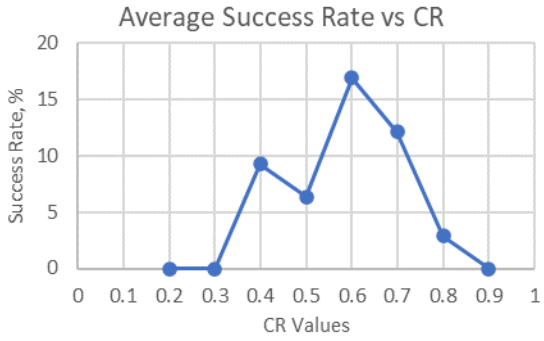

Figure 5. Average success rates by CR values

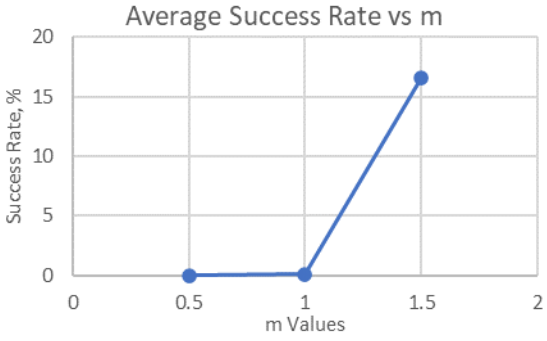

Figure 6. Average success rates by m value

While averages of a success rate is an abstract concept, it is useful to see how variance in parameters affects algorithm success. In this problem, success peaks earlier in population fraction values, later in convergence factor values and lastly in weighting exponent values.

While these figures seem to convey a low success rate for DE-Σ, the following table lists the most successful permutations for problem solving with their success rates and the lowest objective function value achieved.

Table 1. Top 5 Permutation Data for the Lyapunov DE-Σ Solutions

m	CR	Phi	Success Rate	Lowest J
1.5	0.4	0.3	100%	8.12E-15
1.5	0.6	0.4	100%	8.12E-15
1.5	0.6	0.5	100%	1.43E-14
1.5	0.7	0.2	95%	8.12E-15
1.5	0.6	0.3	90%	8.12E-15

Of note from this table, the 3 parameters all fall within the range where they all had the most success. A 100% success rate means that the algorithm successfully avoids the problem of false convergence, at least for the Lyapunov problem.

As a final note on this problem, the DE-Σ algorithm executes qualitatively faster than PSO. As stated before, it is hard to measure execution with the multitude of factors involved. More research is required for quantitative data on performance times.

FTX Problem

The finite thrust problem saw much higher rates of success with DE-Σ. As before with the Lyapunov problem, success was measured by a certain accuracy of the J function. In this case, J's target value was ~1. The finite thrust problem in Ref 1. was used with PSO as a control value. It was found to have a 35% success rate. An example J time history is given over 1000 iterations for this problem:

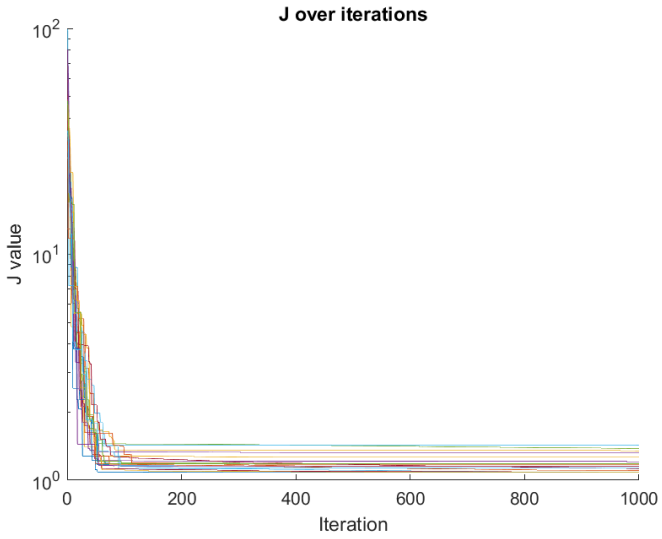

Figure 7. Time History (m = 1.0, CR = 0.4, Phi = 0.9)

Convergence happened relatively quickly in this problem as well. The problem as state in Ref. 1 has double the number of iterations as the Lyapunov problem. The success rates were arranged by parameters as before:

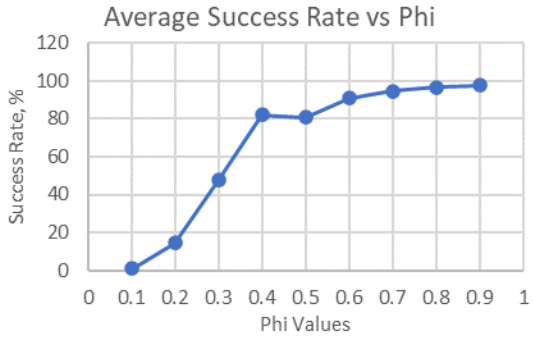

Figure 8. Average success rates by Phi values

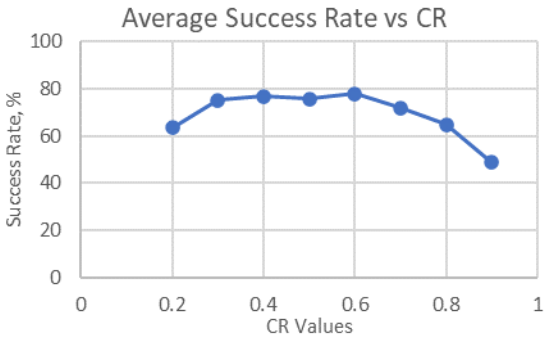

Figure 9. Average success rates by CR values

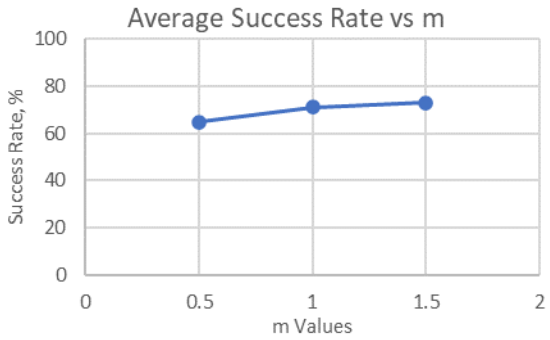

Figure 10. Average success rates by m values

Different peaks and behavior are seen in this problem. Success seemed to be barely affected by the weighting coefficient in this problem. The opposite behavior was seen in the Lyapunov problem for the population fraction, Phi. About the same type of behavior was seen in the convergence factor. Success peaked in the middle of the possible value range.

The top 5 combinations of parameters were tabulated below. The best parameters followed the behavior seen in the individual analysis of the parameters: all values of m, high values of phi, and medium values of CR.

Table 2. Top 5 Permutation Data for FTX DE-Σ Solutions

m	CR	Phi	Success Rate	Lowest J
0.5	0.5	0.9	100%	1.081519
0.5	0.4	0.9	100%	1.081532
1.5	0.5	0.8	100%	1.081572
1.5	0.6	0.8	100%	1.08162
1	0.4	0.9	100%	1.081622

High success was seen in these parameter combinations. In terms of performance, the algorithms operated at about the same speed. Again, more research is required.

Chapter 4

Conclusion

The three parameters had wildly different effects on the overall success of the DE- Σ in each problem. Other research into the parameters of unmodified differential evolution found that medium to high values of CR were effective [3]. Values of 0.6 to 0.8 were most effective in a variety of problems. A more general analysis of PSO and the unmodified DE algorithms using benchmark functions concurred with this finding [4]. It is worth noting that these studies also consider the effect of the mutation factor, F, during the evolution portion of DE. The mutation factor was held to a uniformly distributed random value between 0.4 and 0.9. The other parameters *m* and ϕ are unique to DE-Σ. Consistent behavior was hard to find for these parameters. Across the two problems of this study, the parameters exhibited opposite behaviors. The weighting exponent had almost no effect on success in the FTX problem but had a large effect on the solution of the Lyapunov problem. The population fraction parameter seemed to need a low value for high success in the Lyapunov problem. In the FTX problem, a high value of Phi was needed for high success.

Either way, with the right combination of parameters, the DE-Σ algorithm vastly outperforms the PSO algorithm while avoiding the pitfall of false convergence. This conclusion was found in comparisons of PSO and unmodified DE algorithms [5]. Other studies which went deeper into questions of execution time found that unmodified DE algorithms performed faster than PSO [6]. This slightly faster execution is seen qualitatively in DE-Σ. Further research is needed to determine if there is a pattern of optimal parameter values that changes based on the problem DE-Σ is used in, perhaps with the use of benchmark functions like other studies.

Appendix A

DE-Σ Algorithm

Initialize $\mathbf{P}_1$, a random population of N particles $\mathbf{p}_i$ within upper and lower bounds $\mathbf{B}_L$ and $\mathbf{B}_U$
Compute the centroid of bounds $\mathbf{C} = (\mathbf{B}_U + \mathbf{B}_L)/2$
for $j = 1 : N_{it}$
  $w = (1 - j/N_{it})^m$
  for $i = 1 : \phi N_P$
    Select three different solution vectors randomly from $\mathbf{P}_1$: $\mathbf{p}_a$, $\mathbf{p}_b$, $\mathbf{p}_c$
    Form the trial vectors in the Σ population:
    for $k = 1 : N_e$

$$\boldsymbol{q}(k) = \begin{cases} \boldsymbol{p_a}(k) + w * F * \left(\boldsymbol{p_b}(k) + \boldsymbol{p}_c(k) - \boldsymbol{C}(k)\right), & r < CR \\ \boldsymbol{p}_a(k), & otherwise \end{cases}$$

      if $\mathbf{q}(k) < \mathbf{B}_L(k)$ **then** $\mathbf{q}(k) = \mathbf{B}_L(k)$
      else if $\mathbf{q}(k) > \mathbf{B}_U(k)$ **then** $\mathbf{q}(k) = \mathbf{B}_U(k)$
      end if
    end for
  for $I = (\phi N_P) + 1 : N_P$
    Select three different solution vectors randomly from $\mathbf{P}_1$: $\mathbf{p}_a$, $\mathbf{p}_b$, $\mathbf{p}_c$
    Form the trial vectors in the Δ population:
    for $k = 1 : N_e$

$$\boldsymbol{q}(k) = \begin{cases} \boldsymbol{p_a}(k) + F * \left(\boldsymbol{p_b}(k) + \boldsymbol{p}_c(k)\right), & r < CR \\ \boldsymbol{p}_a(k), & otherwise \end{cases}$$

      if $\mathbf{q}(k) < \mathbf{B}_L(k)$ **then** $\mathbf{q}(k) = \mathbf{B}_L(k)$
      else if $\mathbf{q}(k) > \mathbf{B}_U(k)$ **then** $\mathbf{q}(k) = \mathbf{B}_U(k)$
      end if
    end for
    if $J(\mathbf{q}) < J(\mathbf{p}_i)$ **then** $\mathbf{P}_2(i) = \mathbf{q}$
    end if
  end for
  $\mathbf{P}_1 = \mathbf{P}_2$
end for

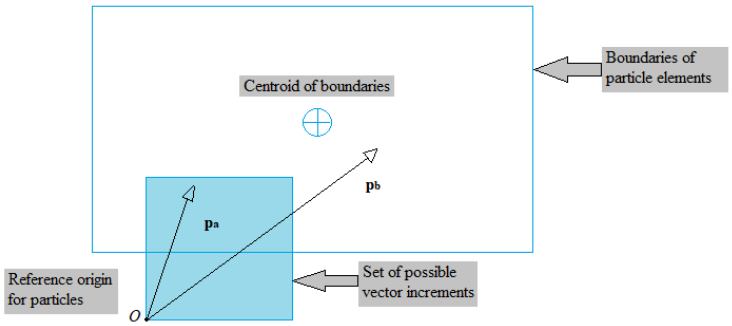

Figure 11. Representation of the titular evolution vector

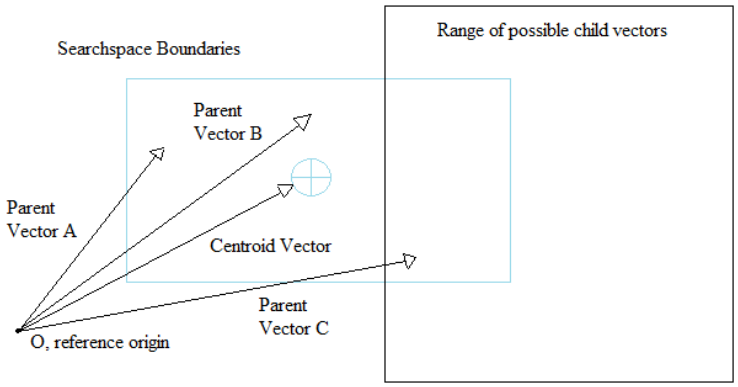

Figure 12. Crossover range overlaid on the search space.

The search space uses the formula for the q vector as seen in the pseudocode in the appendix. Note that most of the child vectors fall outside of the search space.

BIBLIOGRAPHY

[1] M. Pontani and B. A. Conway, “Particle Swarm Optimization Applied to Space Trajectories,” *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 5, 2010, pp 1429-1441.

[2] Robert G Melton, 2020, "DE-Sigma: A Modified Differential Evolution Algorithm", *Advances in the Astronautical Sciences*, Univelt, Inc., San Diego, CA

[3] A. D. Old and C. A. Kluever, “Interplanetary Mission Design Using Differential Evolution,” Journal of Spacecraft and Rockets, Vol. 44, No. 5, 2007, pp 1060-1070

[4] M. Iwan, R. Akmeliawati, T. Faisal, and H. Al-Assadi, “Performance Comparison of Differential Evolution and Particle Swarm Optimization In Constrained Optimization,” *Procedia Engineering*, Vol. 41, 2012, pp 1323-1328

[5] J. Vesterstrøm and R. Thomsen, “A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems,” *Proceedings of the 2004 Congress on Evolutionary Computation*, IEEE, Portland, OR

[6] S. P. Lim and H. Haron, “Performance Comparison of Genetic Algorithm, Differential Evolution and Particle Swarm Optimization Towards Benchmark Functions,” *2013 IEEE Conference on Open Systems*, IEEE, Kuching, Malaysia

ACADEMIC VITA

John Quinn

juq35@psu.edu

OBJECTIVE:

To acquire a full-time aerospace engineering position focused on space and design

EDUCATION:

The Pennsylvania State University Anticipated: May 2021

Intended Major: Aerospace Engineering
Honors: Schreyer Honor College, Abington Honors and Fellows Scholar, Dean's List
Relevant Coursework: Satellite and Orbital Mechanics, Space Craft Design, Propulsion

WORK EXPERIENCE:

Ultra Electronics Herley Division, Lancaster, PA May 2020-August 2020

Software Engineering Intern

- Worked on with automatic testing equipment to test company hardware
- Developed a software program for tracking employee work hours
- Performed product validation on software releases for System Engineers

The Pennsylvania State University, Abington, PA October 2018-May 2019

Peer Assistant Tutor

- Tailored learning experience for each person to maximize understanding of concepts
- Tutored students struggling with math courses on campus
- Utilized communication skills to work past students dislike of course subject

RESEARCH EXPERIENCE:

Schreyer Thesis, Penn State, State College PA April 2020-Present

- Researched DE Sigma and Particle Swarm trajectory optimization methods
- Developing MatLab program to calculate best trajectory using both methods
- Will publish findings to graduate with honors in the spring

ACURA, Penn State Abington, Abington PA October 2017-April 2018

- Designed parts of a machine to simulate the slip and fall movements in human leg muscles
- Developed data entry and data analysis skills
- Published a presentation at the American Society of Biomechanics East Coast Meeting

TECHNICAL SKILLS:

- Proficient with**:** Labview, AutoCAD, SOLIDworks, C++, MatLab, Microsoft Office
- Experience in a Laboratory, Office and Manufacturing environment

EXTRACURRICULARS:

- Member of Tau Beta Pi Honors Engineering Society
- Member of the Abington Engineering Club
- Member of the Civitus Victus Dictio intellectual society