A Mixed Integer Programming Model for Timed Deliveries in Multirobot Systems

Nitin Kamra and Nora Ayanian
Department of Computer Science
University of Southern California
Los Angeles, CA, USA

This work was partially supported by: (a) Viterbi Graduate School Ph.D. Fellowship, and (b) Office of Naval Research grant N00014-14-1-0734.
Robots need resources (power, sensors, actuators) for persistent autonomy.
Objective
Objective

Specifications:
• Delivery robots: Limited on-board power and carrying capacity.
• Priorities over task robots.

Goal:
• Solve the resource delivery problem with timed requests, optimally.
• Minimize total distance traveled and deviation from delivery times.
Existing Approaches to Persistent Autonomy

**Persistent Autonomy**
- Kim & Morrison, JIRS, 2014
  (Persistent operation scheduling for UAVs)
- Song et al. ICUAS, 2013
  (Scheduling for persistent UAV service)
- Smith et al. ICRA, 2011
  (Monitoring in dynamic environments)

**Energy-aware systems**
- Derenick et al. IROS, 2011
  (Energy aware coverage with Docking)
- Kannan et al. ICRA, 2013
  (Autonomous recharging, market-based soln.)
- Mathew et al. ICRA, 2013
  (Multi-robot rendezvous for recharging)

**Swappable Batteries**
- Suzuki et al. JIRS, 2012
  (Design and analysis of battery swapping system)
- Swieringa et al. ICRA 2010
  (Automatic battery swapping for UAVs)
Modeling Paradigms: Delivery Problems

**Stochastic Modeling with Queueing Theory:**

- Bopardikar *et al.* T-RO, 2014
  (Dynamic VRP with time constraints)

- Smith *et al.* CDC, 2008
  (Dynamic VRP with heterogeneous demands)

- Smith *et al.* SIAM J Control Optim, 2010
  (Dynamic VRP, priority classes of demands)

**Mixed-Integer based formulations:**

- Karaman & Frazzoli, IJ Robust Nonlin, 2011
  (LTL Vehicle routing; appl. to multi-uav mission planning)

- Mathew *et al.* WAFR, 2014
  (Path planning, multi-robot delivery systems)

- Stump & Michael, IEEE CASE, 2011
  (Multi-robot persistent surveillance as vehicle routing)

Impose probability distributions on arrival rates and locations of requests.

Stochastic analysis of policies to serve stochastic requests.

Express the objective algebraically.

Guarantees of optimality.

Easier to impose constraints.

Close to VRPTW Formulation.
MIP: Arc vs. Path based approach

Arc-based approach
\[ O(m \cdot n^2) \] variables

Path-based approach
\[ O(m \cdot n!) \] variables
Traveling Salesman Problem - ILP Formulation

\[
\min_x \sum_{i \neq j} c_{ij} x_{ij}
\]

subject to the constraints:

- \( \sum_{j \in V - \{i\}} x_{ij} = 1, \quad \forall i \in V \)  
  Exit each node once

- \( \sum_{i \in V - \{j\}} x_{ij} = 1, \quad \forall j \in V \)  
  Enter each node once

- \( \sum_{i \in S, j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subset V, 2 \leq |S| \leq |V| - 2 \)  
  Eliminate Subtours

- \( x_{ij} \in \{0, 1\}, \quad \forall i, j \in V \)
Vehicle Routing Problem - ILP Formulation

\[
\min_x \sum_{k \in K} \sum_{(i,j) \in E} x_{ij}^k t_{ij}
\]

subject to the constraints:

\[
\sum_{k \in K} \sum_{j \in V \cup \{\omega\}} x_{ij}^k = 1, \quad \forall i \in V
\]

\[
x_{\alpha k}^k = 1, \quad \forall k \in K
\]

\[
\sum_{i \in V} x_{i\omega}^k = 1, \quad \forall k \in K
\]

“Start” and “End” nodes are visited by all robots

\[
\sum_{i \in \{\alpha\} \cup V} x_{ih}^k = \sum_{j \in (V \setminus K) \cup \{\omega\}} x_{hj}^k, \quad \forall h \in V, k \in K
\]

Subtour Elimination Constraints
Vehicle Routing Problem with Capacity Constraints

\[
\min_x \sum_{k \in K} \sum_{(i,j) \in E} x_{ij}^k t_{ij}
\]

subject to the constraints:

\[
\sum_{k \in K} \sum_{j \in V \cup \{\omega\}} x_{ij}^k = 1, \quad \forall i \in V
\]

\[
x_{\alpha k}^k = 1, \quad \forall k \in K
\]

\[
\sum_{i \in V} x_{i\omega}^k = 1, \quad \forall k \in K
\]

\[
\sum_{i \in \{\alpha\} \cup V} x_{ih}^k = \sum_{j \in (V-K) \cup \{\omega\}} x_{hj}^k, \quad \forall h \in V, k \in K
\]

Subtour Elimination Constraints

\[
\sum_{(i,j) \in E} x_{ij}^k - 1 \leq C^k - c^k, \quad \forall k \in K
\]

\[
\sum_{(i,j) \in E} x_{ij}^k B_r^k(t_{ij} v) \leq B^k, \quad \forall k \in K
\]

Delivery Capacity

Total Battery Power
Timed Deliveries

Problem:
- Requests have an associated delivery time.

Solution:
- Impose arrival and departure times at each delivery site.
- Implicitly replace subtour elimination constraints.

\[
a_i - d_i + T_s \leq 0, \quad \forall i \in (V - K)
\]

\[
d_i - a_j + t_{ij} \leq Z \left(1 - \sum_{k \in K} x^k_{ij}\right), \quad \forall (i, j) \in E, j \neq \omega
\]

\[
T_{\text{start}} \leq a_i \leq T_{\text{start}} + t_{\text{bound}}, \quad \forall i \in (V - K)
\]

\[
T_{\text{start}} \leq d_i \leq T_{\text{start}} + t_{\text{bound}}, \quad \forall i \in V
\]

\[
d_k - Z(1 - x^k_{k\omega}) \leq T_{\text{start}}, \quad \forall k \in K
\]

Preventing Stagnation
Soft Delivery Timings

Problem:
- Hard delivery timings make problem infeasible.

Solution:
- Impose soft penalties for delivery time deviations.
- Permit skipping deliveries.

Objective Fn: \( \min_{x,a,d} \left\{ f_{time}(d) + \lambda f_{travel}(x, a, d) \right\} \)

where

\[ f_{time}(d) = \sum_{i \in (V-K)} p_i (d_i - \tau_i + T_{A,i})^2 \]

\[ f_{travel}(x, a, d) = \sum_{k \in K} \left( \sum_{(i,j) \in E \atop j \neq \omega} x_{ij}^k (a_j - d_i) + \sum_{(i,j) \in E \atop j = \omega} x_{ij}^k t_{ij} \right) \]

\[ \sum_{k \in K} \sum_{j \in V \cup \{\omega\}} x_{ij}^k \leq 1, \quad \forall i \in V \] Relaxed Deliveries

\[ T_{start} + t_{bound} \left( 1 - \sum_{k \in K} \sum_{j \in V \cup \{\omega\}} x_{ij}^k \right) \leq d_i \quad \forall i \in V \] Penalty for missing delivery
Full MIQP Formulation

Objective: \( \min_{x,a,d} \{ f_{\text{time}}(d) + \lambda f_{\text{travel}}(x, a, d) \} \)

where
\[
f_{\text{time}}(d) = \sum_{i \in (V - K)} p_i (d_i - \tau_i + T_{A,i})^2
\]
\[
f_{\text{travel}}(x, a, d) = \sum_{k \in K} \left( \sum_{(i,j) \in E, j \neq \omega} x_{ij}^k (a_j - d_i) + \sum_{(i,j) \in E, j = \omega} x_{ij}^k t_{ij} \right)
\]

subject to the constraints:

- **Time Flow:**
  \[
a_i - d_i + T_s \leq 0, \quad \forall i \in (V - K)
  \]
  \[
d_i - a_j + t_{ij} \leq Z \left( 1 - \sum_{k \in K} x_{ij}^k \right), \quad \forall (i, j) \in E, j \neq \omega
  \]
  \[
  T_{\text{start}} \leq a_i \leq T_{\text{start}} + t_{\text{bound}}, \quad \forall i \in (V - K)
  \]
  \[
  T_{\text{start}} \leq d_i \leq T_{\text{start}} + t_{\text{bound}}, \quad \forall i \in V
  \]
  \[
  T_{\text{start}} + t_{\text{bound}} \left( 1 - \sum_{k \in K} \sum_{j \in V \cup \{\omega\}} x_{ij}^k \right) \leq d_i, \quad \forall i \in V
  \]
  \[
  d_k - Z (1 - x_{k\omega}^k) \leq T_{\text{start}}, \quad \forall k \in K
  \]

- **Path Continuity:**
  \[
  \sum_{k \in K} \sum_{j \in V \cup \{\omega\}} x_{ij}^k \leq 1, \quad \forall i \in V
  \]
  \[
  x_{\alpha k}^k = 1, \quad \forall k \in K
  \]
  \[
  \sum_{i \in V} x_{i\omega}^k = 1, \quad \forall k \in K
  \]

- **Capacity:**
  \[
  \sum_{(i,j) \in E} x_{ij}^k - 1 \leq C^k - c^k, \quad \forall k \in K
  \]
  \[
  \sum_{(i,j) \in E} x_{ij}^k B_r^k (t_{ij} v) \leq B^k, \quad \forall k \in K
  \]
Solving the Scheduling Problem

- Always feasible! (Admits at least one solution: all $x_{ij}^{k'} s = 0$)
- MIQP (with non-PSD Hessian matrix) is NP-hard.
- Solution technique used:

Branch-and-Bound

Each node in branch-and-bound is a new MIP

1) Image taken from URL: [http://www.gurobi.com/resources/getting-started/mip-basics](http://www.gurobi.com/resources/getting-started/mip-basics)
Online System: Time Windows

- Complete list of requests not available a priori.
- Finite horizon $\rightarrow$ Time Window scheduling.
- Allows dynamic rescheduling.

\[ T_d \quad T_{wait} \quad T_w \]
Results

**Delivery robots**
(based on AscTec Hummingbirds)
- $v = 2 \text{ m/s}$
- Battery life: 1800 m (approx)
- $B_r = 1/1800 \text{ units/m}$
- Max capacity: Blue=3, Green=2

**Task robots**
(ground robots)
- Battery life: About 1-2 hr

**Scheduling parameters:**
- Time Window
  - $[40, 3080] \text{ s}$
  - $[3080, 6120] \text{ s}$
  - $[6120, 9160] \text{ s}$

---

**Window 1:** $[40, 3080] \text{ s}$
- Control Center
- Task Robots

**Window 2:** $[3080, 6120] \text{ s}$
- Control Center
- Task Robots

**Window 3:** $[6120, 9160] \text{ s}$
- Control Center
- Task Robots

---

**Time Window**
- $40 \text{ s} \rightarrow 3000 \text{ s}$
- $5000 \text{ s}$
Results

**Delivery robots**
(based on AscTec Hummingbirds)
- $v = 2 \text{ m/s}$
- Battery life: 1800 m (approx)
- $B_r^K = 1/1800 \text{ units/m}$
- Max capacity: Blue=3, Green=2

**Task robots**
(ground robots)
- Battery life: About 1-2 hr

**Scheduling parameters:**

<table>
<thead>
<tr>
<th>Time Window</th>
<th>40 s</th>
<th>3000 s</th>
<th>5000 s</th>
</tr>
</thead>
</table>

**Window1:** [40, 3080] s
- $a = 151.8 \text{ s}$
- $d = 2740.0 \text{ s}$
- $t = 2740.0 \text{ s}$
- $a = 2952.2 \text{ s}$
- $d = 2456.5 \text{ s}$
- $a = 2920.3 \text{ s}$
- $t = 3890.0 \text{ s}$
- $t = 4840.0 \text{ s}$

**Window2:** [3080, 6120] s

**Window3:** [6120, 9160] s
Results

**Scheduling parameters:**

**Delivery robots** (based on AscTec Hummingbirds)
- \( v = 2 \text{ m/s} \)
- Battery life: 1800 m (approx)
- \( B_r^k = 1/1800 \text{ units/m} \)
- Max capacity: Blue=3, Green=2

**Task robots** (ground robots)
- Battery life: About 1-2 hr

**Window1:** [40, 3080] s
- \( a = 151.8 \text{ s} \)
- \( d = 2740.0 \text{ s} \)
- \( t = 2740.0 \text{ s} \)

**Window2:** [3080, 6120] s
- \( a = 4140.0 \text{ s} \)
- \( d = 5680.1 \text{ s} \)
- \( t = 5680.0 \text{ s} \)

**Window3:** [6120, 9160] s
- \( a = 5041.6 \text{ s} \)
- \( d = 6180.0 \text{ s} \)
- \( t = 6180.0 \text{ s} \)

**Time Window**

- \( 40 \text{ s} \to 3000 \text{ s} \to 5000 \text{ s} \)
Results

Delivery robots
(based on AscTec Hummingbirds)

- \( v = 2 \text{ m/s} \)
- Battery life: 1800 m (approx)
- \( B_r^k = 1/1800 \text{ units/m} \)
- Max capacity: Blue=3, Green=2

Task robots
(ground robots)

- Battery life: About 1-2 hr

Scheduling parameters:

\begin{align*}
\text{Time Window:} & \quad 40 \text{ s} \quad \longrightarrow \quad 3000 \text{ s} \\
\text{Window1:} & \quad [40, 3080] \text{ s} \\
\text{Window2:} & \quad [3080, 6120] \text{ s} \\
\text{Window3:} & \quad [6120, 9160] \text{ s}
\end{align*}
Time Complexity

- 500 single window trials with random number of robots, locations, delivery timings etc.
- Computation times averaged over instances with same values of M and N.
- Exponential growth – Useful for small groups of robots.
Contributions

- Solved the resource delivery problem with timed requests optimally.
- Problem formulation always has a feasible solution.
- Relaxed scheduling permitted when there is lack of resources or delivery robots.
- Enable dynamic re-routing of delivery robots enroute.
- Impose relative priorities when all task robots are not equally important.
Future Work

- **Tradeoff**: Approximate solution vs. faster computation.
- Decentralized planning.
- Removing synchronous time windows, to make planning asynchronous.
Thank you

Questions?