On the Scheduling of Systems of Heterogeneous UAVs and Fuel Service Stations for Long-Term Mission Fulfillment

Jonghoe Kim, Byungduk Song and James R. Morrison
Department of Industrial and Systems Engineering
KAIST, South Korea

Thursday, June 14, 2012
Presentation Overview

- Motivation and literature
- System description
- System scheduling: MILP and genetic algorithm
- Numerical example
- Concluding remarks
Motivation

- Large expensive UAVs
  - Operate for many hours
  - Travel long distances

- Small inexpensive UAVs
  - Limited duration of mission
  - Limited distance

- Persistent operation of a system of small UAVs in the field
  - Collection of UAVs
  - Refueling stations
  - Methods to orchestrate their operations

GOAL:
Develop scheduling methods for a system of UAVs and service platforms distributed across the field of operations
Literature Review

- **Scheduling methods without a distance or time restriction**

- **Scheduling methods for limited flight duration**

- **Battery recharge/exchange methods**
Literature Review

• Scheduling of UAVs and service station(s) at a single location
  – B. Bethke, "Persistent Vision-Based Search and Track Using Multiple UAVs," Master Thesis, Massachusetts Institute of Technology, 2005
System Description

Persistent UAV scheduling model with heterogeneous UAVs and *multiple service stations*

- A system of UAVs that is supported by automated replacement systems can support long term or even indefinite duration missions in a near autonomous mode with multiple service stations

- The UAVs can return to any service station, replenish their resources and resume their duties
System Description

To follow a time-space trajectory, the trajectory is divided into pieces (split jobs)

- Objective moves
  - From point \((50,250)\) to \((950,350)\)
  - From 13:10 to 13:20

<table>
<thead>
<tr>
<th>Split job</th>
<th>Start point</th>
<th>End point</th>
<th>Start time</th>
<th>End time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50,250</td>
<td>150,250</td>
<td>13:10</td>
<td>13:11</td>
</tr>
<tr>
<td>2</td>
<td>150,250</td>
<td>250,250</td>
<td>13:11</td>
<td>13:12</td>
</tr>
<tr>
<td>3</td>
<td>250,250</td>
<td>350,250</td>
<td>13:12</td>
<td>13:13</td>
</tr>
<tr>
<td>4</td>
<td>350,250</td>
<td>450,250</td>
<td>13:13</td>
<td>13:14</td>
</tr>
<tr>
<td>5</td>
<td>450,250</td>
<td>550,250</td>
<td>13:14</td>
<td>13:15</td>
</tr>
<tr>
<td>6</td>
<td>550,250</td>
<td>650,250</td>
<td>13:15</td>
<td>13:16</td>
</tr>
<tr>
<td>7</td>
<td>650,250</td>
<td>750,250</td>
<td>13:16</td>
<td>13:17</td>
</tr>
<tr>
<td>8</td>
<td>750,250</td>
<td>850,250</td>
<td>13:17</td>
<td>13:18</td>
</tr>
<tr>
<td>9</td>
<td>850,250</td>
<td>950,250</td>
<td>13:18</td>
<td>13:19</td>
</tr>
<tr>
<td>10</td>
<td>950,250</td>
<td>950,350</td>
<td>13:19</td>
<td>13:20</td>
</tr>
</tbody>
</table>
Mathematical Program to Schedule the System

We proposed a MILP model which allows

- Mobile UAVs to return to service after refueling
- Collection of mobile robots to follow a specific time-space trajectory
- Long-term mission to receive uninterrupted mobile robot service by successively handing off the task to replacement UAVs
Mathematical Program to Schedule the System

Notation

$i, j$ : Indices for jobs  
$s$ : Index for stations  
$k$ : Index for UAVs  
$r$ : Index of a UAV's $r^{th}$ flight  
$N_J$ : Number of split jobs  
$N_{UAV}$ : Number of UAVs in the system  
$N_{STA}$ : Number of recharge stations  
$N_R$ : Maximum number of flight per UAV during the time horizon  
$M$ : Large positive number  
$(x_j^s, y_j^s)$ : Start point of split job $j$  
$(x_j^e, y_j^e)$ : End point of split job $j$  
$D_{ij}$ : Distance from split job $i^{th}$ finish point to split job $j^{th}$ start point, $D_{ij} \neq D_{ji}$  
$E_i$ : Start time of split job $i$  
$P_i$ : Processing time or split job $i$  
$q_k$ : Maximum traveling time of UAV $k$  
$S_{ok}$ : Initial location(station) of UAV $k$  
$TS_k$ : Travel speed of UAV $k$
Mathematical Program to Schedule the System

**Notation**

- \( \Omega_J \) : \( \{1, \ldots, N_J\} \), Set of split jobs
- \( \Omega_{JD} \) : \( \{1, \ldots, N_J+1\} \), Set of split jobs and dummy jobs
- \( \Omega_{SS} \) : \( \{N_J+2, N_J+4, \ldots, N_J+2 \cdot N_{STA}\} \), set of UAV flight start station
- \( \Omega_{SE} \) : \( \{N_J+3, N_J+5, \ldots, N_J+2 \cdot N_{STA}+1\} \), set of UAV flight end station
- \( \Omega_A \) : \( (\Omega_{JD} \cup \Omega_{SS} \cup \Omega_{SE}) = \{1, \ldots, N_J+2 \cdot N_{STA}+1\} \), set of all jobs and recharge stations

**Decision Variables**

- \( X_{ijkr} = 1 \) if UAV \( k \) processes split job \( j \) or recharges at station \( j \) after processing split job \( i \) or recharging at station \( i \) during the \( r \)th flight; 0, otherwise
- \( Y_{ikr} = 1 \) if UAV \( k \) processes split job \( i \) during its \( r \)th flight; 0, otherwise.
- \( C_{ikr} \) is job \( i \)'s start time by UAV \( k \) during its \( r \)th flight or UAV \( k \)'s recharge start time at station \( i \); otherwise its value is 0.
Mathematical Program to Schedule the System

Mathematical formulation

Minimize \[ \sum \sum \sum \sum D_{ij} \cdot X_{jkr} \]

Subject to

\[ \sum_{j \in \Omega_{jo}} X_{s_{ik},jk} = 1 \ (k \in K) \]

Initial recharge station constraints

\[ \sum_{j \in \Omega_{jo}} X_{sjkr} = 1 \ (k \in K, r \in R) \]

Recharge station constraints

\[ \sum_{k \in K} X_{s_{ikr}} = 1 \ (k \in K, r \in R \backslash \mathbb{R}_0) \]

\[ \sum_{i \in \Omega_{jo}} X_{is_{kr}} = \sum_{i \in \Omega_{jo}} X_{s_{i,j}kr+1} \ (k \in K, r = 1 \ldots N_R - 1, s \in \Omega_{SE}) \]

\[ C_{s_{kr}} = C_{s_{i,j}kr+1} \ (k \in K, r = 1 \ldots N_R - 1, s \in \Omega_{SE}) \]

Split job assignment constraints

\[ \sum_{i \in \Omega_{jo}} X_{ijkr} = 1 \ (j \in \Omega_{j}) \]

\[ \sum_{j \in \Omega_{jo}} X_{ijkr} - \sum_{j \in \Omega_{jo}} X_{jikr} = 0 \ (i \in \Omega_{JD}, k \in K, r \in R) \]

Mathematical Program to Schedule the System

Mathematical formulation

\[
C_{ikr} + P_i + D_{ij} / TS_k - C_{jkr} \leq M (1 - X_{ijk}) \quad (i \in \Omega_{JD} \cup \Omega_{SS}, j \in \Omega_{JD} \cup \Omega_{SE}, k \in K, r \in R)
\]

\[
\sum_{j \in \Omega_{JD} \cup \Omega_{SE}} X_{ijk} \geq Y_{ikr} \quad (i \in \Omega_{j}, k \in K, r \in R)
\]

\[
M \cdot Y_{ikr} \geq C_{ikr} \quad (i \in \Omega_{j}, k \in K, r \in R)
\]

\[
\sum_{k \in K} \sum_{r \in R} C_{ikr} = E_{ij} \quad (i \in \Omega_{j})
\]

\[
\sum_{i \in \Omega_{A}} \sum_{j \in \Omega_{A}} D_{ij} / TS_k \cdot X_{ijk} + \sum_{i \in \Omega_{JD}} \sum_{j \in \Omega_{A}} P_i \cdot X_{ijk} \leq q_k \quad (k \in K, r \in R)
\]

\[
X_{sdr} = X_{d,s+1,kr} \quad (k \in K, r \in R, s \in \Omega_{SS})
\]

\[
X_{dikr} + X_{sdr} = 0 \quad (k \in K, r \in R, i \in \Omega_{j})
\]

\[
C_{ikr} \geq 0 \quad (k \in K, r \in R, i \in \Omega_{A})
\]

\[
X_{ijk} \in \{0,1\} \quad (k \in K, r \in R, i \in \Omega_{A}, j \in \Omega_{A})
\]

\[
Y_{ikr} \in \{0,1\} \quad (k \in K, r \in R, i \in \Omega_{A})
\]
Genetic Algorithm

Genetic algorithm to overcome the computational limit of CPLEX and allow the potential for near real-time use

- Chromosome Structure
  - 1 rows, length of each chromosome is # of split jobs

- Example: 100 split jobs and 4 UAVs

Random Assignment

Randomly generate between 1 and # of UAVs

Node 1 2 3 4 5 6 7 8 · · · · · · · 99 100

Random Assignment

UAV number
Genetic Algorithm

<table>
<thead>
<tr>
<th>Node</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>· · ·</th>
<th>· · ·</th>
<th>· · ·</th>
<th>99</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Assignment</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>· · ·</td>
<td>· · ·</td>
<td>· · ·</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Will be divided like this

<table>
<thead>
<tr>
<th>UAV 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 5 15 21 24 34 54 · · · 75 82 87 100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UAV 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 6 12 27 37 55 67 · · · 73 84 92 97</td>
</tr>
</tbody>
</table>
Genetic Algorithm

Rearrange each UAV’s assigned split jobs in increasing order of job start time

UAV 1

| 87 | 82 | 54 | 75 | 24 | 34 | 21 | ... | ... | ... | 100 | 1 | 15 | 5 |

UAV 4

| 12 | 6 | 27 | 4 | 37 | 73 | 55 | ... | ... | ... | ... | 92 | 97 | 84 | 67 |

Each array represents the split job sequence of each UAV
Genetic Algorithm

■ Service station assignment

- Heuristic to determine when a UAV will visit a service station

- After each assigned split job in the chromosome, UAV goes to its next split job or to a station

- If there is not enough energy remaining to conduct the next task and reach a station, UAV goes to next station which will give minimum distance from current location to next job

- If there is enough energy, UAV conducts the next split job
Genetic Algorithm

- Fitness function

  - Consists of 2 parts, which represent the obj. value and feasibility of the MILP formulation

  1) Calculate the total traveling distance of UAVs

  2) Calculate the penalty which comes from
     ① violation of job start time
     ② violation of battery capacity
Numerical example

Example 1: Tracking a Moving Ground Target

<table>
<thead>
<tr>
<th>Split job</th>
<th>Start point</th>
<th>End point</th>
<th>Start time</th>
<th>End time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50,250</td>
<td>150,250</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>150,250</td>
<td>250,250</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>250,250</td>
<td>350,250</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>350,250</td>
<td>450,250</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>450,250</td>
<td>550,250</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>550,250</td>
<td>650,250</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>650,250</td>
<td>750,250</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>750,250</td>
<td>850,250</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>850,250</td>
<td>950,250</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>950,250</td>
<td>950,350</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>950,350</td>
<td>950,450</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>12</td>
<td>950,450</td>
<td>950,550</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>13</td>
<td>950,550</td>
<td>950,650</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>14</td>
<td>950,650</td>
<td>950,750</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>
Numerical example

Example 1: Tracking a Moving Ground Target
- It is sufficient to solve by using CPLEX (171 sec)

<table>
<thead>
<tr>
<th>UAV</th>
<th>Travel Speed</th>
<th>Maximum Traveling Time</th>
<th>Start Station</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100m / min</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>100m / min</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>100m / min</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>100m / min</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>150m / min</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>150m / min</td>
<td>13</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UAV</th>
<th>Split jobs served</th>
<th>Start station</th>
<th>End station</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>4</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>5</td>
<td>2,3,4,5,6,7</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>8,9,10,11,12,13,14</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Total traveling distance: 1846 m
Numerical example

Example 1: Tracking a Moving Ground Target

1 2 3 4 5 6 7 8 9 10
(50,250) (150,250) (250,250) (350,250) (450,250) (550,250) (650,250) (750,250) (850,250) (950,250)

11 12 13 14
(950,350) (950,450) (950,550) (950,650) (950,750)

(250,550) (650,550)
Numerical example

Example 1: Tracking a Moving Ground Target
Example 1: Tracking a Moving Ground Target

Numerical example
Numerical example

Example 1: Tracking a Moving Ground Target

(50,250) (150,250) (250,250) (350,250) (450,250) (550,250) (650,250) (750,250) (850,250) (950,250)


(750,150) (950,750)
Numerical example

Example 1: Tracking a Moving Ground Target
Example 1: Tracking a Moving Ground Target
Numerical example

Example 1: Tracking a Moving Ground Target
Numerical example

- Example 1: Tracking a Moving Ground Target

![Diagram showing tracking of a moving ground target with coordinates and nodes labeled 1 to 14.]
Numerical example

Example 1: Tracking a Moving Ground Target
Numerical example

Example 1: Tracking a Moving Ground Target

```
(50,250)  (150,250)  (250,250)  (350,250)  (450,250)  (550,250)  (650,250)  (750,250)  (850,250)  (950,250)
(250,550)  (650,550)  (750,150)  (950,350)  (950,450)  (950,550)  (950,650)  (950,750)
(50,250)  (150,250)  (250,250)  (350,250)  (450,250)  (550,250)  (650,250)  (750,250)  (850,250)  (950,250)
```

Numerical example

Example 1: Tracking a Moving Ground Target
Numerical example

Example 1: Tracking a Moving Ground Target
Numerical example

Example 1: Tracking a Moving Ground Target
Example 1: Tracking a Moving Ground Target

1. (50, 250)
2. (150, 250)
3. (250, 250)
4. (350, 250)
5. (450, 250)
6. (550, 250)
7. (650, 250)
8. (750, 250)
9. (850, 250)
10. (950, 250)
11. (950, 350)
12. (950, 450)
13. (950, 550)
14. (950, 650)

(750, 150)
Numerical example

Example 1: Tracking a Moving Ground Target
Numerical example

Example 1: Tracking a Moving Ground Target

[Diagram showing tracking of a moving target with coordinates and paths]
Numerical example

Example 1: Tracking a Moving Ground Target
Numerical example

Example 1: Tracking a Moving Ground Target

(250,550) (650,550)
(50,250) (150,250) (250,250) (350,250) (450,250) (550,250) (650,250) (750,250) (850,250) (950,250)

(750,150) (950,350) (950,450) (950,550) (950,650) (950,750)

Example 1: Tracking a Moving Ground Target

1. (50, 250)
2. (150, 250)
3. (250, 250)
4. (350, 250)
5. (450, 250)
6. (550, 250)
7. (650, 250)
8. (750, 250)
9. (850, 250)
10. (950, 250)

11. (950, 350)
12. (950, 450)
13. (950, 550)
14. (950, 750)

Example:

- Target position:
  - (750, 150)
  - (950, 350)

- Sensor positions:
  - (250, 550)
  - (650, 550)
  - (750, 150)
Numerical example

Example 1: Tracking a Moving Ground Target
Numerical example

Example 1: Tracking a Moving Ground Target

(250,550)  (650,550)
(50,250)    (250,250)    (350,250)    (450,250)    (550,250)    (650,250)    (750,250)    (850,250)    (950,250)
(750,150)
Numerical example

Example 1: Tracking a Moving Ground Target

(50,250) (150,250) (250,250) (350,250) (450,250) (550,250) (650,250) (750,250) (850,250) (950,250)
Example 2: UAV Border Patrol

- It is sufficient to solve by using CPLEX (2 sec)

<table>
<thead>
<tr>
<th>Split job</th>
<th>Start point</th>
<th>End point</th>
<th>Start time</th>
<th>End time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>300,900</td>
<td>0,900</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>0,900</td>
<td>0,600</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>0,300</td>
<td>0,0</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>0,0</td>
<td>300,0</td>
<td>20</td>
<td>22</td>
</tr>
<tr>
<td>5</td>
<td>600,0</td>
<td>900,0</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>900,0</td>
<td>900,300</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>900,600</td>
<td>900,900</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>900,900</td>
<td>600,900</td>
<td>20</td>
<td>22</td>
</tr>
</tbody>
</table>
### Numerical example

#### Example 2: UAV Border Patrol

<table>
<thead>
<tr>
<th>UAV</th>
<th>Travel Speed</th>
<th>Start Station</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>150m / min</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>150m / min</td>
<td>2</td>
</tr>
</tbody>
</table>

#### First Flight

<table>
<thead>
<tr>
<th>UAV</th>
<th>Serving job</th>
<th>Start station</th>
<th>End station</th>
<th>$q_k$ (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,2,3,4</td>
<td>1</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>5,6,7,8</td>
<td>2</td>
<td>1</td>
<td>15</td>
</tr>
</tbody>
</table>

**Obj. value = 2048 m**

#### Second Flight

<table>
<thead>
<tr>
<th>UAV</th>
<th>Serving job</th>
<th>Start station</th>
<th>End station</th>
<th>$q_k$ (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3,4</td>
<td>1</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>7,8</td>
<td>2</td>
<td>1</td>
<td>14</td>
</tr>
</tbody>
</table>

**Obj. value = 2896 m**
Numerical example

Example 3: practical use of GA for UAV Border Patrol

- It is not sufficient to solve by using CPLEX

- Two Patrols are processed in a counterclockwise direction
- Patrols start at time 10 minutes
- Four cases tested with various patrol distances (2000m, 4000m, 6000m and 8000m for each patrol path)
- 10, 20, 30 and 40 split jobs having 2 minute process time to describe 2000m, 4000m, 6000m and 8000m patrol
Numerical example

Example 3: Practical use of GA for UAV Border Patrol

<table>
<thead>
<tr>
<th>UAV</th>
<th>Travel Speed</th>
<th>Start Station</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200m/min</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>200m/min</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>200m/min</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>200m/min</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>200m/min</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>200m/min</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>200m/min</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>200m/min</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>qk</th>
<th># of split job</th>
<th>CPU time (sec)</th>
<th>Obj. Value</th>
<th>CPU time (sec)</th>
<th>Result</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>20</td>
<td>N/A</td>
<td>1696</td>
<td>4.142</td>
<td>2088.4</td>
<td>292.7</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>N/A</td>
<td>N/A</td>
<td>2.090</td>
<td>3279.1</td>
<td>328.1</td>
</tr>
<tr>
<td>20</td>
<td>40</td>
<td>N/A</td>
<td>N/A</td>
<td>4.067</td>
<td>5169.1</td>
<td>418.4</td>
</tr>
<tr>
<td>20</td>
<td>60</td>
<td>N/A</td>
<td>N/A</td>
<td>6.440</td>
<td>13936</td>
<td>1211.8</td>
</tr>
<tr>
<td>20</td>
<td>80</td>
<td>N/A</td>
<td>N/A</td>
<td>9.295</td>
<td>23112.1</td>
<td>1733.2</td>
</tr>
</tbody>
</table>
Concluding Remarks

• Fleets of UAVs conducting missions over a field of operations can achieve persistence
  – Supported by shared service stations distributed over the field
  – Supported by decision methods

• MILP scheduling model
  – Allows a mobile robot to return to the field following a visit to any shared base for energy resupply
  – UAVs must follow a specific time-space trajectory
  – The presence of any one at each time of the mission is sufficient.

• Develop a genetic algorithm to overcome computational limit of MILP formulation

• Examples

• Future directions
  – System design
  – Real time approaches and decomposition methods