Distributed Asynchronous Planning and Task Allocation for Autonomous Cluster Flight of Fractionated Spacecraft

Jing Chu, Jian Guo, Eberhard Gill
Chair of Space Systems Engineering, Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands
Outline

1. Introduction
2. Generalized Formulation of the Problem
3. An Example of the Generalized Formulation
4. Distributed Asynchronous Planning and Task Allocation Algorithm
   - Related Work
   - Core Algorithm
5. Simulation
6. Conclusion
Introduction: Autonomous Cluster Flight

1. Fractionated Spacecraft is a rising star for future space systems due to its non-traditional attributes, such as flexibility, robustness and responsiveness.

2. Autonomous Cluster Flight

- Fly in Open and Dynamic Environment
- Relieve the Workload of Ground Stations
- Reduce Cost after Launching Fractionated Spacecraft
- Form the Virtual Spacecraft
- Perform Unique Operations

Autonomous Cluster Flight

Modules are more intelligent
Introduction: Planning and Task Allocation

3. Organizational Architecture

Planning and task allocation bridge the gap between the top-level layer (interpreting inputs from the environment) and the bottom-level layer (local controllers) of the distributed space system.

4. Cluster Reconfiguration

There are primarily three aspects of the cluster reconfiguration.

1) Generation of the set of new clusters that satisfy the objectives of the input from the environment.
2) Assignment of the allowable position in the generated clusters to the individual module.
3) Reconfiguration maneuver to move a module from its current cluster to the assigned location.

The first one and the third one belong to the planning problem that interacts with the environment, while the second one is the output of the task allocator that generates references for the local controller of each module.
Problem Formulation (1)

1. Cluster
   1) Dynamics of the cluster
      \[ \dot{x}_i = f_i(x_i, w_i, u_i), \quad i = 1, \ldots, n \]  
   2) Solutions of the distance bounded relative motion
      \[ x_i = s_i(\theta_i, t), \quad i = 1, \ldots, n \]  
   3) Description of the cluster
      - Based on solutions
        \[ C = \{C_i \mid C_i = s_i, i = 1, \ldots, n \} \]  
      - Based on allowable positions
        \[ C = \left\{ P^{i_1}_{N_i} \ldots P^{i_q}_{N_i} \ldots P^{j_1}_{N_j} \ldots P^{j_q}_{N_j} \ldots P^{m_1}_{N_m} \ldots P^{m_q}_{N_m} \right\}, \quad 1 \leq m \leq n, \sum_{j=1}^{m} N_j = n \]  
      - Based on parameter vectors
        \[ C = \begin{bmatrix} \theta_{i_1} \\ \vdots \\ \theta_{i_{(k-1)}} \\ \theta_{i_{(k-1)+1}}^{N_i} \\ \vdots \\ \theta_{i_k}^{N_i} \end{bmatrix} \ldots \begin{bmatrix} \theta_{j_1} \\ \vdots \\ \theta_{j_{(k-1)}} \\ \theta_{j_{(k-1)+1}}^{N_j} \\ \vdots \\ \theta_{j_k}^{N_j} \end{bmatrix} \ldots \begin{bmatrix} \theta_{m_1} \\ \vdots \\ \theta_{m_{(k-1)}} \\ \theta_{m_{(k-1)+1}}^{N_m} \\ \vdots \\ \theta_{m_k}^{N_m} \end{bmatrix},\quad 1 \leq m \leq n, \sum_{j=1}^{m} N_j = n \]
2. Planning and Task Allocation

Consider $n$ modules performing operations in a cluster $C_0$. Due to the change of mission objectives, $n' \ (1 \leq n' \leq n)$ modules are required to reconfigure to a new cluster.

1) Cluster candidates

$$\{C_1, ..., C_M\}$$

2) Assignment vector

$$A_i \triangleq [a_{1}, ..., a_{N_i}]^T$$

3) Cost function

$$c_{kj}^{i} = c_{kj}^{i}(C_k, A_{kj}) \quad (6)$$

4) Fuel-optimized planning and task allocation

$$\min_{k} \left( \min_{A_k} \sum_{i=1}^{n'} f_{kj}^{i} \right) \quad (7)$$

5) Time-optimized planning and task allocation

$$\min_{k} \left( \min_{A_k} \sum_{i=1}^{n'} t_{kj}^{i} \right) \quad (8)$$

Note: 1. $N$ is usually set to $n'$

2. When $k$ is fixed, the reconfiguration problem reduces to the low-level planning and task allocation
An Example of Generalized Formulation

1) Distance Bounded Relative Motion

\[
\begin{align*}
\ddot{x} - 2n\dot{y} - 3n^2x &= 0 \\
\dot{y} + 2n\dot{x} &= 0 \\
z + n^2z &= 0
\end{align*}
\]

Define \(d\) as the drift rate along the \(y\) direction. If the drift rate \(d\) is set to zero, then the relative motion is closed; if other constraints are imposed on the drift rate, then distance-bounded relative motion is generated.

2) Parameter Vector

\[ \theta = [R_0, d, y_{off}, B_0, \alpha, \beta]^T \]

3) Initial Cluster and Cluster Candidates for Reconfiguration

\[ C_0 = \begin{bmatrix} R_{0,1} & R_{0,2} & R_{0,3} \\ d_{0,1} & d_{0,2} & d_{0,3} \end{bmatrix}, \quad m = 3, n = 4 \]

There are 24 possible assignment vectors, i.e. 24 choices of \(A_k\), for each cluster candidate. On the other hand, there are 3 choices of the target cluster, i.e. 3 choices of \(k\).
Distributed Planning and Task Allocation
——Related Work

1. Architecture of Existing Algorithms
   1) Centralized
      Pros: global optimization
      Cons: assumption, computation, scalability, single point failure
   2) Distributed
      Pros: communication, computation, scalability, robustness
      Cons: local optimization, conflicts
   3) Mixed
      Cons: communication topology, uneven computation

2. Methodology of Existing Algorithms
   1) Linear Programming (LP) and Extensions
      Pros: efficient, low computational cost
      Cons: iterative, centralized
   2) Auction-Based Algorithm
      Cons: local optimization
   3) Dynamic Programming
      Pros: sequential, distributed
   4) Analytical Approaches
Distributed Planning and Task Allocation
——Core Algorithm

1. General Description

1) The input of the algorithm is the set of all cluster candidates that meet mission objectives, where all the allowable positions are specified by means of the parameter vector. The output of the algorithm is the target cluster and its related assignment vector.

2) The algorithm consists of iterations between two parts. One is the auction algorithm, where the module bids for the allowable positions in the cluster; the other is the consensus algorithm that is used to converge the assignment vector.

3) By iterating between the two parts, the algorithm can exploit the efficiency and robustness of the auction algorithm as well as the distributed convergence properties of the consensus algorithm.

4) The cost induced by the maneuver is calculated onboard each module.
Distributed Planning and Task Allocation  
——Core Algorithm

2. Auction Process

<table>
<thead>
<tr>
<th>Algorithm 1</th>
<th>Auction Process for module $i$ bidding for cluster $k$ at iteration $t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>procedure SelectAllowablePosition ($c_k^i$, $a_k^i(t-1)$, $b_k^i(t-1)$)</td>
</tr>
<tr>
<td>2:</td>
<td>$a_k^i(t) = a_k^i(t-1)$, $b_k^i(t) = b_k^i(t-1)$</td>
</tr>
<tr>
<td>3:</td>
<td>if $\sum_{j=1}^{N_k} a_k^j(t) = 0$ then</td>
</tr>
<tr>
<td>4:</td>
<td>calculate $d_{kj}^i$ ($1 \leq j \leq N_k$)</td>
</tr>
<tr>
<td>5:</td>
<td>if $d_{kj}^i \neq 0$ then</td>
</tr>
<tr>
<td>6:</td>
<td>$J_k^i = \text{argmin}<em>j d</em>{kj}^i \times c_{kj}^i$</td>
</tr>
<tr>
<td>7:</td>
<td>$a_{kj}^i(t) = 1$</td>
</tr>
<tr>
<td>8:</td>
<td>$b_{kj}^i(t) = -c_{kj}^i$</td>
</tr>
<tr>
<td>9:</td>
<td>end if</td>
</tr>
<tr>
<td>10:</td>
<td>end if</td>
</tr>
<tr>
<td>11:</td>
<td>End procedure</td>
</tr>
</tbody>
</table>

Note:
1. The iteration is performed asynchronously, and each module's number of iterations can be different; therefore, the iteration period of each module can be different.
2. The procedure shown in algorithm 1 is only for one cluster, and the procedures for the rest of the cluster candidates are performed by each module concurrently, which results in $2M$ vectors stored and updated onboard each module.

Two Vectors:
1) $a_{kj}^i$: module $i$'s assignment vector for cluster $k$, where $a_{kj}^i = 1$ if module $i$ is assigned to the allowable $j$th position in cluster $k$, and 0 otherwise.
2) the bid list $b_{kj}^i$, where $b_{kj}^i$ is the as up-to-date as possible update of the highest bid for each allowable position in the cluster $k$

Decision Making Variable $d_{kj}^i$:

$$d_{kj}^i = \begin{cases} 
1 & \text{if } c_{kj}^i < -b_{kj}^i \\
0 & \text{if } c_{kj}^i > -b_{kj}^i
\end{cases}, \quad 1 \leq j \leq N_k$$

Best Offer $J_k^i$

$J_k^i$ connects the auction and consensus algorithm.
Distributed Planning and Task Allocation
——Core Algorithm

3. Consensus Process

**Algorithm 2** Consensus Process for module $i$ performing for all cluster candidates at iteration $t$

```plaintext
1: while $k \leq M(k = 1, \ldots, M)$ do
2:     Send $b^p_k(t)$ to module $p$ with $G_{ip}(\tau) = 1$
3:     Receive $b^p_k(t)$ from module $p$ with $G_{ip}(\tau) = 1$
4:     procedure UpdateAssignment($G(\tau)$, $a^i_k(t)$, $b^p_k(t)$, $J_i$)
5:         $b_{kj}^i(t) = \max G_{ip} \times b_{kj}^p$ (1 $\leq j \leq N_k$) searching for all neighbors $p$
6:         $Assignment_{of} J_k^i = \arg\max_p G_{ip}(\tau) \times b_{kj}^p(t)$
7:         if $i \neq Assignment_{of} J_k^i$ then
8:             $a_{kJ_k^i}^i(t) = 0$
9:         end if
10:     end procedure
11: end while
```

Note:
1. The communication topology is described by the adjacency matrix $G(\tau)$, which is defined in such a way that $G_{ij}(\tau) = 1$ if there is a communication link between module $i$ and $j$ at time $\tau$, and 0 otherwise.
2. Module $i$ loses its assignment if it’s outbid by other modules (lines 6,7,8).
1. Architecture of the Simulation System

There is a PC that hosts the main container, which holds three agents, i.e. the ground station (GS) agent performing functionalities of a ground station, the AMS agent providing management services of the multi-agent system and the DF agent providing the yellow page service. In the figure there are three module agents in the fractionated system. Each module agent is hosted by a peripheral container. Two peripheral containers reside on Android smartphones, while the third one on a emulator of the Android smartphone. All three module agents are connected via Wi-Fi network, and it is the same case for the connection between ground station and the fractionated spacecraft.
A simple reconfiguration case with fuel-optimized objective is simulated to show the performance of the algorithm. The initial configuration is shown in the left figure and the expected configuration is an “A-Train” cluster in the along-track direction. In the “A-Train” cluster module 2 is 10km behind module 3 that is also 10km behind module 1. The transfer time is set to 0.745h. The allowable positions of the new cluster is generated by discretizing every 1km along the Y-axis. The propellant cost is calculated based on the theory of two-impulse maneuver.
Conclusion

This paper addressed the need and approaches of autonomous cluster flight for fractionated spacecraft, especially in the context of planning and task allocation.

1. The planning and task allocation problem of autonomous cluster flight is formulated by taking advantage of a generalized methodology, which is applicable to the linear programming and its extensions, dynamic programming, and auction-based algorithms with the fuel-optimized or time-optimized objective.

2. The planning and task allocation algorithm proposed in this paper is distributed (only neighbors’ information is needed), asynchronous (iterations are run locally onboard each module) and the computational load is very low (due to the auction algorithm and the simple consensus law).

3. A hardware-in-the-loop platform is developed, which is based on the multi-agent and smartphone technology. Fruitful research findings in the field of MAS, such as multi-agent optimization, can be tested and incorporated in the platform when applying them to the autonomous cluster flight of fractionated spacecraft.
Thank you!

&

Questions?