Math for the Digital Factory

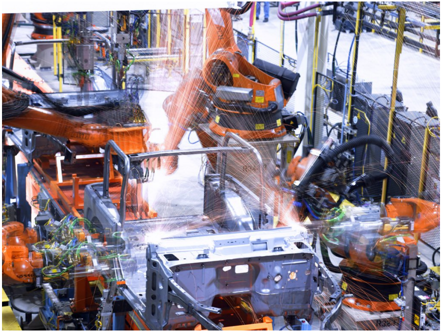
Combinatorial Optimization Aspects of Robot Tour Planning

Chantal Landry Martin Skutella Wolfgang Welz

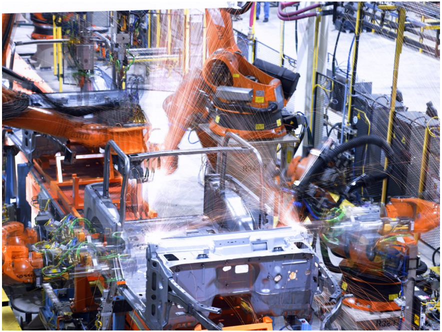
DFG Research Center MATHEON
Mathematics for key technologies

May 8, 2014

DFG Research Center MATHEON Project:
Automatic reconfiguration of robotic welding cells



DFG Research Center MATHEON Project:
Automatic reconfiguration of robotic welding cells



Problem:

- Robots perform spot welding tasks on single component
- Some points can only be processed by specific robots
- Robots must not collide
- Given cycle time

Discrete Part

- task assignment
- sequencing of weld points

Requires

- distances between weld points
- collision information

Continuous Part

- path planning
- collision detection and avoidance

Requires

- weld point sequence

Discrete Part

- task assignment
- sequencing of weld points

Continuous Part

- path planning
- collision detection and avoidance

Given

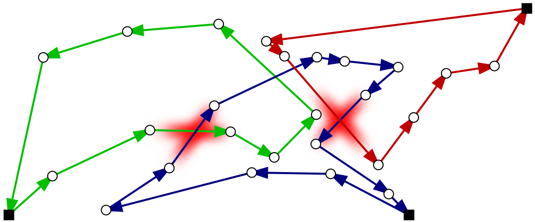
- distances between weld points
- collision information

Requires

- weld point sequence

Vehicle Routing Problem with collision constraints

- Representation as a graph for each robot:
  - nodes ⇔ weld points that can be visited
  - arcs ⇔ paths between two weld points
- Each arc has a travel time



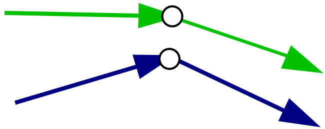
Collisions: Certain moves of two robots must not be made at the same time

Scheduled Tours for the robots:

- A tour with integer start and end times for each arc:
  - $end_a - start_a =$ traversal time of a
  - If $end_a < start_b$ we wait in node v

Collisions between robots at the same time:

- Both robots waiting: node-node collision $\Downarrow$
- One robot moving and one waiting: node-arc collision $\Downarrow$
- Both robots moving: arc-arc collision

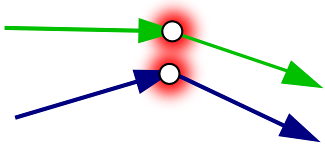


Scheduled Tours for the robots:

- A tour with integer start and end times for each arc:
  - $end_a - start_a =$ traversal time of a
  - If $end_a < start_b$ we wait in node v

Collisions between robots at the same time:

- Both robots waiting: node-node collision
  $\Downarrow$
- One robot moving and one waiting: node-arc collision
  $\Downarrow$
- Both robots moving: arc-arc collision

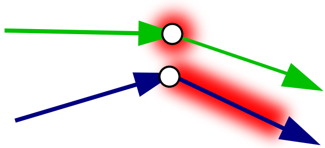


Scheduled Tours for the robots:

- A tour with integer start and end times for each arc:
  - $end_a - start_a =$ traversal time of a
  - If $end_a < start_b$ we wait in node v

Collisions between robots at the same time:

- Both robots waiting: node-node collision $\Downarrow$
- One robot moving and one waiting: node-arc collision $\Downarrow$
- Both robots moving: arc-arc collision



Scheduled Tours for the robots:

▷ A tour with integer start and end times for each arc:

- $end_a - start_a =$ traversal time of a
- If $end_a < start_b$ we wait in node v

Collisions between robots at the same time:

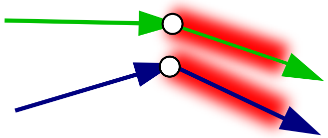
▷ Both robots waiting: node-node collision

$\Downarrow$

▷ One robot moving and one waiting: node-arc collision

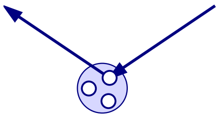
$\Downarrow$

▷ Both robots moving: arc-arc collision



- the distance depends on the orientation of the robot
- collisions are too restrictive
- slight changes of trajectories to avoid collisions

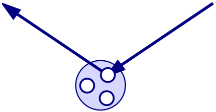
- the distance depends on the orientation of the robot



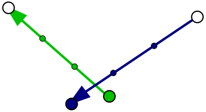
- collisions are too restrictive

- slight changes of trajectories to avoid collisions

▷ the distance depends on the orientation of the robot

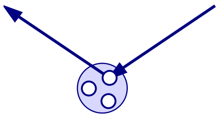


▷ collisions are too restrictive

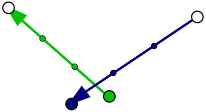


▷ slight changes of trajectories to avoid collisions

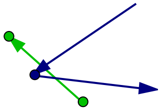
- the distance depends on the orientation of the robot



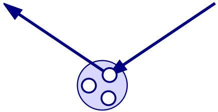
- collisions are too restrictive



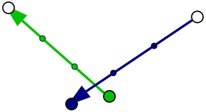
- slight changes of trajectories to avoid collisions



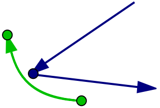
- the distance depends on the orientation of the robot



- collisions are too restrictive



- slight changes of trajectories to avoid collisions



For each feasible scheduled tour $t \in \mathcal{T}$ there is a 0/1–variable x_t

$$\min \sum_{t \in \mathcal{T}} c_t x_t \tag{WCP}$$

$$\text{s.t.} \sum_{t \in \mathcal{T}} \delta_{vt} x_t = 1 \qquad \forall v \in V$$

$$\mathbf{x} \ \textbf{ is collision free} \tag{1}$$

$$x_t \in \{0, 1\} \qquad \forall t \in \mathcal{T} \tag{2}$$

For each feasible scheduled tour $t \in \mathcal{T}$ there is a 0/1–variable x_t

$$
\begin{aligned}
\min \ & \sum_{t \in \mathcal{T}} c_t x_t & & \text{(WCP)} \\
\text{s.t.} \ & \sum_{t \in \mathcal{T}} \delta_{vt} x_t = 1 & \forall v \in V & \\
& \mathbf{x} \ \textbf{ is collision free} & & (1) \\
& x_t \in \{0, 1\} & \forall t \in \mathcal{T} & \quad (2)
\end{aligned}
$$

Using branch-and-price approach:

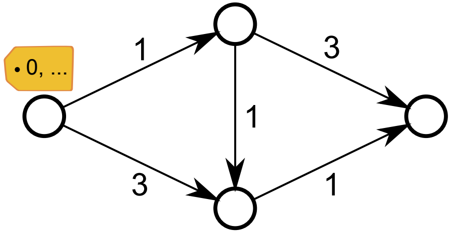
- Constraints (1) and (2) are enforced by branching in a branch and bound framework.
- For (1), conflicting arcs are forced/forbidden in certain time windows
- Pricing: Elementary shortest path with negative costs and time windows

Resource Constrained Shortest Path

- Sub-problem for many VRP branch-and-price approaches
- State of the art: Bidirectional labeling algorithms

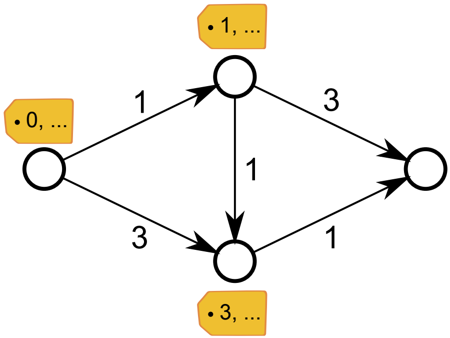
Resource Constrained Shortest Path

- Sub-problem for many VRP branch-and-price approaches
- State of the art: Bidirectional labeling algorithms



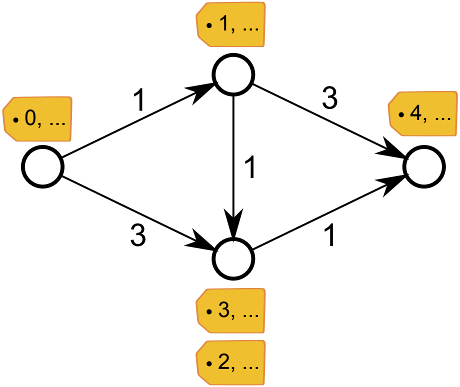
Resource Constrained Shortest Path

- Sub-problem for many VRP branch-and-price approaches
- State of the art: Bidirectional labeling algorithms



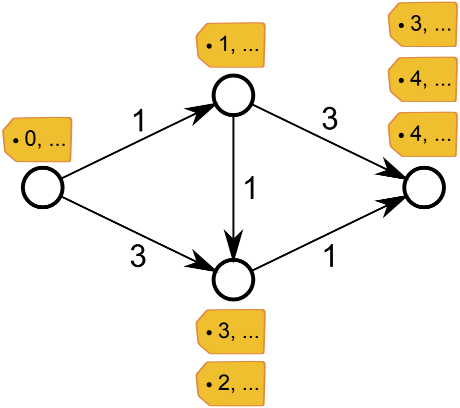
Resource Constrained Shortest Path

- Sub-problem for many VRP branch-and-price approaches
- State of the art: Bidirectional labeling algorithms



Resource Constrained Shortest Path

- Sub-problem for many VRP branch-and-price approaches
- State of the art: Bidirectional labeling algorithms



Drawbacks

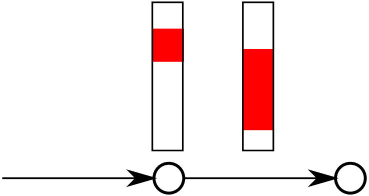
- Elementary shortest path is expensive
- Algorithms do note scale very well

Standard dominance criterion does not hold!

Drawbacks

- Elementary shortest path is expensive
- Algorithms do note scale very well

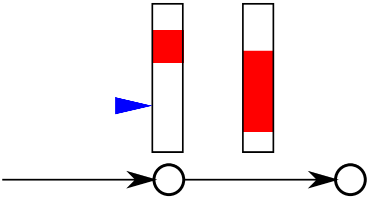
Standard dominance criterion does not hold!



Drawbacks

- Elementary shortest path is expensive
- Algorithms do note scale very well

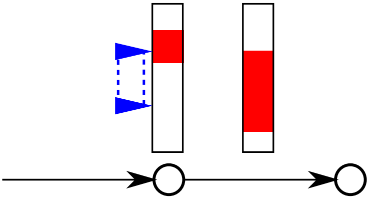
Standard dominance criterion does not hold!



Drawbacks

- Elementary shortest path is expensive
- Algorithms do note scale very well

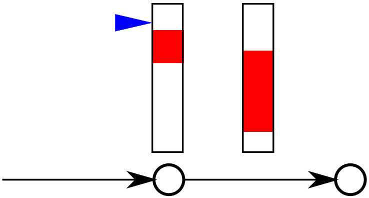
Standard dominance criterion does not hold!



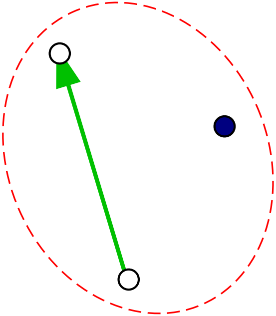
Drawbacks

- Elementary shortest path is expensive
- Algorithms do note scale very well

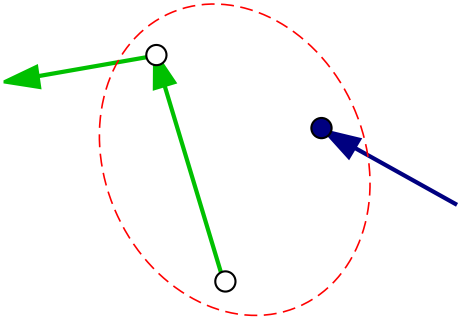
Standard dominance criterion does not hold!



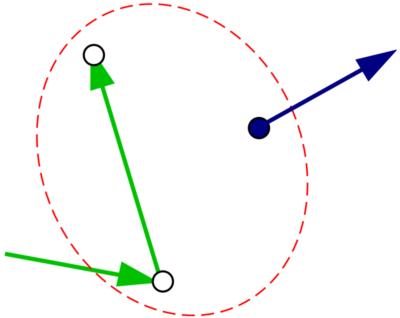
- Take the collision informaiton into account
- Find conflicting edge-node pairs



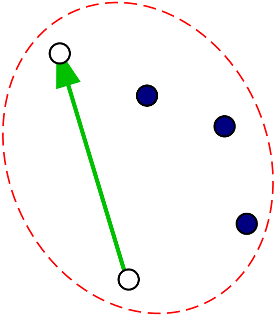
- Take the collision informaiton into account
- Find conflicting edge-node pairs



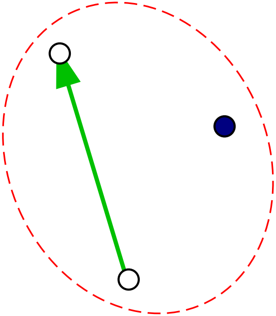
- Take the collision informaiton into account
- Find conflicting edge-node pairs



- Take the collision informaiton into account
- Find conflicting edge-node pairs



- Take the collision informaiton into account
- Find conflicting edge-node pairs



- If this leads to an infeasible node for all robots ⇒ delete the edge
- This is a cut for every infeasible robot

Greedy-partitioning heuristic:

- Assign every node to the closest robot
- Solve the TSP-Problem (approximately) for all robots
- If the cycle time is violated reassign some nodes

LP-rounding heuristic:

- Remove nodes from tours with smaller LP-values
- Concatenate all tours for one robot
- Solve additional sub-MIP to find feasible waiting times:
  - Leads to one binary variable for every conflicting
  - Even for larger instances ($>$ 35 nodes) solving takes less than 1 second

Improving Computational Performance

- Penalizing high collision edges
- Efficient data structures for forbidden periods
- Solve the elementary Shortest Path Problem:
  - Using 2-cycle elimination
  - Using decremental state-space relaxation
- Use Robust Tours instead of Scheduled Tours:
  - Reducing the number of variables
  - Detecting collisions is more complicated
- Solve the pricing problems in parallel on shared memory systems
- Reverse tours in heuristics

Math for the Digital Factory

Combinatorial Optimization Aspects of Robot Tour Planning

Chantal Landry Martin Skutella Wolfgang Welz

DFG Research Center Matheon
Mathematics for key technologies

May 8, 2014