A Numerical Solution to Orbital Pursuit-Evasion Games

Dan Shen
Intelligent Fusion Technology, Inc., dshen@intfusointech.com
Khanh Pham
Air Force Research Laboratory (AFRL/RVSWS), khanh.pham.1@us.afrl.mil
Genshe Chen
Intelligent Fusion Technology, Inc., gchen@intfusiontech.com

ABSTRACT

Pursuit-evasion (PE) games are mathematical tools to model the space situational awareness (SSA) problems, such as satellite interception, collision avoidance, and space sensor management. Early work in pursuit-evasion games took place before the wide availability of computers and software packages. Consequently, the application of PE game theory to useful problems has been limited, since all calculations had to be done analytically. In this paper, we developed enhanced indirect solutions by combing two-sided non-linear programming and multi-objective optimization based initial solution guess for orbital PE games. In the proposed solution framework, the first step is to obtain a good initial value, and the second step is to iteratively revise the solution by using an open source non-linear programming (NLP) solver, Nonlinear Optimization (NLOPT). The better the initial value, the faster the numerical solution converges. In this paper, we apply nondominated sorting genetic algorithm II (NSGA II, a dominance-based algorithm) to generate an initial guess for our numerical game solution. Numerical results are simulated to demonstrate the effectiveness of the enhanced numerical PE game solution.

1. INTRODUCTION

Pursuit-evasion (PE) games [1,2] are mathematical tools to model the space situational awareness (SSA) problems, such as satellite interception, collision avoidance, and space sensor management. Early work in pursuit-evasion games took place before the wide availability of computers and software packages. Consequently, the application of PE game theory to useful problems [3,4,5] has been limited, since all calculations had to be done analytically. It is well known [6] that the solution of pursuit-evasion games with realistic dynamics must be found numerically. As is the case for optimal control problems, numerical solutions for PE games often fall into two categories, direct methods and indirect methods. The basic procedure for indirect methods is to take the system dynamics, form the Hamiltonian, derive the necessary conditions, and then solve the boundary value problem numerically using information about the states and costates at the boundaries. Conversely, in direct methods, the game problem is converted to a nonlinear programming problem and the solvers only need to know the system dynamics, control and state constraints, and the objective function, without the need for costate information.

Indirect methods are marked by their high accuracy and the production of a solution that satisfies the necessary conditions of optimality. In this paper, we developed enhanced indirect solutions by combing two-sided non-linear programming (NLP) and multi-objective optimization based initial solution guess for orbital PE games, where the pursuer minimizing the satellite interception time while the evader maximizing interception time for collision avoidance. The interception-avoidance (IA) PE game approach provides a worst-case solution, which is the robust lower-bound performance case.

In the proposed solution framework, the first step is to obtain a good initial value, which will be iteratively revised by an open source NLP solver, NLOPT [7]. The better the initial value, the faster the numerical solution converges. In this paper, we develop a multi-objective optimization approach to approximately solve PE game problems and use the approximate solution as the initial value for NLOPT. In the literature, there are three categories of evolutionary solutions for multiple objective problems: dominance-based algorithms, aggregation-based algorithms, and indicator-based algorithms. In this paper, we apply NSGA II (a dominance-based algorithm) [8] to generate an initial guess for our numerical game solution. The main reason is that NSGA II has been coded in SciLab [9], which is free and open source software for numerical computation providing a powerful computing environment for engineering and scientific applications. As a user case, a two-satellite IA problem is modelled using a PE game and is simulated to demonstrate the effectiveness of the enhanced numerical PE game solution. Various multi-objective settings are tested and analyzed to obtain a good initial value. Then the settings of NLOPT to solve the PE game are presented and results are discussed. At the end, the conclusion is drawn and the future work is laid out too.

2. PURSUIT-EVASION GAME AND SOLUTION FRAMEWORK

For a general pursuit-evasion game with pursuer player *P* and evader player *E*, the system dynamics is

$$\dot{x} = \begin{bmatrix} \dot{x}_p \\ \dot{x}_e \end{bmatrix} = \begin{bmatrix} f_p(x_p,\, u_p,\, t) \\ f_e(x_e,\, u_e,\, t) \end{bmatrix} \tag{1}$$

where, x_p is the state of player *P*, and x_e is the state of player *E*. The f_p and f_e are the state function of player *P* and *E*, respectively. The initial state is denoted as $x_0 = x(t_0)$, and the terminal constraints satisfies the following equation.

$$\Psi\left(x_o,\, x_f,\, t_f\right) = 0 \tag{2}$$

A general cost function is

$$J(u_p,\, u_e) = \phi\left(x_f,\, t_f\right) + \int_{t_o}^{t_f} L\left(x,\, u_p,\, u_e,\, t\right) dt \tag{3}$$

In this paper, the cost function is minimized by pursuer and maximized by evader:

$$J\left(u_p^*,\, u_e\right) \leq J\left(u_p^*,\, u_e^*\right) \leq J\left(u_p,\, u_e^*\right) \tag{4}$$

To derive the analytic necessary conditions, we defined a Hamiltonian:

$$H \equiv \lambda^\intercal f + L = \lambda_p^\intercal f_p + \lambda_e^\intercal f_e + L \tag{5}$$

where λ is the vector of costates.

The terminal conditions are

$$\Phi \;\equiv\; \phi \;+ \nu^\intercal \Psi \tag{6}$$

Then costates should satisfy

$$\dot{\lambda} = -H_x = -\left[\frac{\partial f}{\partial x}\right]^\intercal \lambda \,-\, \frac{\partial L}{\partial x} \tag{7}$$

and the costate terminal conditions:

$$\lambda^\intercal\left(t_f\right) = \frac{\partial \Phi}{\partial x}\left(x_f,\, t_f\right) \tag{8}$$

The saddle-point solution must satisfy the conditions:

$$u_p^* = arg \min_{u_p} H \tag{9}$$

$$u_e^* = arg \max_{u_e} H \tag{10}$$

If there are no constraints for controls, then

$$H_{u_p}^\intercal = 0 \text{ and } H_{u_p u_p} \geq 0,\; H_{u_e}^\intercal = 0 \text{ and } H_{u_e u_e} \leq 0 \tag{11}$$

If the final time t_f is free, then the following transversality condition must be satisfied:

$$H\left(x_f^*,\, u_p^*,\, u_e^*,\, \lambda\left(t_f\right),\, t_f\right) = -\frac{\partial \Phi}{\partial t_f} = -\frac{\partial \phi}{\partial t_f} - \nu^\intercal \frac{\partial \Psi}{\partial t_f} \tag{12}$$

The first part of our numerical game solution is to convert a two-sided game problem to a one-sided optimal control problem. In this step, we use the derived analytic necessary conditions (1)-(12) to replace evader controls with costates and states:

$$u_e = u_e(x_e,\ \lambda_e,\ t) \tag{13}$$

Then we solve an extended system with extended control $\tilde{u} = u_p$ and extended system states

$$\tilde{x} = \left[x_p^{\intercal}(t), x_e^{\intercal}(t), \lambda_e^{\intercal}(t)\right]^{\intercal} \tag{14}$$

Based on the equation (1) and (7), the extended states must satisfy (for the simplicity purpose, we assume $\frac{\partial L}{\partial x_e} = 0$):

$$\dot{\tilde{x}} = \left[f_p^{\intercal},\ \bar{f}_e^{\intercal},\ -\lambda_e^{\intercal}\frac{\partial f_e}{\partial x_e}\right]^{\intercal} \triangleq \tilde{f} \tag{15}$$

where $\bar{f}_e(x_e,\ \lambda_e,\ t) = f_e(x_e,\ u_e(x_e,\ \lambda_e,\ t),\ t)$.

Using the extended system, we convert the two-sided (pursuer and evader) pursuit-evasion game into a one-sided (pursuer only) optimal control problem:

$$\min_{\tilde{u}} J \text{ s.t. } \dot{\tilde{x}}(t) = \tilde{f} \text{ and } \widetilde{\Psi} = 0 \tag{16}$$

where in $\widetilde{\Psi}$ we select components from eq. (6) such that these components are independent of *v* since the *v*-dependent terminals can be handled by *v*.

The problem in eq. (16) cannot be directly solved by nonlinear programming (NLP) because it contains the differential constraints. We apply the collocation method to approximate these differential conditions.

We partition the time interval into *N* subarcs: $[t_{i-1},\ t_i]_{i=1,\dots,N}$ with $t_N = t_f$. Then we employ the following direct collocation techniques to discretize a given differential equation $\dot{x} = f(x, u)$

$$x_{i+1} - x_i \approx \Delta t \cdot f(x_i,\ u_i)\ \ :\ \ Rectangle\ Rule \tag{17}$$

$$x_{i+1} - x_i \approx \frac{\Delta t}{2}[(f(x_{i+1},\ u_{i+1}) + f(x_i,\ u_i)]:\ \ Trapezoidal\ Rule \tag{18}$$

$$x_{k+1} - x_k \approx \frac{\Delta t}{6}[(f(x_{k+1},\ u_{k+1}) + 4f(x_{k+o.5},\ u_{k+o.5}) + f(x_k,\ u_k)]: Simpson's\ Rule \tag{19}$$

Then we obtained a nonlinear optimal control problem, which can be solved by NLP solvers, which can interactively revise numerical solutions. The solution framework needs an initial guess. The better the initial values, the faster the numerical ones converge.

3. MULTI-OBJECTIVE OPTIMIZATION

In this section, we developed a multi-objective optimization approach to approximately solve PE game problems. The approximate solution will provide a good initial start point for the NLP solvers.

In general, a multiple objective minimization problem is

$$min\ \boldsymbol{J}(\boldsymbol{x}, \boldsymbol{p}) \tag{20}$$

s.t.

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{p}) \leq 0$$

$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{p}) = 0$$

$$x_{i,LB} \leq x_i \leq x_{i,UP}\ ,\ (i = 1,2, \dots, n)$$

where p is a vector of fixed parameters (constants). $\boldsymbol{x} = [x_1,\ x_2,\ \ldots,\ x_n]^\intercal$ is the vector of n variables bounded by side constraints $x_{i,LB} \le x_i \le x_{i,UP}$, $(i = 1,2, \ldots, n)$. The k-multiple objective function is

$$\boldsymbol{J}(\boldsymbol{x},\boldsymbol{p}) \ = \ [f_1(\boldsymbol{x},\boldsymbol{p}),\ f_2(\boldsymbol{x},\boldsymbol{p}),\ \ \ldots,\ f_k(\boldsymbol{x},\boldsymbol{p})]^\intercal \tag{21}$$

There are m_1 inequality and m_2 equality constraints:

$$\boldsymbol{g}(\boldsymbol{x},\boldsymbol{p}) = \ \left[g_1(\boldsymbol{x},\boldsymbol{p}),\ g_2(\boldsymbol{x},\boldsymbol{p}),\ \ \ldots,\ g_{m_1}(\boldsymbol{x},\boldsymbol{p})\right]^\intercal \tag{22}$$

$$\boldsymbol{h}(\boldsymbol{x},\boldsymbol{p}) = \ \left[h_1(\boldsymbol{x},\boldsymbol{p}),\ h_2(\boldsymbol{x},\boldsymbol{p}),\ \ \ldots,\ h_{m_2}(\boldsymbol{x},\boldsymbol{p})\right]^\intercal \tag{23}$$

Let y and z be two feasible solutions to the above k-objective minimization problem, if

$$\forall i\colon f_i(\boldsymbol{y},\boldsymbol{p}) \le f_i(\boldsymbol{z},\boldsymbol{p}),\ \text{and}\ \exists j\colon f_j(\boldsymbol{y},\boldsymbol{p}) < f_j(\boldsymbol{z},\boldsymbol{p}) \tag{24}$$

then, y is better than z, or y dominates z, or z is dominated by y. When a feasible solution y is not dominated by any other feasible solutions, y is a Pareto-optimal solution. The set of all Pareto-optimal solutions forms the trade-off surface in the objective space. Usually, the trade-off surface is referred to as the Pareto Front (PF).

In the literature, there are three categories of evolutionary solutions for multiple objective problems: dominance-based algorithms [10,11], aggregation-based algorithms [12], indicator-based algorithms [13,14]. In this paper, we decided to apply NSGA II [8] to generate an initial guess for our numerical game solution. The main reason is that NSGA II has been coded in SciLab, which is free and open source software for numerical computation providing a powerful computing environment for engineering and scientific applications. SciLab includes hundreds of mathematical functions. It has a high-level programming language allowing access to advanced data structures, 2-D and 3-D graphical functions. For example, the ZDT 3 problem defined as $\min\{f_1,\ f_2\}$, $f_1 = x_1$, $f_2 = g.\left(1 - \sqrt{\frac{x_1}{g}}\right) - \frac{x_1}{g} sin(10\pi x_1)$, $g(x_2,\ x_3, \ldots, x_n) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$, and $0 \le x_i \le 1$. We implemented the SciLab codes (Figure 1) and obtained the correct results in Figure 2.

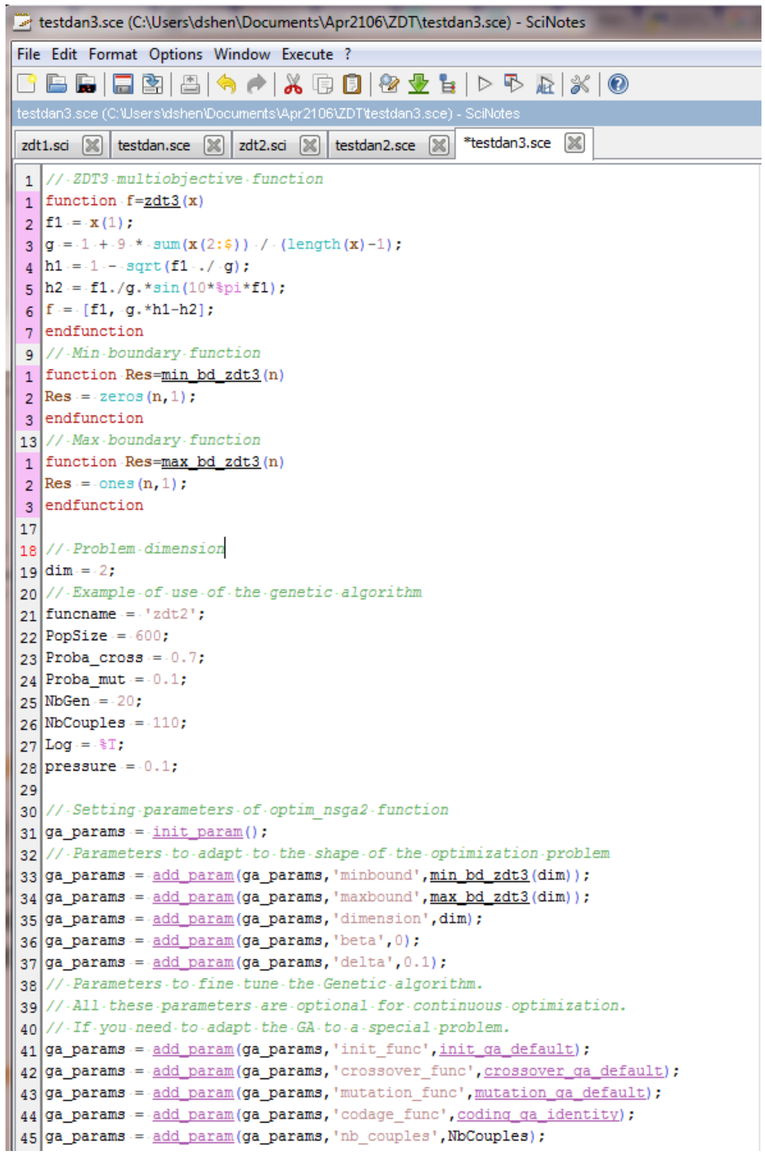

Figure 1: Scilab Codes for ZDT3 problem

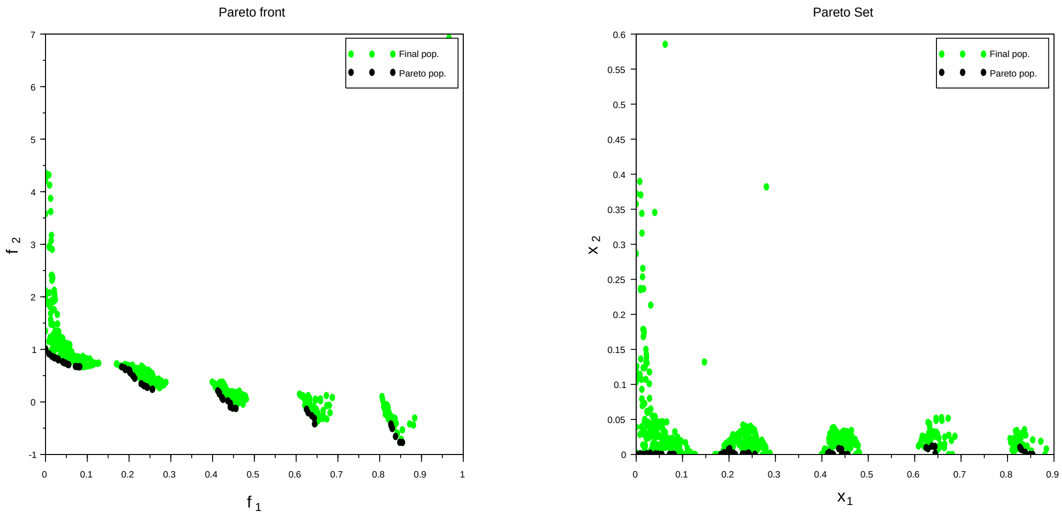

Figure 2: Results of multi-objective optimization ZDT 3 problem

4. CASE STUDY: NUMERICAL SOLUTION FOR ORBITAL PE GAMES

For the orbital PE games, we use the following states to describe the kinematics and dynamics of the spacecrafts *i* =*p* (pursuer) or *e* (evader):

$$\dot{r}_i = v_i \sin\gamma_i \tag{25}$$

$$\dot{v}_i = \frac{T_i}{m_i}\cos\alpha_i \cos\beta_i - \frac{\mu \sin\gamma_i}{r_i^2} \tag{26}$$

$$\dot{\gamma}_i = \frac{v_i \cos\gamma_i}{r_i} + \frac{T_i}{m_i}\frac{\sin\alpha_i \cos\beta_i}{v_i} - \frac{\mu\cos\gamma_i}{r_i^2 v_i} \tag{27}$$

$$\dot{\xi}_i = \frac{v_i \cos\gamma_i \cos\zeta_i}{r_i \cos\phi_i} \tag{28}$$

$$\dot{\phi}_i = \frac{v_i \cos\gamma_i \sin\zeta_i}{r_i} \tag{29}$$

$$\dot{\zeta}_i = \frac{T_i}{m_i}\frac{\sin\beta_i}{v_i\cos\gamma_i} - \frac{v_i\cos\gamma_i \sin\phi_i \cos\zeta_i}{r_i^2 \cos\phi_i} \tag{30}$$

The set of variables (*r*, *v*, *γ*, *ζ*, *ξ*, *ϕ*) defines the 3-D motions of spacecrafts. As shown in **Figure 3**, *r* is the instantaneous radius from Earth center, *v* is the velocity magnitude, *γ* is the flight path angle. *ζ* is the velocity azimuth angle, *ξ* is the right ascension, and ϕ is the declination. T is the thrust (here we assume the magnitude of the thrust is fixed constant) and m is mass of the satellite. The control is conducted with the thrust direction, specified by the two angles *α* and *β*. μ_E is the standard gravitational parameter of Earth. μ_E=398600 (km^3/s^2). Local horizon plane is perpendicular to the position vector $\hat{r}$. Instantaneous plane is determined by the position vector $\hat{r}$ and velocity vector $\hat{v}$.

The system states are $x_p = \left[r_p,\ v_p,\ \gamma_p,\ \xi_p,\ \phi_p,\ \zeta_p\right]^\intercal = [x_1,\ x_2, x_3,\ x_4.\ x_5,\ x_6]^\intercal$ and $x_e = [r_e,\ v_e,\ \gamma_e,\ \xi_e,\ \phi_e,\ \zeta_e]^\intercal = [x_7, x_8. x_9, x_{10}, x_{11}, x_{12}]^\intercal$. The controls for pursuer and evader are $u_p = [u_1,\ u_2]^\intercal = \left[\alpha_p,\ \beta_p\right]^\intercal$ and $u_e = [u_3,\ u_4]^\intercal = [\alpha_e,\ \beta_e]^\intercal$, respectively.

Since the cost function is $J = t_f$, then eq. (3) becomes

$$J\left(u_p,\ u_e\right) = \phi\left(x_f,\ t_f\right) + \int_{t_0}^{t_f} L\left(x,\ u_p,\ u_e,\ t\right)dt = t_f \tag{31}$$

From the above equation, we know that $L = 0$ and $\phi = t_f$. The Hamiltonian is

$$H \equiv \lambda^\intercal f + L = \lambda_p^\intercal f_p + \lambda_e^\intercal f_e + L\ = \lambda_p^\intercal f_p + \lambda_e^\intercal f_e \tag{32}$$

where $\lambda_p = [\lambda_1, \lambda_2, \dots \lambda_6]^\intercal$ and $\lambda_e = [\lambda_7, \lambda_8, \dots \lambda_{12}]^\intercal$.

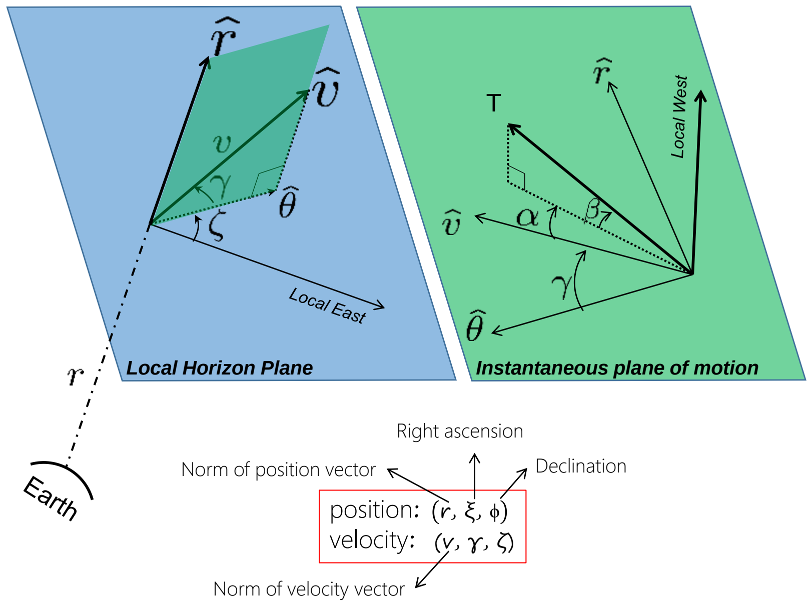

Figure 3: Spacecraft system states and controls.

Then the equation (7), $\dot{\lambda} = -H_x = -\left[\frac{\partial f}{\partial x}\right]^{\mathsf{T}} \lambda - \frac{\partial L}{\partial x}$, can be specified as 12 scalar differential equations. For example,

$$\dot{\lambda}_1 = \left(x_2\lambda_3 \cos x_3 + \lambda_4 x_2 \cos x_3 \frac{\cos x_6}{\cos x_5} + x_2\lambda_5 \cos x_3 \sin x_6 - x_2\lambda_6 \cos x_3 \sin x_5 \frac{\cos x_6}{\cos x_5}\right)\frac{1}{x_1^2} - \left(2\mu_E\lambda_2 \sin x_3 + \frac{2\mu_E\lambda_3 \cos x_3}{x_2}\right)\frac{1}{x_1^3} \tag{33}$$

and

$$\dot{\lambda}_{12} = \left(\frac{\partial f_e}{\partial x_{12}}\right)[\lambda_7, \dots \lambda_{12}]^{\mathsf{T}} = \frac{x_8 \cos x_9}{x_7 \cos x_{11}}(\lambda_{10} \sin x_{12} - \lambda_{11} \cos x_{11} \cos x_{12} - \lambda_{12} \sin x_{11} \sin x_{12}) \tag{34}$$

The terminal conditions:

$$\Psi = \begin{bmatrix} x_{1f} - x_{7f} \\ x_{4f} - x_{10f} \\ x_{5f} - x_{11f} \end{bmatrix} = 0 \tag{35}$$

Equation (6) becomes:

$$\Phi \equiv \phi + v^{\mathsf{T}}\Psi = t_f + v_1\left(x_{1_f} - x_{7_f}\right) + v_2\left(x_{4_f} - x_{10_f}\right) + v_3\left(x_{5_f} - x_{11_f}\right) \tag{36}$$

Then the terminal conditions for costates are:

$$\lambda^{\mathsf{T}}(t_f) = \frac{\partial \Phi}{\partial x}(x_f, t_f)$$

Substitute Φ with equation (36), we obtained the following scalar terminal conditions:

$$\lambda_{1_f} = v_1 \tag{37}$$
$$\lambda_{4_f} = v_2 \tag{38}$$
$$\lambda_{5_f} = v_3 \tag{39}$$
$$\lambda_{7_f} = -v_1 \tag{40}$$
$$\lambda_{10_f} = -v_2 \tag{41}$$
$$\lambda_{11_f} = -v_3 \tag{42}$$
$$\lambda_{2_f} = \lambda_{3_f} = \lambda_{6_f} = \lambda_{8_f} = \lambda_{9_f} = \lambda_{12_f} = 0 \tag{43}$$

For this application, we divided the time interval into N=14 subarcs. At each point, there are 17 (extended states) + 2 (pursuer controls) =19 variables. There are total 19*15(N+1 points) + 2 (for t_f and λ_{10}) = 287 variables. At each point (except the end point $t_f = t_N$), there are 17 (dimension of extended states) scalar constraints. In addition, there are 12 initial scalar constraints and 6 terminal constraints ($x_{1f} = x_{7f}, x_{4f} = x_{10f}, x_{5f} = x_{11f}, \lambda_{8f} = 0, \lambda_{9f} = 0, \lambda_{12f} = 0$). Therefore, we have total of 17*14 (points except the end point) + 18 = 256 scalar constraints.

To solve the nonlinear optimal problem, we employed the open source NLP solver, NLOPT, which has many algorithms, many language bindings, global and local optimizers, including derivative-free and gradient-driven ones. We apply the NSGA II to find a good initial point for the NLOPT. The orbital PE game problem has 13 variables $[\lambda_{1,0},\ \lambda_{2,0}, \ldots, \lambda_{12,0},\ t_f]^{\mathrm{T}}$, where, $\lambda_{i,0}$ is the initial values of costate λ_i, and t_f is the final time. It has 13 constrains. we portioned the 13 constrains into 2 groups: The first function f_1 is relevant to the interception requirement. The second f_2 is about the necessary conditions for the analytic analysis. For this 2-objetive optimization, we applied the NSFA II algorithm in SciLab. The calculated Pareto-Front (PF) is shown in Figure 4.

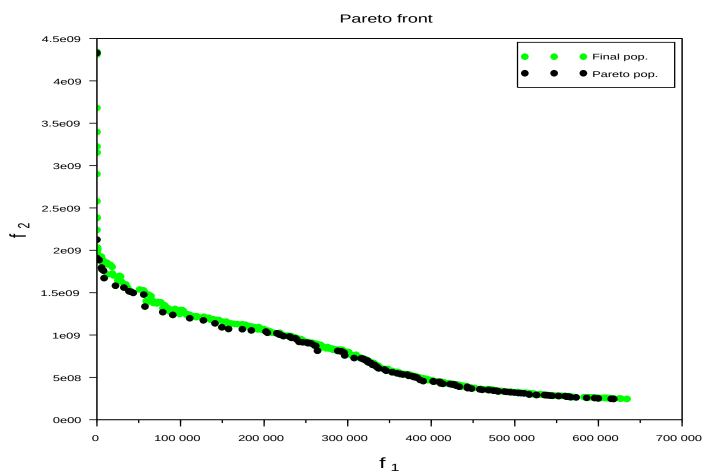

Figure 4: The PF of the 2-objetive minimization problem related to the orbital PE games

The Pareto set are shown Figure 5. We obtained the following mean values:

- 2.4913742 (λ_1)
51.335386 (λ_2)
- 18.226864 (λ_3)
54.089533 (λ_4)
11.348011 (λ_5)
- 37.18182 (λ_6)
5.7826764 (λ_7)
- 51.434703 (λ_8)
- 51.663036 (λ_9)
31.779191 (λ_{10})
9.2538513 (λ_{11})
9.0133522 (λ_{12})
2352.2112 (t_f)

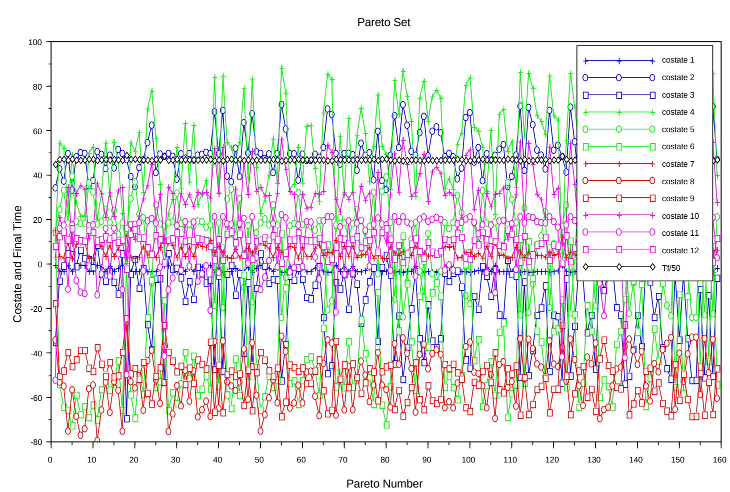

Figure 5: The values of the Pareto set for NSGA II algorithm for 2-objective minimization

Using the initial guess from NSGA II, we run the NLOPT engine and obtained good results as shown in Figure 6.

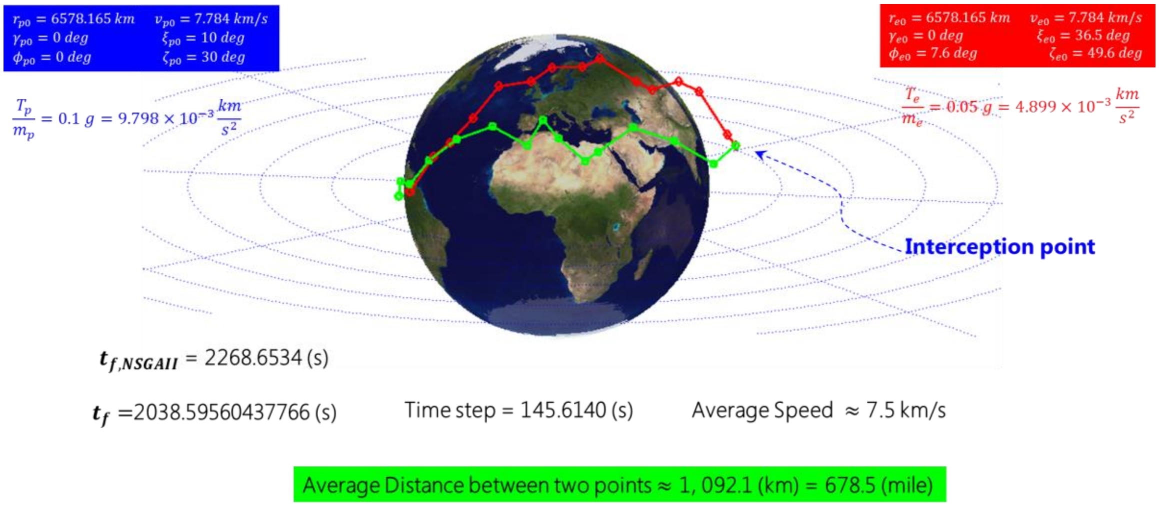

Figure 6: NLOPT results (orbits) with initial guess from NSGA II

The fish tank views are shown in the Figure 7.

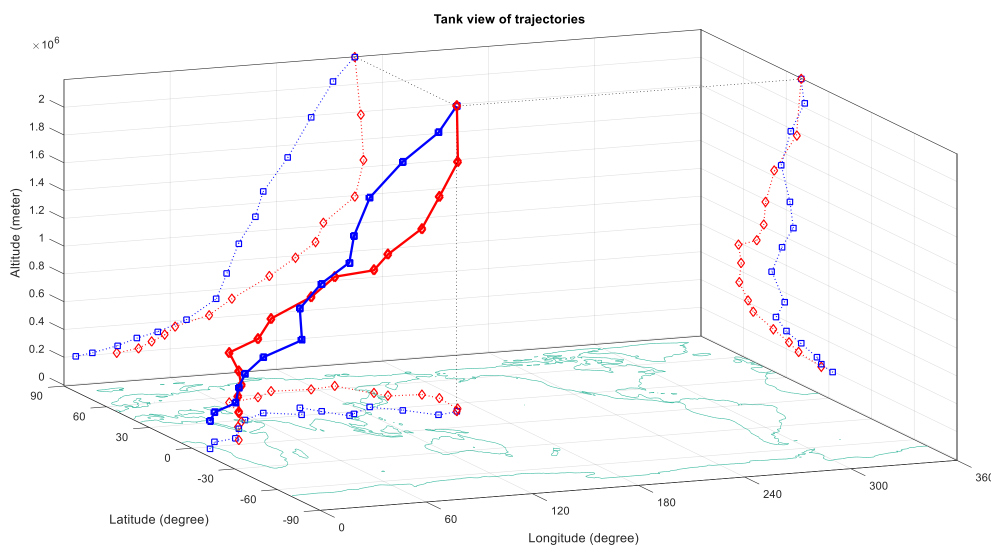

Figure 7: Fish tank view of the orbits engaged in an orbital PE game
(the dotted line shown the orbits projected on 2-D views).

The first and second system state, *i.e.*, the distance to Earth center and velocity, are shown in the Figure 8 and Figure 9. In Figure 8, we can see the NSGA II solution is not accurate because it cannot guarantee the final interception. The NSGA II's objective is to minimize the 13 constraints. On the other hand, NLOPT-NAGA II (the NLOPT with initial guess from NSGA II) tries to solve the interception and collision avoidance problem. At the final time, r_p and r_e are almost the same. In Figure 9, the velocities are from the same intimal condition. Because the controls are different, the final velocities are different.

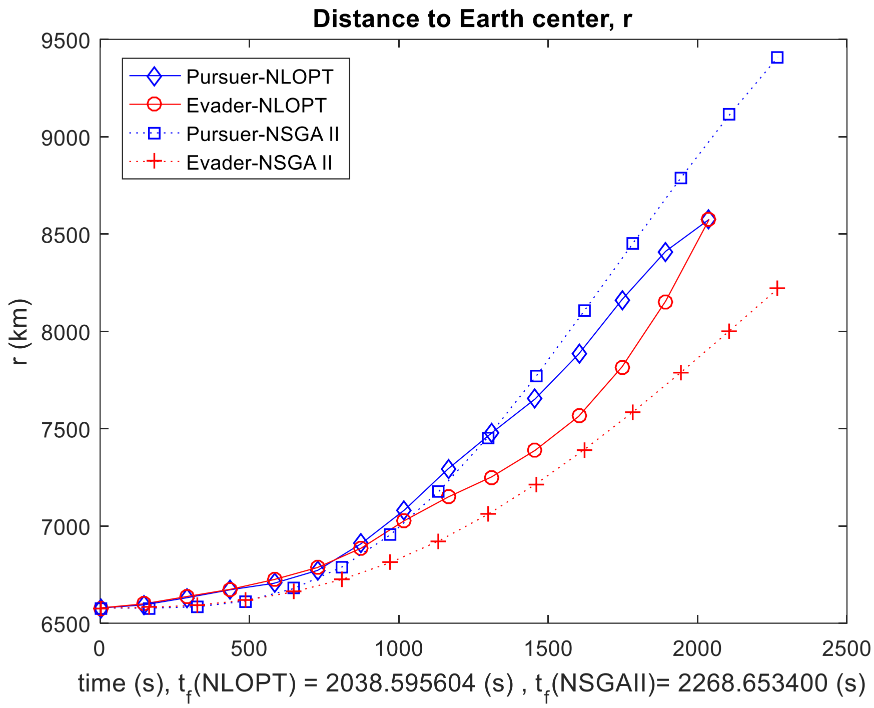

Figure 8: State 1 (r, distance to Earth center) trajectories with different control solutions.

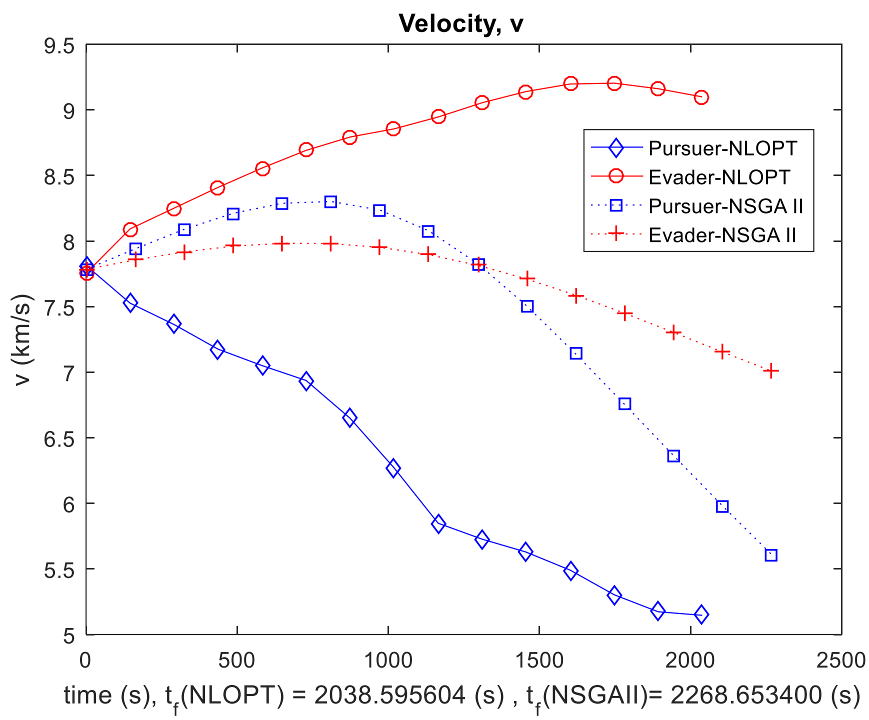

Figure 9: State 2 (velocity) trajectories with different control solutions.

5. CONCLUSION AND FUTURE DIRECTION

In this paper, we developed a numerical solution framework for a class of orbital pursuit-evasion games. In the framework, we first use multi-objective optimization package, NSGA-II, to directly solve the PE game. Then the approximate solution is used as a good initial point in the second step: solving the PE game as a nonlinear optimal problem. We applied the solution framework on an orbital PE game model for orbital interception and collision avoidance problems. We obtained promising results. As a future research direction, the smaller sampling interval and more refined solution will be explored.

6. ACKNOWLEDGEMENTS

This work was supported by contract FA9453-15-C-0423 from the Air Force Research Laboratory. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the United States Air Force.

7. REFERENCES

1. Isaacs, R., *Differential Games*, Wiley, New York, 1965.
2. Shinar, J., “Solution techniques for realistic pursuit-evasion games”, In: Leondes CT (ed), *Advances in Control and Dynamic Systems*, Academic Press, New York, vol. 17, 1981, pp. 63–124.
3. Shen, D., Pham, K., Blasch, E., Chen, H., and Chen, G., “Pursuit-evasion orbital game for satellite interception and collision avoidance”, In *Sensors and Systems for Space Applications IV, SPIE*, 2011.
4. D. Li, J. B. Cruz, G. Chen, C. Kwan, and M. Chang, “A hierarchical approach to multi-player pursuit-evasion differential games,” in *CDCECC ’05*. 44th IEEE Conference on Decision and Control, 2005.
5. X. Wang, J. B. C. Jr., G. Chen, K. Phamc, and E. Blasch, “Formation control in multi-player pursuit evasion game with superior evaders,” in *Proc. SPIE*, Orlando, FL, USA, May 2007.
6. Johnson, Philip Arthur, *Numerical solution methods for differential game problems*. Diss. Massachusetts Institute of Technology, 2009.
7. NLOPT, https://nlopt.readthedocs.io/en/latest/
8. Deb, Kalyanmoy, Amrit Pratap, Sameer Agarwal, and T. A. M. T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE transactions on evolutionary computation* 6, no. 2 (2002): 182-197.
9. Scilab, https://www.scilab.org/
10. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002), “A fast and elitist multiobjective genetic algorithm: NSGA-II”, *Evolutionary Computation, IEEE Transactions on*, 6(2), 182-197, 2002.
11. Zitzler, Eckart, Marco Laumanns, and Lothar Thiele. "SPEA2: Improving the strength Pareto evolutionary algorithm." *TIK-report* 103 (2001).
12. Zhang, Qingfu, and Hui Li. "MOEA/D: A multiobjective evolutionary algorithm based on decomposition." *Evolutionary Computation, IEEE Transactions on* 11.6 (2007): 712-731.
13. Zitzler, Eckart, and Simon Künzli. "Indicator-based selection in multiobjective search." *Parallel Problem Solving from Nature-PPSN VIII*. Springer Berlin Heidelberg, 2004.
14. Beume, Nicola, Boris Naujoks, and Michael Emmerich. "SMS-EMOA: Multiobjective selection based on dominated hypervolume." *European Journal of Operational Research* 181.3 (2007): 1653-1669.