Real Time Adaptive Scheduling for a Swarm of Manufacturing Robots

Mac Schwager
Assistant Professor
Aeronautics and Astronautics
Stanford University
Flexible Manufacturing

Credit: Martin Sehr, Siemens

Credit: Martin Sehr, Siemens

Credit: ECM
Flexible Manufacturing Scenario

Process Stations

Robots

Queues

Out Bin

Data Center
How to coordinate, control, and plan for large teams of manufacturing robots in dynamic, stochastic environments?

Fully distributed
Local information
Limited communication
Reactive control and sensing

Fully centralized
Global information
Communication intensive
Optimal planning and scheduling

Credit: Business-opportunities.biz
Credit: Farelli.info
Our Approach

Local sensor information → Local-Global Map → Global state

Individual robots → Global-Local Online replanning → Centralized planner
• High level assignment scheduling
• Low-level path planning
• Local-global map
• High level assignment scheduling

• Low-level path planning

• Local-global map
Assignment Scheduling

Patrick Washington
Problem Setup

Robot State Machine

Algorithm 1 Robot i State Machine

Inputs:
\( b_i^{max} \), \( D \), \( v_i \), \( r_c \), \( r_d \)

Initialize:
\( b_i \leftarrow b_i^{max} \)
\( a_i \leftarrow 0 \)
\( \tau_i \leftarrow 0 \)
\( w_i \leftarrow 0 \)

\( p_i \leftarrow 0 \)

if \( \tau_i = 0 \) and \( w_i = 0 \) then \( \triangleright \) Set Time to Goal
if \( a_i > 0 \) then \( \triangleright \) Time to Widget
\( \tau_i \leftarrow D(p_i, O_{a_i}(s_{a_i}))/v_i \)
else if \( a_i = 0 \) then \( \triangleright \) Time to Charger
\( \tau_i \leftarrow D(p_i, 0)/v_i \)
end if
end if

if \( \tau_i = 0 \) and \( a_i > 0 \) then \( \triangleright \) Arrived with/at Widget
\( p_i \leftarrow O_{a_i}(s_{a_i}) \) \( \triangleright \) Record Position
if \( w_i = 0 \) and \( \tau_s \leq 0 \) then \( \triangleright \) Pick Up and Go
\( w_i \leftarrow 1 \)
\( \tau_s \leftarrow D(O_{a_i}(s_{a_i}), O_{a_i}(s_{a_i} + 1)) \)
else \( \triangleright \) Drop Off/Idle
\( w_i \leftarrow 0 \)
end if
end if

if \( \tau_i > 0 \) then \( \triangleright \) Moving/Discharging
\( \tau_i \leftarrow -1 \)
\( b_i \leftarrow -r_d \)
else if \( a_i = 0 \) then \( \triangleright \) Charging
\( p_i \leftarrow 0 \)
\( \tau_i \leftarrow 0 \)
if \( b_i = b_i^{max} \) then \( \triangleright \) Do Not Overcharge
\( b_i \leftarrow 0 \)
else \( \triangleright \) Idle
\( b_i \leftarrow r_c \)
end if
else \( \tau_i \leftarrow 0 \)
\( b_i \leftarrow 0 \)
end if

Widget State Machine

Algorithm 2 Widget j State Machine

Inputs:
\( O_{j}, T_{j}, \tau_j(0) \)

Initialize:
\( s_j \leftarrow 1 \)
\( \tau_j \leftarrow \tau_j(0) \)
if \( r_j = 0 \) and \( a_i = j \) and \( \tau_i = 0 \) and \( \tau_j \leq 0 \) then \( \triangleright \) Pick Up
\( r_j \leftarrow 1 \)
\( s_j \leftarrow s_j + 1 \)
end if
if \( r_j > 0 \) and \( \tau_j = 0 \) then \( \triangleright \) Drop Off
\( r_j \leftarrow 0 \)
\( \tau_j \leftarrow T_j(s_j) \)
end if
if \( s_j = \text{length}(O_j) \) and \( \tau_j = 0 \) then \( \triangleright \) Widget Finished
Finished
end if
\( \tau_j \leftarrow -1 \)
\( r_j \leftarrow -1 \)

Unexpectedly Complex!
Optimization Formulation

- Minimize robot battery discharge

- Dynamics set in state machines

- Robots must not die

- Deadlines must be met

\[
\min_{a(t)} \int_{0}^{T} \sum_{i=1}^{N} b_{i}^{\text{out}}(t) dt
\]

subject to

- Timer dynamics
- Charge dynamics

\[
b_{i}(t) > 0 \quad \forall i, \forall t
\]

\[
\tau_{d}^{j}(t) \geq 0 \quad \forall j, \forall t
\]

Intractable!
Greedy Assignment Heuristic

Robots  Tasks

Poly time Algorithm: Hungarian Algorithm

Resolve at each time step, real time for 1000s of robots
## Assignment Cost Design

<table>
<thead>
<tr>
<th>Robots</th>
<th>Tasks</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>$c_{11}$</td>
<td>$c_{12}$</td>
<td>$c_{13}$</td>
<td>$c_{14}$</td>
<td>$c_{15}$</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>$c_{21}$</td>
<td>$c_{22}$</td>
<td>$c_{23}$</td>
<td>$c_{24}$</td>
<td>$c_{25}$</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>$c_{31}$</td>
<td>$c_{32}$</td>
<td>$c_{33}$</td>
<td>$c_{34}$</td>
<td>$c_{35}$</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>$c_{41}$</td>
<td>$c_{42}$</td>
<td>$c_{43}$</td>
<td>$c_{44}$</td>
<td>$c_{45}$</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>$c_{51}$</td>
<td>$c_{52}$</td>
<td>$c_{53}$</td>
<td>$c_{54}$</td>
<td>$c_{55}$</td>
</tr>
</tbody>
</table>

- The Black Magic: Hand-tuned assignment costs
- Incorporates:
  - Travel distance
  - Time deadlines
  - Heuristic look-ahead penalty
  - Lots of intuition
- Working to formalize cost design mechanisms
• High level assignment scheduling
• Low-level path planning
• Local-global map
Multi-Robot Routing in Graph

Oriana Peltzer

Job Queue
C, D, A
A, B
C, A, C

Out Bin
Transform to Network Flow Problem

One edge
One robot

Solve as Integer Linear Program

maximize \[ c^T x \]
subject to \[ Ax \leq b \]
\[ x \geq 0 \]
\[ x \in \mathbb{Z}^n \]

- Poly time for interchangeable jobs (reduces to LP)
- Flexible capacity limits on edges
- Wait times on nodes
• High level assignment scheduling
• Low-level path planning
• Local-global map
Local-Global Map

Kyle Brown

Job Queue

Out Bin
From Observed Interruptions to Trajectory Prediction
From Observed Interruptions to Trajectory Prediction

Bayesian filter for trajectory estimation
Model regression for intent prediction
Update edge travel times for re-routing

Data Center
Looking Forward

• Formalize assignment costs in Hungarian heuristic
• Frame multi-robot routing problem as single (or few) commodity flow for computational efficiency
• Build estimation and intent prediction framework
• Integrate in simulation environment
Thanks!

Prof. Mykel Kochenderfer  
Dr. Martin Sehr

Patrick Washington  
Oriana Peltzer  
Kyle Brown

Feb 7, 2019  
Platform Annual Review