SCIENCE CHINA
Information Sciences

• LETTER •
Special Focus on Pigeon-Inspired Optimization

July 2019, Vol. 62 070209:1–070209:3
https://doi.org/10.1007/s11432-018-9691-8

Spacecraft formation reconfiguration trajectory planning with avoidance constraints using adaptive pigeon-inspired optimization

Bing HUA, Yu HUANG*, Yunhua WU, Zhiming CHEN & Dery NICHOLAS

Microsatellite Research Center, College of Aeronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016*, China*

Received 16 July 2018/Revised 14 September 2018/Accepted 30 November 2018/Published online 24 April 2019

Citation Hua B, Huang Y, Wu Y H, et al. Spacecraft formation reconfiguration trajectory planning with avoidance constraints using adaptive pigeon-inspired optimization. Sci China Inf Sci, 2019, 62(7): 070209, https://doi.org/10.1007/s11432-018-9691-8

Dear editor,
The autonomous formation flight of satellite clusters has been identified as a beneficial technology for many future space missions [1, 2]. The research on spacecraft formation technology has become much more important for ever-increasing space applications. However, the major challenge lies in effective formation path planning to ensure collision avoidance and fuel savings.

For collision avoidance, the collision probability method has been proposed for many years, but it is mostly used for collision warning involving two spacecraft. Chan [3] has developed an approximate method by the summation of infinite series. Sushnigdha and Joshi [4] presented an orthogonal collocation-based entry trajectory solution strategy using pigeon-inspired optimization (PIO). Li and Chou [5] presented a self-adaptive learning particle swarm optimization (PSO) for mobile robot path planning in 2D environments. Zhang and Duan [6] applied the improved Gaussian distribution algorithm to a path planning problem. Yang et al. [7] proposed a cauchy mutation pigeon-inspired optimization (CMPIO) to optimize automatic carrier landing system parameters in each layer.

In this study, an adaptive PIO (APIO) is proposed. The sum of velocity changes of the spacecraft at each discrete path point is used as the fuel consumption index. The collision probability between spacecraft at each discrete point is calculated by the summation of infinite series. The performance of the APIO in path planning for spacecraft formation reconfiguration was analyzed via numerical experiments. The contributions of this study are as follows. (a) The APIO is proposed, improving the searching ability of optimization. (b) The collision probability is introduced in the formation reconfiguration, which is more efficient for collision avoidance than just using the safety distance.

Methodology. Suppose that the initial position of spacecraft in the $C-W$ system is given by $\boldsymbol{r}_0 = [x_0, y_0, z_0]$. The velocity in the $C-W$ system is given by $\boldsymbol{v}_0 = [u_0, v_0, w_0]$. When the effect of gravity is the only consideration, the relative position and velocity in the $C-W$ system after t seconds can be reduced to the following matrix expression:

$$\begin{aligned}\boldsymbol{r}(t) &= \phi_{rr}(t)\boldsymbol{r}_0 + \phi_{rv}(t)\boldsymbol{v}_0,\\ \boldsymbol{v}(t) &= \phi_{vr}(t)\boldsymbol{r}_0 + \phi_{vv}(t)\boldsymbol{v}_0.\end{aligned} \tag{1}$$

The detailed $C-W$ matrices are shown in Appendix A.

The APIO algorithm introduces the map and compass factor that varies with the iterations. The

* Corresponding author (email: 18727465720@163.com)

fitness function consists of the fuel consumption and collision probability. In this problem, each possible solution $\boldsymbol{X}$ is a series of discrete points along the flight path for the formation reconfiguration, where $\boldsymbol{X}_p$ denotes the flight path of the p-th spacecraft.

$$\boldsymbol{X}(k)=\begin{bmatrix}\boldsymbol{X}_1(k)\\ \boldsymbol{X}_2(k)\\ \vdots\\ \boldsymbol{X}_p(k)\end{bmatrix}=\begin{bmatrix}x_1^1(k) & x_1^2(k) & \cdots & x_1^n(k)\\ x_2^1(k) & x_2^2(k) & \cdots & x_2^n(k)\\ \vdots & \vdots & & \vdots\\ x_p^1(k) & x_p^2(k) & \cdots & x_p^n(k)\end{bmatrix}. \quad (2)$$

The APIO algorithm also contains two different iteration operators: the map and compass operator and the landmark operator. When calculating with the map operator, the individual in the pigeon population updates its position and velocity in the solution space according to the optimal individual in the current iteration. The formula is as follows:

$$\begin{cases}\boldsymbol{V}(k)=\boldsymbol{V}(k-1)\mathrm{e}^{-Rk}+[\mathbf{rand}]\cdot(\boldsymbol{X}_b(k-1)\\ \qquad -\boldsymbol{X}(k-1)),\\ \boldsymbol{X}(k+1)=\boldsymbol{X}(k)+\boldsymbol{V}(k),\end{cases} \quad (3)$$

where R is the map and compass factor, k is the number of iterations, $\boldsymbol{X}_b$ is the global optimal solution in current population, and $[\mathbf{rand}]$ represents the vector composed of random numbers from 0 to 1.

After a certain number of iterations for optimization, the individual pigeon will arrive near the optimal solution. Then, the landmark operator is employed. The center point of the population is computed as the landmark to update the positions of individual pigeons, which increases the convergence speed of the algorithm. The updated formula is as follows:

$$\begin{cases}\boldsymbol{C}(k)=\dfrac{\sum_{i=1}^{p}\boldsymbol{X}_i(k)\text{fitness}[\boldsymbol{X}_i(k)]}{\sum_{i=1}^{p}\text{fitness}[\boldsymbol{X}_i(k)]},\\ \boldsymbol{X}(k)=\boldsymbol{X}(k-1)+[\mathbf{rand}]\cdot[\boldsymbol{C}(k)-\boldsymbol{X}(k-1)].\end{cases} \quad (4)$$

In Eq. (4), fitness$(\cdot)$ is the fitness function. In this algorithm, it consists of fuel consumption and collision probability of spacecraft formation. $\boldsymbol{C}(k)$ indicates the position of the center point of the population at the k-th iteration.

In the APIO algorithm, the map and compass factor is improved to a variable which balances the global and local search optimization ability of the algorithm. The improved map and compass factor can be expressed as follows:

$$R=R_{\max}-\left[\frac{N}{N_{\max}}\right]^{\alpha}(R_{\max}-R_{\min}), \quad (5)$$

where $R_{\max}$, $R_{\min}$ are the maximum and minimum values of the map and compass factor setting. N is the iteration number. $N_{\max}$ is the maximum iteration. α is the parameter R speed change factor. Using simplified series in collision probability calculations [3], the collision probability of two spacecraft at a certain moment can be obtained as

$$P=\mathrm{e}^{-(v+u)}\sum_{k=0}^{\infty}\frac{v^k}{k!}\left(\mathrm{e}^{u}-\sum_{j=0}^{k}\frac{u^j}{j!}\right). \quad (6)$$

According to [3], We can only take the first three items as approximate calculation and the truncation error is negligible. v and u are dimensionless parameters, whose formula can be seen in Appendix B. In this way, the expression of the total collision probability of the spacecraft formation flight can be written as follows:

$$P(\boldsymbol{X})=\sum_{i=1}^{p-1}\sum_{j=i+1}^{p}\sum_{k=1}^{n}P_{ijk}(\boldsymbol{X}), \quad (7)$$

where P_{ijk} represents the collision probability of the i-th spacecraft and j-th spacecraft at k-th points along the flight path.

In addition, the APIO combines fuel consumption with collision probability to construct the fitness function using the external penalty function method. The penalty factor σ is introduced, thereby transforming the constraint problem of collision probability into the problem of finding the minimum value without constraint. The penalty function is shown as follows:

$$\widetilde{P}(\boldsymbol{X})=\sigma\left[\min\left(0,\log\frac{P_{\max}}{P(\boldsymbol{X})}\right)\right]^2, \quad (8)$$

where $P(\boldsymbol{X})$ represents the collision probability of the total collision probability in the spacecraft formation flight. $P_{\max}$ is the maximum permissible collision probability.

Assume that there are p spacecraft in the formation. Each spacecraft flight path consists of n points $[x_1,x_2,\ldots,x_n]$. The flight time between neighboring points is T seconds. Then the velocity change of the i-th spacecraft at k-th point is

$$\Delta\boldsymbol{v}_i^k=\boldsymbol{v}_i^{kf}-\boldsymbol{v}_i^{ki}. \quad (9)$$

The total fuel consumption of the formation configuration can be expressed as follows:

$$J(\boldsymbol{X})=\sum_{i=1}^{p}\sum_{k=1}^{n}|\Delta\boldsymbol{v}_i^k|. \quad (10)$$

Therefore, the augmented fitness function is

$$\text{fitness}(\boldsymbol{X})=J(\boldsymbol{X})+\widetilde{P}(\boldsymbol{X}). \quad (11)$$

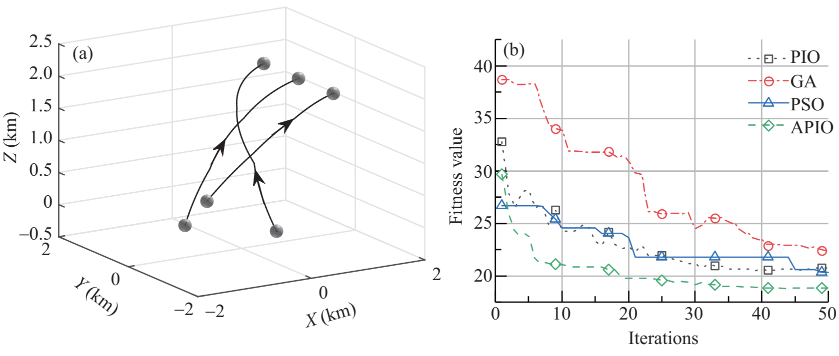

Figure 1 (Color online) (a) Optimized flight path of spacecraft formation; (b) learning curve of the different optimization algorithms.

In order to reduce the complexity of the problem, coordinate transformation is needed. The transformation matrix is presented in Appendix C.

Numerical case. The simulation takes round-orbit spacecraft formation as example. The initial conditions are listed in Tables D1 and D2. Figure 1(a) shows the resulting path by considering the collision avoidance factor. Figure 1(b) shows the result when using different nature-inspired algorithms. It shows that the APIO algorithm has better global searching ability and faster convergence than other algorithms. It should be noted that owing to the penalty factor, only when the collision probability is exactly in the right small neighborhood of $P_{\max}$ will the fitness value include the part caused by collision probability. But this is a very small probability event. Hence, we can take the fitness value as the total fuel consumption. The concrete results under the different discrete points, when the total flight time of the mission was set to 400 s, are presented in Table D3. It shows that the number of selected discrete points will affect the performance of the optimization algorithm.

Conclusion. This study proposes an APIO algorithm to solve the reconfiguration path planning problem of spacecraft formation in round-orbit. The method combines the fuel consumption and collision avoidance factors by using the external penalty function in constructing the fitness function. In the APIO, the parameter of the map and compass factor is set to change with the iterations. The simulation results under different conditions show that when the number of discrete points increase, the optimization characteristics will worsen. When the number of discrete points decrease, the optimization performance improves. However, when the number of discrete points is too small, it will be difficult for the flight path to be approximatively linearized, which will reduce the efficiency of collision avoidance. Therefore, we will improve the collision probability calculation for nonlinear motion and the adaptive change strategy of the algorithm in a future study.

Acknowledgements This work was supported by Fundamental Research Funds for the Central Universities (Grant No. NS2018054).

Supporting information Appendixes A–D. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Yu Q Y, Meng W X, Yang M C, et al. Virtual multi-beamforming for distributed satellite clusters in space information networks. IEEE Wirel Commun, 2016, 23: 95–101
2 Nag S, Gatebe C, Hilker T, et al. Gross primary productivity estimation using multi-angular measurements from small satellite clusters. In: Proceedings of Geoscience and Remote Sensing Symposium, 2014
3 Chan F K. Spacecraft Collision Probability. El Segundo: Aerospace Press, 2008
4 Sushnigdha G, Joshi A. Trajectory design of re-entry vehicles using combined pigeon inspired optimization and orthogonal collocation method. IFAC-PapersOnLine, 2018, 51: 626–662
5 Li G S, Chou W S. Path planning for mobile robot using self-adaptive learning particle swarm optimization. Sci China Inf Sci, 2018, 61: 052204
6 Zhang S J, Duan H B. Gaussian pigeon-inspired optimization approach to orbital spacecraft formation reconfiguration. Chin J Aeronaut, 2015, 28: 200–205
7 Yang Z Y, Duan H B, Fan Y M, et al. Automatic carrier landing system multilayer parameter design based on cauchy mutation pigeon-inspired optimization. Aerospace Sci Tech, 2018, 79: 518–530