Distributed Scheduling Using Constraint Optimization and Multiagent Path Planning

Christopher T. Cannon\textsuperscript{1}, Robert N. Lass\textsuperscript{1}, Evan A. Sultanik\textsuperscript{1}, William C. Regli\textsuperscript{1}, David Šišlák\textsuperscript{2}, and Michal Pěchouček\textsuperscript{2}

\textsuperscript{1} Department of Computer Science, College of Engineering
Drexel University, Philadelphia, PA, USA

\textsuperscript{2} Agent Technology Center, Faculty of Electrical Engineering
Czech Technical University in Prague, Prague, CZ

12th International Workshop on Distributed Constraint Reasoning
11 May 2010
Problem? Solutions to the distributed scheduling problem (assigning \( n \) workers to \( m \) tasks at time points) only consider worker-task assignment or space deconfliction.

Why? Distributed scheduling problems often occur in three-dimensional continuous environments where workers must be assigned tasks and then must physically travel to that task.

Solution? An approach which first uses distributed constraint optimization to assign workers to tasks and then uses a distributed multiagent path planner to create a path from the worker to its tasks.
Outline

1. Motivating Example
2. Technical Approach
3. Experiments
4. Conclusions & Future Work
Outline

1. Motivating Example
2. Technical Approach
3. Experiments
4. Conclusions & Future Work
Outline

1. Motivating Example
2. Technical Approach
3. Experiments
4. Conclusions & Future Work
Outline

1 Motivating Example

2 Technical Approach

3 Experiments

4 Conclusions & Future Work
The problem consists of:

- A set of agents;
- each equipped with Doppler radar sensors;
- each sensor has three sectors;
- allowed one active sector at any given time;
- communicating over an ad-hoc network;
- tracking moving targets; and
- target must lie within at least three sensors for accurate tracking.
Motivating Example: Unmanned Aerial Vehicle Surveillance

The problem consists of:

- A set of UAVs (agents);
- each equipped with a camera sensor;
- assigned to monitor a subset of the enemy targets;
- communicating over a wireless network; and
- the goal is to minimize the amount of time between a UAV surveilling a target.
Motivating Example: Unmanned Aerial Vehicle Surveillance

The problem consists of:

- A set of UAVs (agents);
- each equipped with a camera sensor;
- assigned to monitor a subset of the enemy targets;
- communicating over a wireless network; and
- the goal is to minimize the amount of time between a UAV surveilling a target.

Traditional Approach

Traditionally, a DisCOP approach focuses on selection (ignoring path creation) and a multiagent planning approach focuses on path creation (ignoring selection).

Cannon et al. (DU & CTU)
Motivating Example: Unmanned Aerial Vehicle Surveillance

The problem consists of:

- A set of UAVs (agents);
- each equipped with a camera sensor;
- assigned to monitor a subset of the enemy targets;
- communicating over a wireless network; and
- the goal is to minimize the amount of time between a UAV surveilling a target.

Traditional Approach

Traditionally, a DisCOP approach focuses on selection (ignoring path creation) and a multiagent planning approach focuses on path creation (ignoring selection).

Cannon et al. (DU & CTU)
Motivating Example: Unmanned Aerial Vehicle Surveillance

The problem consists of:

- A set of UAVs (agents);
- each equipped with a camera sensor;
- assigned to monitor a subset of the enemy targets;
- communicating over a wireless network; and
- the goal is to minimize the amount of time between a UAV surveilling a target.
Motivating Example: Unmanned Aerial Vehicle Surveillance

The problem consists of:

- A set of UAVs (agents);
- each equipped with a camera sensor;
- assigned to monitor a subset of the enemy targets;
- communicating over a wireless network; and
- the goal is to minimize the amount of time between a UAV surveilling a target.

![Diagram of time intervals and gap between them, with labels a1 and a2.]
Motivating Example: Unmanned Aerial Vehicle Surveillance

The problem consists of:

- A set of UAVs (agents);
- each equipped with a camera sensor;
- assigned to monitor a subset of the enemy targets;
- communicating over a wireless network; and
- the goal is to minimize the amount of time between a UAV surveilling a target.

Traditional Approach

Traditionally, a DisCOP approach focuses on selection (ignoring path creation) and a multiagent planning approach focuses on path creation (ignoring selection).
Outline

1. Motivating Example
2. Technical Approach
3. Experiments
4. Conclusions & Future Work
The algorithm consists of four steps:

1. The DisCOP agent selects the target assignments;
2. the TSP approximation algorithm orders the selected targets based upon the current position;
3. the AA* path planner creates a conflict-free flight plan; and
4. after the UAV completes its flight path, the algorithm restarts.
Step 1: DisCOP Agent

DisCOP Agent

Selected Targets

TSP Approx.

Ordered Sequence

Path Planner

Re-solve

Flight Plan
There are four components to a DisCOP:

1. A set of *agents* \( A = \{a_1, a_2, \ldots, a_n\} \);
2. a set of *variables* \( V = \{v_1, v_2, \ldots, v_{|V|}\} \);
3. a set of *domains* that contain the values that may be assigned to said variables \( D = \{D_1, D_2, \ldots, D_{|V|}\} \); and
4. a set of *constraints* over the variable’s assignments.

The objective is to have the agents assign values to their variables such that some metric over the resulting constraints’ values is either minimized or maximized.
Distributed Stochastic Search (DSA)

The DSA family of algorithms generally follows these steps:

1. Selects random values for the agents' own variables;
2. Enters a loop which checks for new neighbor messages;
3. Stochastically decides whether to update its values based on the received messages;
4. Sends the updated values to its neighbors; and
5. Ends when a solution is requested or a terminating condition is met.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$\Delta &gt; 0$</th>
<th>Conflict</th>
<th>No Conflict</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSA-A</td>
<td>$v \text{ with } p$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>DSA-B</td>
<td>$v \text{ with } p$</td>
<td>$v \text{ with } p$</td>
<td>$-$</td>
</tr>
<tr>
<td>DSA-C</td>
<td>$v \text{ with } p$</td>
<td>$v \text{ with } p$</td>
<td>$v \text{ with } p$</td>
</tr>
<tr>
<td>DSA-D</td>
<td>$v$</td>
<td>$v \text{ with } p$</td>
<td>$-$</td>
</tr>
<tr>
<td>DSA-E</td>
<td>$v$</td>
<td>$v \text{ with } p$</td>
<td>$v \text{ with } p$</td>
</tr>
</tbody>
</table>

An incomplete algorithm (e.g., DSA-B) was chosen over a complete algorithm (e.g., ADOPT, DPOP) because of its:

1. Computational and memory cost;
2. Any-time properties; and
3. Fault tolerance.

The mapping is as follows:

- **Agents**: Each UAV agent corresponds to a DisCOP agent;
- **Variables**: each agent has a set of variables which contains a single variable for each constrained target;
- **Domain**: the domain for each variable is boolean (Covered, Not Covered); and
- **Constraints**: the cost for each target is as follows:
  - Low Cost (more than one agent): the number of agents constrained with the target;
  - High Cost (no agents): twice the number of agents; and
  - No Cost (one agent): zero.
The mapping is as follows:

- **Agents**: Each UAV agent corresponds to a DisCOP agent;
- **Variables**: each agent has a set of variables which contains a single variable for each constrained target;
- **Domain**: the domain for each variable is boolean (Covered, Not Covered); and
- **Constraints**: the cost for each target is as follows:
  - Low Cost (more than one agent): the number of agents constrained with the target;
  - High Cost (no agents): twice the number of agents; and
  - No Cost (one agent): zero.

\[
\begin{array}{c|c|c|c|c}
\text{ } & t_1 & t_2 & t_3 & \text{Total} \\
\hline
a_1 & \{t_1=\text{true}, t_2=\text{true}\} & \{t_3=\text{true}, t_2=\text{true}\} \\
\hline
0 & 2 & 0 & 2 \\
\end{array}
\]
DisCOP to UAV Surveillance Assignment Mapping

The mapping is as follows:

- **Agents**: Each UAV agent corresponds to a DisCOP agent;
- **Variables**: each agent has a set of variables which contains a single variable for each constrained target;
- **Domain**: the domain for each variable is boolean (Covered, Not Covered); and
- **Constraints**: the cost for each target is as follows:
  - Low Cost (more than one agent): the number of agents constrained with the target;
  - High Cost (no agents): twice the number of agents; and
  - No Cost (one agent): zero.

\[
\begin{array}{c|c|c|c}
  & t_1 & t_2 & t_3 \\
  \hline
  a_1 & 0 & 0 & 0 \\
  a_2 & 4 & 0 & 0 \\
  \hline
  \text{Total} & 4 & 4 & 0 \\
\end{array}
\]
DisCOP to UAV Surveillance Assignment Mapping

The mapping is as follows:

- **Agents**: Each UAV agent corresponds to a DisCOP agent;
- **Variables**: each agent has a set of variables which contains a single variable for each constrained target;
- **Domain**: the domain for each variable is boolean (Covered, Not Covered); and
- **Constraints**: the cost for each target is as follows:
  - Low Cost (more than one agent): the number of agents constrained with the target;
  - High Cost (no agents): twice the number of agents; and
  - No Cost (one agent): zero.

\[
\begin{array}{c|c|c|c|c}
 & t_1 & t_2 & t_3 & \text{Total} \\
\hline
 a_1 & 0 & 0 & 0 & 0 \\
 a_2 & \{t_1=true, t_2=true\} & \{t_3=true, t_2=false\} \\
\end{array}
\]
Step 2: TSP Approximation

DisCOP Agent → TSP Approx. → Path Planner → Flight Plan

- Selected Targets
- Re-solve
- Ordered Sequence

Cannon et al. (DU & CTU) Distributed Scheduling DCR 2010 05-11
Traveling Salesman Problem (TSP) Approximation

As input the path planner requires an ordered sequence of waypoints.

The problem of finding the shortest path between a set of targets is equivalent to the TSP, which is \( \text{NP} \)-complete.

The 2-Opt exchange algorithm approximates the shortest path by removing the crossing edges in a graph.
Step 3: Path Planner

DisCOP Agent \(\rightarrow\) Selected Targets \(\rightarrow\) TSP Approx. \(\rightarrow\) Ordered Sequence \(\rightarrow\) Path Planner

Path Planner \(\rightarrow\) Flight Plan

Cannon et al. (DU & CTU) Distributed Scheduling DCR 2010 05-11 15 / 26
AA* is an A* based distributed multiagent path planning algorithm which is:

- Completely decentralized;
- utilizes adaptive sampling to remove the trade-off between speed and search precision; and
- progressively smooths the flight path through a set of waypoints.

Step 4: Re-Solve

DisCOP Agent

Selected Targets

TSP Approx.

Ordered Sequence

Re-solve

Ordered Sequence

Path Planner

Flight Plan

Cannon et al. (DU & CTU)

Distributed Scheduling

DCR 2010 05-11

17 / 26
Outline

1. Motivating Example
2. Technical Approach
3. Experiments
4. Conclusions & Future Work
Experimental Setup

(a) Uniform targets.  
(b) Two-cluster targets.  
(c) Random targets.
Experimental Parameters

Most UAV surveillance algorithms rely upon a central decision maker and have no concept of privacy.

The two distributed algorithms we compare our algorithm against are:

1. **Distributed Greedy Cover Set**: Each UAV selects the closest target in its group and de-conflicts plan assignments with identification numbers; and

2. **Random**: Each UAV randomly selects targets to monitor in its group.
Experimental Testbed: AGENTFLY

Experimental Testbed: AGENTFLY

Flight Paths

Experimental Testbed: AGENTFLY

On average, our algorithm produces plans with 38% fewer overlapping target assignments than distributed greedy set cover.

This is ideal as it reduces the strain on AA* to de-conflict flight paths.

<table>
<thead>
<tr>
<th></th>
<th>Uniform</th>
<th>Two-Cluster</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overlap</td>
<td>Excluded</td>
<td>Overlap</td>
</tr>
<tr>
<td>DSA-B</td>
<td>4.63 ± 1.56</td>
<td>2.13 ± 1.31</td>
<td>4.82 ± 1.86</td>
</tr>
<tr>
<td>Greedy</td>
<td>6.87 ± 3.99</td>
<td>0</td>
<td>8.16 ± 3.55</td>
</tr>
<tr>
<td>Random</td>
<td>4.82 ± 1.39</td>
<td>1.88 ± 0.88</td>
<td>4.94 ± 1.18</td>
</tr>
</tbody>
</table>

Table: Average number of targets with overlapping and excluded assignments.
**Analysis: Gap Duration**

- However, distributed greedy set cover performs marginally better than our algorithm in average gap duration.
- In both metrics, random has a high variance, which does not make it a tractable solution.

<table>
<thead>
<tr>
<th></th>
<th>Uniform</th>
<th></th>
<th></th>
<th>Two-Cluster</th>
<th></th>
<th></th>
<th></th>
<th>Random</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exclusive</td>
<td>Shared</td>
<td>All</td>
<td>Exclusive</td>
<td>Shared</td>
<td>All</td>
<td>Exclusive</td>
<td>Shared</td>
<td>All</td>
<td>Exclusive</td>
</tr>
<tr>
<td>DSA-B</td>
<td>3.8 ± 1.1</td>
<td>1.9 ± 0.2</td>
<td>3.2 ± 1.3</td>
<td>4.2 ± 1.9</td>
<td>2.3 ± 0.6</td>
<td>3.6 ± 1.8</td>
<td>4.9 ± 1.4</td>
<td>3.7 ± 1.5</td>
<td>4.5 ± 1.5</td>
<td></td>
</tr>
<tr>
<td>Greedy</td>
<td>2.7 ± 0.6</td>
<td>1.7 ± 0.2</td>
<td>2.4 ± 0.7</td>
<td>2.4 ± 0.6</td>
<td>1.9 ± 0.3</td>
<td>2.2 ± 0.5</td>
<td>4.8 ± 1.3</td>
<td>3.6 ± 1.0</td>
<td>4.4 ± 1.3</td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>4.2 ± 1.4</td>
<td>2.0 ± 0.5</td>
<td>3.5 ± 1.6</td>
<td>4.0 ± 1.6</td>
<td>2.5 ± 1.0</td>
<td>3.5 ± 1.6</td>
<td>9.2 ± 2.9</td>
<td>5.4 ± 2.4</td>
<td>7.9 ± 3.3</td>
<td></td>
</tr>
</tbody>
</table>

**Table:** Average gap duration of unwatched target rounds in minutes.
Outline

1. Motivating Example
2. Technical Approach
3. Experiments
4. Conclusions & Future Work
Conclusions

The contributions of this work are:

1. A method of using DisCOP and distributed multiagent path planning to solve the distributed scheduling problem;
2. a formalization of this method mapped to the UAV surveillance scenario; and
3. evidence that this approach does reduce the amount of plan deconfliction by limiting the amount of overlapping target assignments.

Future work will consist of:

- Experimenting with different DisCOP algorithms (e.g., a variant of DSA which remembers the best global solution); and
- altering the DisCOP constraints (e.g., including the distance to decrease flight path length).
Thank you for your time and attention.

Christopher T. Cannon  
ctc82@cs.drexel.edu  
http://cs.drexel.edu/~ctc82/

Robert N. Lass  
urlass@cs.drexel.edu  
http://cs.drexel.edu/~urlass/

Evan A. Sultanik  
eas28@cs.drexel.edu  
http://cs.drexel.edu/~eas28/

William C. Regli  
regli@cs.drexel.edu  
http://cs.drexel.edu/~regli/

David Šišlák  
sislakd@fel.cvut.cz  
http://agents.felk.cvut.cz/

Michal Pěchouček  
pechoucek@agents.felk.cvut.cz  
http://agents.felk.cvut.cz/