A bit about NICTA

• National Information and Communications Technology Australia (NICTA)
• Research in ICT since 2004
• Major Labs in Sydney, Melbourne, Canberra
• 700 researchers, including 300 PhD students
• Currently government funded
• Areas:
  – Broadband and the Digital Economy
  – Health
  – Infrastructures, Transport and Logistics
  – Safety and Security
Outline

• What is the vehicle routing problem (VRP)?
• ‘Traditional’ (Non-CP) solution methods
• A CP model for the VRP
• Propagation
• Large Neighbourhood Search revisited
What is the Vehicle Routing Problem?

Given a set of customers, and a fleet of vehicles to make deliveries, find a set of routes that services all customers at minimum cost.
Vehicle Routing Problem
Vehicle Routing Problem
Vehicle Routing Problem

VRP is hard

• With a VRP solver I can solve the Travelling Salesman Problem (using 1 vehicle and infinite capacity)
Why study the VRP?

• The logistics task is 9% of economic activity in Australia
• Logistics accounts for 10% of the selling price of goods
Vehicle Routing Problem

For each customer, we know
- Quantity required
- The cost to travel to every other customer

For the vehicle fleet, we know
- The number of vehicles
- The capacity

We must determine which customers each vehicle serves, and in what order, to minimise cost
Vehicle Routing Problem

Objective function

In academic studies, usually a combination:
- First, minimise number of routes
- Then minimise total distance or total time

In real world
- A combination of time and distance
- Must include vehicle- and staff-dependent costs
- Usually vehicle numbers are fixed
Routing with constraints

• Each new twist (business rule, practice, limitation) changes which solutions are good, and which are even acceptable

• The types of constraints that must be observed may impact on the way the problem is solved

• Many types of constraint studied – but usually in isolation

• We will look at a few that have been studied in the literature
Time window constraints

Vehicle routing with Constraints

- Time Window constraints
  - A window during which service can start
  - E.g. only accept delivery 7:30am to 11:00am

- Additional input data required
  - Duration of each customer visit
  - Time between each pair of customers

- (Travel time can be vehicle-dependent or time-dependent)
  - Makes the route harder to visualise
Time Window constraints
Pickup and Delivery problems

- Most routing considers delivery to/from a depot (depots)
- Pickup and Delivery problems consider FedEx style problem:

\[ \text{pickup at location A, deliver to location B} \]
Other variants

Profitable tour problem
- Not all visits need to be completed
- Known profit for each visit
- Choose a subset that gives maximum return (profit from visits – routing cost)
VRP meets the real world

Many groups now looking at real-world constraints

*Rich Vehicle Routing Problem*

- Attempt to model constraints common to many real-life enterprises
  - Multiple Time windows
  - Multiple Commodities
  - Heterogeneous vehicles
  - Compatibility constraints
    - Goods for customer $A$ can’t travel with goods from customer $B$
    - Goods for customer $A$ can’t travel on vehicle $C$
VRP meets the real world

Other real-world considerations

- Fatigue rules and driver breaks
- Vehicle re-use
- Ability to change vehicle characteristics (composition)
  - Add trailer, or move compartment divider
- Use of limited resources
  - e.g. limited docks for loading, hence need to stagger dispatch times
- Variable loading / unloading times
Solution Methods
Solving VRPs

VRP is hard

- *NP Hard* in the strong sense
- Exact solutions only for small problems (20-50 customers)
- Most solution methods are heuristic
ILP

\[
\text{minimise: } \sum_{i,j} c_{ij} \sum_k x_{ijk}
\]

subject to

\[
\sum_{i} \sum_k x_{ijk} = 1 \quad \forall j
\]

\[
\sum_{j} \sum_k x_{ijk} = 1 \quad \forall i
\]

\[
\sum_{j} \sum_k x_{ihk} - \sum_{j} \sum_k x_{hjk} = 0 \quad \forall k, h
\]

\[
\sum_i q_i \sum_j x_{ijk} \leq Q_k \quad \forall k
\]

\[
\{x_{ijk}\} \subseteq S
\]

\[
x_{ijk} \in \{0,1\}
\]

Advantages
- Can find optimal solution

Disadvantages
- Only works for small problems
- One extra constraint \(\rightarrow\) back to the drawing board
**ILP – Column Generation**

<table>
<thead>
<tr>
<th></th>
<th>89</th>
<th>76</th>
<th>99</th>
<th>45</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>4</strong></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>5</strong></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- **Columns** represent routes
- **Column/route cost** $c_k$
- **Rows** represent customers
- **Array entry** $a_{ik} = 1$ iff customer $i$ is covered by route $k$
Column Generation

- Decision var $x_k$: Use column $k$?
- Column only appears if feasible ordering is possible
- Cost of best ordering is $c_k$
- Best order stored separately
- Master problem at right

\[
\min \sum c_k x_k \\
\text{subject to} \quad \sum_{k} a_{ik} = 1 \quad \forall i \\
x_k \in \{0,1\}
\]
Column Generation

Subproblem

\[
\min \sum_{i,j} x_{ij} (c_{ij} + \lambda_j)
\]

s.t. \[\sum_j x_{ij} = 1 \quad \forall i\]

\[\sum_i x_{ij} = 1 \quad \forall j\]

Route constraints

- i.e. shortest path with side constraints
- If min is –ve add to master problem
- CP!
Heuristic Methods

• Most operate as
  – Construct
  – Improve (Local Search)
Heuristic methods -
Construction

Insert methods

Insert1.dig

Order is important:

Insert2.dig

Compare.dig
Heuristic methods -

Construction

Savings method (Clarke & Wright 1964)

- Calculate $S_{ij}$ for all $i, j$
- Consider cheapest $S_{ij}$
- If $j$ can be appended to $i$
  - merge them to new $i$
  - update all $S_{ij}$
- else
  - delete $S_{ij}$
- Repeat
Improvement Methods

2-opt (3-opt, 4-opt…)
- Remove 2 arcs
- Replace with 2 others
Improvement methods

Large Neighbourhood Search
= Destroy & Re-create

• Destroy part of the solution
  – Remove visits from the solution

• Re-create solution
  – Use favourite construct method to re-insert customers

• If the solution is better, keep it

• Repeat
Improvement methods

Variable Neighbourhood Search

- Consider multiple neighbourhoods
  - 1-move (move 1 visit to another position)
  - 1-1 swap (swap visits in 2 routes)
  - 2-2 swap (swap 2 visits between 2 routes)
  - 2-opt
  - 3-opt
  - Or-opt size 2 (move chain of length 2 anywhere)
  - Or-opt size 3 (chain length 3)
  - Tail exchange (swap final portion of routes)
Improvement methods

Variable Neighbourhood Search

- Consider multiple neighbourhoods
- Find local minimum in smallest neighbourhood
- Advance to next-largest neighbourhood
- Search current neighbourhood
  - If a change is found, return to smallest neighbourhood
  - Otherwise, advance to next-largest
Genetic Programming

- Generate a population of solutions (construct methods)
- Evaluate fitness (objective)
- Create next generation:
  - Choose two solutions from population
  - Combine the two (two ways)
  - (Mutate)
  - Produce offspring (calculate fitness)
  - (Improve)
  - Repeat until population doubles
- Apply selection:
  - Bottom half “dies”
- Repeat
Solution Methods

.. and the whole bag of tricks

- Tabu Search
- Simulated Annealing
- Ants
- Bees
- ....
Solution Methods

What’s wrong with that?

• New constraint $\rightarrow$ new code
  – Often right in the core

• New constraints interact
  – e.g. Multiple time windows mess up duration calculation

• Code is hard to understand, hard to maintain
Solution Methods

An alternative: Constraint Programming

Advantages:
• Expressive language for formulating constraints
• Each constraint encapsulated
• Constraints interact naturally

Disadvantages:
• Difficulty in representation
• Can be slower
Two ways to use constraint programming

- Rule Checker
- Properly

Rule Checker:
- Use favourite method to create/improve a solution
- Check it with CP
  - Very inefficient.
A CP Model for the VRP
A Model for a (Rich) VRP

Locations

• Fixed locations
  – where things happen
  – one for each customer + one for (each?) depot

Metrics

• Know time $T_{i,j}$ between each location pair $i, j$
• Know cost $C_{i,j}$ between each location pair $i, j$
  – Both obey triangle inequality
  – (Not always true in practice)

Commodities

• $c$ commodities (e.g. weight, volume, pallets)
A Model for a (Rich) VRP

Customers

- $n$ customers (fixed in this model)
  - Known demand $D_{i,k}$ of commodity $k$ for customer $i$
  - Known “value” $V_i$ of customer $i$ – penalty for not servicing
  - Know Time Windows $E_i, L_i$, earliest and latest start times
  - Know Service time $S_i$
A Model for a (Rich) VRP

Vehicles

- $m$ vehicles / routes (fixed in this model)
  - Assume 1 route per vehicle

- Known Capacity $Q_{j,k}$ for commodity $k$ on vehicle $j$

- Known start location
- Know start time
- Known end location
- Know latest return time
Vocabulary

• A solution is made up of routes (one for each vehicle)
• A route is made up of a sequence of visits
• Some visits serve a customer (customer visit)

(Some tricks)
• Each route has a “start visit” and an “end visit”
• Start visit is first visit on a route – location is depot
• End visit is last visit on a route – location is depot
• Also have an additional route – the unassigned route
  – Where visits live that cannot be assigned
Referencing

Customers
- Each customer has an index in $N = \{1..n\}$
- Customers are ‘named’ in CP by their index

Routes
- Each route has an index in $M = \{1..m\}$
- Unassigned route has index 0
- Routes are ‘named’ in CP by their index

Visits
- Customer visit index same as customer index
- Start visit for route $k$ has index $n + k$; aka $\text{start}_k$
- End visit for route $k$ has index $n + m + k$; aka $\text{end}_k$
Vocabulary
Sets

- $N = \{1 \ldots n\}$ – customers
- $M = \{1 \ldots m\}$ – routes
- $R = M \cup \{0\}$ – includes ‘unassigned’ route
- $S = \{n+1 \ldots n+m\}$ – start visits
- $E = \{n+m+1 \ldots n+2m\}$ – end visits
- $V = N \cup S \cup E$ – all visits
- $V^S = N \cup S$ – visits that have a sensible successor
- $V^E = N \cup E$ – visits that have a sensible predecessor
Data

We know (note uppercase)

- $V_i$ The ‘value’ of customer $i$
- $D_{ik}$ Demand by customer $i$ for commodity $k$
- $E_i$ Earliest time to start service at $i$
- $L_i$ Latest time to start service at $i$
- $S_i$ Service time of visit at $i$
- $Q_{jk}$ Capacity of vehicle $j$ for commodity $k$
- $T_{ij}$ Travel time from visit $i$ to visit $j$
- $C_{ij}$ Cost (w.r.t. objective) of travel from $i$ to $j$
Decision Variables

Successor variables: \( s_i \)
- \( s_i \) gives direct successor of \( i \), i.e. the index of the next visit on the route that visits \( i \)
- \( s_i \in V^E \) for \( i \) in \( V^S \)  \( s_i = 0 \) for \( i \) in \( E \)

Predecessor variables \( p_i \)
- \( p_i \) gives the index of the previous visit in the route
- \( p_i \in V^S \) for \( i \) in \( V^E \)  \( p_i = 0 \) for \( i \) in \( S \)
- Redundant – but empirical evidence for its use

Route variables \( r_i \)
- \( r_i \) gives the index of the route (vehicle) that visits \( i \)
- \( r_i \in R \)
Example

<table>
<thead>
<tr>
<th>i</th>
<th>s_i</th>
<th>p_i</th>
<th>r_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

Route 1

Route 2
Other variables

Accumulation Variables

- $q_{ik}$ Quantity of commodity $k$ after visit $i$
- $c_i$ Objective cost getting to $i$

For problems with time constraints

- $a_i$ Arrival time at $i$
- $t_i$ Start time at $i$ (time service starts)
- $d_i$ Departure time at $i$

- Actually, only $t_i$ is required, but others allow for expressive constraints
Objective

Objective is

• Minimize
  – sum of objective \((C_{ij})\) over used arcs, plus
  – value of unassigned visits

• Minimize

\[
\sum_{i \in V^S, r_i \neq 0} C_{i,s_i} + \sum_{i \in N | r_i = 0} V_i
\]
Basic constraints

Path \( (S, E, \{ s_i \mid i \in V \} ) \)

AllDifferentExceptZero \( (\{ p_i \mid i \in V^E \} ) \)

Accumulate obj.
\[
c_{s_i} = c_i + C_{i,s_i} \quad \forall i \in V^S
\]

Accumulate time
\[
a_{s_i} = d_i + T_{i,s_i} \quad \forall i \in V^S
\]

Time windows
\[
t_i \geq a_i \quad \forall i \in V
\]
\[
t_i \leq L_i \quad \forall i \in V
\]
\[
t_i \geq E_i \quad \forall i \in V
\]
<table>
<thead>
<tr>
<th>Constraints</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Load</strong></td>
<td>$q_{sik} = q_{ik} + Q_{sik}$ $\forall i \in V^S, k \in C$</td>
</tr>
<tr>
<td></td>
<td>$q_{ik} \leq Q_{r_{ik}}$ $\forall i \in V, k \in C$</td>
</tr>
<tr>
<td></td>
<td>$q_{ik} \geq 0$ $\forall i \in V, k \in C$</td>
</tr>
<tr>
<td></td>
<td>$q_{ik} = 0$ $\forall i \in S, k \in C$</td>
</tr>
<tr>
<td><strong>Consistency</strong></td>
<td>$c_i = 0$ $\forall i \in S$</td>
</tr>
<tr>
<td></td>
<td>$s_{pi} = i$ $\forall i \in V^S$</td>
</tr>
<tr>
<td></td>
<td>$p_{si} = i$ $\forall i \in V^E$</td>
</tr>
<tr>
<td></td>
<td>$r_i = r_{si}$ $\forall i \in V^S$</td>
</tr>
<tr>
<td></td>
<td>$r_{n+k} = k$ $\forall k \in M$</td>
</tr>
<tr>
<td></td>
<td>$r_{n+m+k} = k$ $\forall k \in M$</td>
</tr>
</tbody>
</table>
What can we model?

- Basic VRP
- VRP with time windows (VRPTW)
- Multi-depot
- Heterogeneous fleet
- Orienteering / Profitable tour problems
- Open VRP (vehicle not required to return to base)
  - Requires *anywhere* location
  - Route end visits located at *anywhere*
  - Distance $i \rightarrow anywhere = 0$
- Compatibility
  - Customers on different / same vehicle
  - Customers on/not on given vehicle
- Pickup and Delivery problems
What can we model?

- Variable load/unload times
  - by constraining departure time relative to start time
- Dispatch time constraints
  - e.g. limited docks
  - $e_i$ for $i$ in $S$ is load-start time
- Depot close time
  - Time window on end visits
- Fleet size and mix
  - Add lots of vehicles
  - Need to introduce a ‘fixed cost’ for a vehicle
  - $C_{ij}$ is increased by fixed cost for all $i \in S$, all $j \in N$
What can’t we model

• Can’t handle dynamic problems
  – Fixed domain for \( s, p, r \) vars

• Can’t introduce new visits post-hoc
  – E.g. optional driver break must be allowed at start

• Can’t decide how many visits to same customer
  – ‘Larger than truck-load’ problems
  – If qty is fixed, can have multiple visits / cust
  – Heterogeneous fleet is a pain

• Can’t handle time- or vehicle-dependent travel times/costs

• Soft Constraints
Solving the CP
Solving

Pure CP

- Assign to ‘successor’ variables
- Form chains of visits
- Decision 1: Which visit to insert
- Decision 2: Where to insert it

- ‘Rooted chain’ starts at Start
- ‘Free chain’ otherwise
- Can reason about free chains but rooted chains easier
Propagation – Cycles

Subtour elimination

• Rooted chains are fine
• For free chains:
  – “last \(j\)” is last visit in chain starting at \(j\)
  – for any chain starting at \(j\),
    • remove \(j\) from \(S_{\text{last}}(j)\)

• Some CP libraries have built-ins
  – Comet: ‘circuit’
  – ILOG: Path constraint
Propagation – Cycles

‘Path’ constraint

• Propagates subtour elimination
• Also propagates cost

• path \((S, E, succ, P, z)\)
  – \(succ\) array implies path
  – ensures path from nodes in \(S\) to nodes in \(E\) through nodes in \(P\)
  – variable \(z\) bounds cost of path
  – cost propagated incrementally based on shortest / longest paths
Propagation – Cost

‘Path’ constraint
Maintains sets

- Path is consistent if it starts in $S$, ends in $T$, goes through $P$, and has bounds consistent with $z$
- $Pos \subseteq P \cup S \cup T$: Possibly in the path
  - If no consistent paths use $i$, then $i \notin Pos$
- $Req \subseteq Pos$: Required to be in the path
  - If there is a consistent path that does not need $i$, then $i \notin Req$
- Shortest path in $Req \rightarrow$ lower bound on $z$
- Longest path in $Pos \rightarrow$ upper bound on $z$
  == Shortest path with –ve costs
- Updated incrementally (and efficiently)
Simulated Annealing

shortcut

- leverages cost estimate from Path constraint

- std SA:
  - Generate sol
  - test delta obj against uniform rand var

- improved SA
  - generate random var first
  - calc acceptable obj
  - constrain obj

- Much more efficient

\[ P(\text{accept}) = e^{\frac{-\partial}{T}} \]

\[ \partial < T \log(x) \]

\[ z < z^* + \partial \]
Propagation – Capacity

Rooted Chain

- For each visit \( i \) with \( p_i \) not bound
  - For each route \( k \) in domain of \( r_i \)
    - If spare space on route \( k \) won’t allow visit \( i \)
      - Remove \( k \) from \( r_i \)
      - Remove \( i \) from \( p_{\text{end}}(k) \)
      - Remove \( i \) from \( S_{\text{last}}(\text{start}(k)) \)

Free chains

- As above (pretty much)
- Before increasing chain to load \( L \)
  - \( v = \) Count routes with free space \( \geq L \)
  - \( c = \) Count free chains with load \( \geq L \)
  - if \( c > v \), can’t form chain
Propagation – Time

Time Constraints:

Rooted chains:
• For each route $k$
  – For each visit $i$ in domain of $S_{last}(k)$
    • If vehicle can’t reach $i$ before $E_i$
      – Remove $k$ from $r_i$
      – Remove $i$ from $S_{last}(start(k))$

Free chains:
• Form “implied time window” from chain
• Rest is as above
Savings

Note:

- Clarke-Wright “Savings” method grows chains of visits
- Very successful early method
- Still used in many methods for an initial solution
- Very appropriate for CP
A gotchya: Chronological backtracking

Assign a successor
A gotchya: Chronological backtracking

Assign a successor
Can’t re-assign
• Domains can only shrink
Assign a successor

s₁ = [2,3,4,5,8,9]
s₂ = [1,3,4,5]
s₃ = [1,2,4,5,8,9]
s₄ = [1,3,5,9]
s₅ = [1,2,3,4,8,9]
s₆ = [1,3,5,8]
s₇ = [1,2,3,5]
s₈ = [0]
s₉ = [0]
Assign a successor

\[ s_1 = [3,4,5] \]
\[ s_2 = [1,3,5] \]
\[ s_3 = [1,2,4,5,8,9] \]
\[ s_4 = [3,5,9] \]
\[ s_5 = [1,2,3,4,8,9] \]
\[ s_6 = [3,5,8] \]
\[ s_7 = [2,3,5] \]
\[ s_8 = [0] \]
\[ s_9 = [0] \]
A gotchya: Chronological backtracking

Assume: $T_{i,j} = 1$ for all $i, j$

TWs for 1-5 are [1-3]

$a_1 = [1,2,3]$
$a_2 = [1,2]$
$a_3 = [1,2,3]$
$a_4 = [2,3]$
$a_5 = [1,2,3]$
$a_6 = [0]$
$a_7 = [0]$
$a_8 = [0,1,2,3,4]$
$a_9 = [3,4]$
A gotchya: Chronological backtracking

Assume: $T_{i,j} = 1$ for all $i$, $j$

TWs for 1-5 are are [1-3]

$a_1 = [2]$
$a_2 = [1]$
$a_3 = [1,2,3]$
$a_4 = [3]$
$a_5 = [1,2,3]$
$a_6 = [0]$
$a_7 = [0]$
$a_8 = [0,1,2,3,4]$
$a_9 = [4]$
Propagation – an alternative

- Alternative relies on knowledge of “incumbent” solution
- Use shared data structure
- Can use full insertion (insert into middle of ‘chain’)

Diagram:

- CP System
  - Var/value choice
  - Propagators
- Incumbent Solution
Propagation

Insertion ‘in the middle’

- Propagators know incumbent solution
- Only propagate non-binding implications
- e.g. don’t “set” $s_1$ $s_2$ here:
- But 1 is removed from $s_4$, $s_6$
- Can calculate earliest start, latest start to bound start time
- Can calculate feasible inserts, and update $s$ vars appropriately
- Strong propagation for capacity, time constraints
Example (1 cdyt)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_i$</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$E_i$</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>$L_i$</td>
<td>150</td>
<td>65</td>
<td>150</td>
<td>150</td>
<td>150</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>$S_i$</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>V2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_k$</td>
<td>30</td>
<td>30</td>
</tr>
</tbody>
</table>

+ Request Compatibility

Different Routes: R2, R4
## Initial

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0,2-5, 8,9</td>
<td>0,1, 3-5,8,9</td>
<td>0-2, 4,5,8,9</td>
<td>0-3,5, 8,9</td>
<td>0-4, 8,9</td>
<td>1-5, 8</td>
<td>1-5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>20-150</td>
<td>20-65</td>
<td>20-150</td>
<td>20-150</td>
<td>20-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>0-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>0-30</td>
</tr>
</tbody>
</table>
## Initial

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0,2-5, 8,9</td>
<td>0,1, 3-5,8,9</td>
<td>0-2, 4,5,8,9</td>
<td>0-3,5, 8,9</td>
<td>0-4, 8,9</td>
<td>1-5, 8</td>
<td>1-5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>42-150</td>
<td>50-150</td>
<td>30-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>0-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>0-30</td>
</tr>
</tbody>
</table>

**Initial propagations (arrive time)**
### Initial

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0, 3-5,8,9</td>
<td>0,1, 3-5,8,9</td>
<td>0,2 4,5,8,9</td>
<td>0,1 3,5,8,9</td>
<td>0,1 3,4,8,9</td>
<td>1-5, 8</td>
<td>1-5, 8</td>
<td>1-5, 8</td>
<td>0, 0, 0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>42-150</td>
<td>50-150</td>
<td>30-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>0-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>0-30</td>
</tr>
</tbody>
</table>

**Initial propagations (time windows)**
### Initial propagations (compatibility constraint)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0, 3-5,8,9</td>
<td>0,1, 3,5,8,9</td>
<td>0-2, 4,5,8,9</td>
<td>0,1, 3,5,8,9</td>
<td>0,1, 3,4,8,9</td>
<td>1-5, 8</td>
<td>1-5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>42-150</td>
<td>50-150</td>
<td>30-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>0-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>0-30</td>
</tr>
</tbody>
</table>
Choose R5 after R7 (start V2)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0, 3-5,8,9</td>
<td>0,1, 3,5,8,9</td>
<td>0-2 4,5,8,9</td>
<td>0,1 3,5,8,9</td>
<td>0,1 3,4,8,9</td>
<td>1-5, 8</td>
<td>1-5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>42-150</td>
<td>50-150</td>
<td>30-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>0-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>0-30</td>
</tr>
</tbody>
</table>
## Choose R5 after R7 (start V2)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0, 3-5,8,9</td>
<td>0,1, 3,5,8,9</td>
<td>0-2, 4,5,8,9</td>
<td>0,1, 3,5,8,9</td>
<td>0,1, 3,4,8,9,9</td>
<td>1-4, 8</td>
<td>1-5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>42-150</td>
<td>50-150</td>
<td>30-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>0-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>0-30</td>
</tr>
</tbody>
</table>

Propagate successor implications
Choose R5 after R7 (start V2)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0, 3-5,8,9</td>
<td>0,1, 3,5,8,9</td>
<td>0-2, 4,5,8,9</td>
<td>0,1, 3,5,8,9</td>
<td>0,1, 3,4,9</td>
<td>1-4, 8</td>
<td>1-5</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>42-150</td>
<td>50-150</td>
<td>30-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>60-200</td>
<td></td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>10-30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>10-30</td>
<td></td>
</tr>
</tbody>
</table>

Propagate changes to time and load
Choose R5 after R7 (start V2)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0, 3-5, 8, 9</td>
<td>0, 1, 3, 5, 8, 9</td>
<td>0-2, 4, 5, 8, 9</td>
<td>0, 1, 3, 5, 8, 9</td>
<td>0, 1, 3, 4, 9</td>
<td>1-4, 8</td>
<td>1-5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0, 1, 2</td>
<td>0, 1, 2</td>
<td>0, 1, 2</td>
<td>0, 1, 2</td>
<td>0, 1, 2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>42-150</td>
<td>50-150</td>
<td>30-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>60-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>10-30</td>
</tr>
</tbody>
</table>

Bind route var
Choose R2 after R7 (start V2)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0, 3-5, 8, 9</td>
<td>0, 1, 3, 5, 8, 9</td>
<td>0-2, 4-5, 8, 9</td>
<td>0, 1, 3, 5, 8, 9</td>
<td>0, 1, 3, 4, 9</td>
<td>1-4, 8, 1-5, 0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$r_i$</td>
<td>0, 1, 2,</td>
<td>0, 1, 2,</td>
<td>0, 1, 2,</td>
<td>0, 1, 2,</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150, 32-65</td>
<td>42-150, 50-150</td>
<td>30-150, 0</td>
<td>0</td>
<td>0</td>
<td>0-200, 60-200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>10-30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>10-30</td>
</tr>
</tbody>
</table>
Choose R2 after R7 (start V2)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0,</td>
<td>1,3,5</td>
<td>0-2</td>
<td>0,1</td>
<td>1,3,4,9</td>
<td>1,3,4,8</td>
<td>1-4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3-5,8,9</td>
<td>4,5,8,9</td>
<td>3,5,8,9</td>
<td>9</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>42-150</td>
<td>50-150</td>
<td>30-150</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>0-30</td>
<td>10-30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Propagate successor implications
Choose R2 after R7 (start V2)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0,</td>
<td>1,3,5</td>
<td>0-2</td>
<td>0,1</td>
<td>1,3,4,</td>
<td>1,3,4,</td>
<td>1-4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3-5,8,9</td>
<td>4,5,8,9</td>
<td>3,5,8,9</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>42-150</td>
<td>50-150</td>
<td>78-150</td>
<td>0</td>
<td>0</td>
<td>200</td>
<td>118-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>0-30</td>
<td>0-30</td>
<td>20-30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>20-30</td>
</tr>
</tbody>
</table>

Update load and time
Choose R2 after R7 (start V2)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0, 3-5,8,9</td>
<td>1,3,5</td>
<td>0-2</td>
<td>0,1</td>
<td>1,3,4,9</td>
<td>1,3,4,8</td>
<td>1-4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>42-150</td>
<td>50-150</td>
<td>78-150</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>118-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>0-30</td>
<td>0-30</td>
<td>20-30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20-30</td>
</tr>
</tbody>
</table>

Bind route var
Choose R2 after R7 (start V2)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0, 3-5,8,9</td>
<td>1,3,5</td>
<td>0-2</td>
<td>0,1,3,8</td>
<td>1,3,4,9</td>
<td>1,3,4,9</td>
<td>1-3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1,2</td>
<td>2</td>
<td>0,1,2</td>
<td>0,1,2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>42-150</td>
<td>50-150</td>
<td>78-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>118-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>0-30</td>
<td>0-30</td>
<td>20-30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>20-30</td>
</tr>
</tbody>
</table>

Propagate request compatibility constraint
Choose R3 after R2

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0, 3-5, 8, 9</td>
<td>1, 3, 5</td>
<td>0-2, 4, 5, 8, 9</td>
<td>0, 1, 3, 8</td>
<td>1, 3, 9</td>
<td>1, 3, 4, 8</td>
<td>1-3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0, 1, 2</td>
<td>2</td>
<td>0, 1, 2</td>
<td>0, 1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>42-150</td>
<td>50-150</td>
<td>78-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>118-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>0-30</td>
<td>0-30</td>
<td>20-30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>20-30</td>
</tr>
</tbody>
</table>
Choose R3 after R2

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0, 3-5, 8, 9</td>
<td>1, 3, 5</td>
<td>1, 4, 5</td>
<td>0, 1, 3, 8</td>
<td>1, 3, 9</td>
<td>1, 3, 4, 8</td>
<td>1, 2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0, 1, 2</td>
<td>2</td>
<td>0, 1, 2</td>
<td>0, 1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>42-150</td>
<td>50-150</td>
<td>78-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>118-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>0-30</td>
<td>0-30</td>
<td>20-30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>20-30</td>
</tr>
</tbody>
</table>

Propagate successor implications
## Choose R3 after R2

<table>
<thead>
<tr>
<th>$s_i$</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 3-5,8,9</td>
<td>1,3</td>
<td>1,4,5</td>
<td>0,1,3,8</td>
<td>1,9</td>
<td>1,4,8</td>
<td>1,2</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1,2</td>
<td>2</td>
<td>0,1,2</td>
<td>0,1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>62-110</td>
<td>50-150</td>
<td>102-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>142-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>20-30</td>
<td>0-30</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>30</td>
</tr>
</tbody>
</table>

**Update time and load**
Choose R3 after R2

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0, 3-5, 8, 9</td>
<td>1, 3</td>
<td>1, 4, 5</td>
<td>0, 1, 3, 8</td>
<td>1, 9</td>
<td>1, 4, 8</td>
<td>1, 2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0, 1, 2</td>
<td>2</td>
<td>0, 1, 2</td>
<td>0, 1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>62-110</td>
<td>50-150</td>
<td>102-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>142-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>20-30</td>
<td>0-30</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>30</td>
</tr>
</tbody>
</table>

Bind route var
Choose R3 after R2

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0, 3-5,8,9</td>
<td>1,3</td>
<td>1,4,5</td>
<td>0,1,8</td>
<td>1,9</td>
<td>1,4,8</td>
<td>1,2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1,2</td>
<td>2</td>
<td>2</td>
<td>0,1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>62-110</td>
<td>50-150</td>
<td>102-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>142-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>20-30</td>
<td>0-30</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>30</td>
</tr>
</tbody>
</table>

Propagate request incompatibility constraint
Choose R3 after R2

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0,4,8,3</td>
<td>5</td>
<td>0,1,8</td>
<td>9</td>
<td>1,4,8</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1,8</td>
<td>2</td>
<td>2</td>
<td>0,1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>62-110</td>
<td>50-150</td>
<td>102-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>142-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>20-30</td>
<td>0-30</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>30</td>
</tr>
</tbody>
</table>

Propagate effects of full load
Choose R4 after R6 (start V1)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0,4,8</td>
<td>3</td>
<td>5</td>
<td>0,1,8</td>
<td>9</td>
<td>1,4,8</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1</td>
<td>2</td>
<td>2</td>
<td>0,1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>62-110</td>
<td>50-150</td>
<td>102-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>142-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>20-30</td>
<td>0-30</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>30</td>
</tr>
</tbody>
</table>
Choose R4 after R6 (start V1)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0,4,8</td>
<td>3</td>
<td>5</td>
<td>1,8</td>
<td>9</td>
<td>1,4,8</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1</td>
<td>2</td>
<td>2</td>
<td>0,1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>62-110</td>
<td>50-150</td>
<td>102-150</td>
<td>0</td>
<td>0</td>
<td>0-200</td>
<td>142-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>20-30</td>
<td>0-30</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0-30</td>
<td>30</td>
</tr>
</tbody>
</table>

Propagate successor implications
Choose R4 after R6 (start V1)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0,4,8</td>
<td>3</td>
<td>5</td>
<td>1,8</td>
<td>9</td>
<td>1,4</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1</td>
<td>2</td>
<td>2</td>
<td>0,1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>62-110</td>
<td>50-140</td>
<td>102-150</td>
<td>0</td>
<td>0</td>
<td>110-200</td>
<td>142-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>20-30</td>
<td>10-30</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>10-30</td>
<td>30</td>
</tr>
</tbody>
</table>

Update time and load
Choose R4 after R6 (start V1)
Choose R4 after R6 (start V1)

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0,4,8</td>
<td>3</td>
<td>5</td>
<td>1,8</td>
<td>9</td>
<td>1,4</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>62-110</td>
<td>50-140</td>
<td>102-150</td>
<td>0</td>
<td>0</td>
<td>110-200</td>
<td>142-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>20-30</td>
<td>10-30</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>10-30</td>
<td>30</td>
</tr>
</tbody>
</table>

Bind route var
Choose R1 after R4

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0.4,8</td>
<td>3</td>
<td>5</td>
<td>1,8</td>
<td>9</td>
<td>1.4</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>62-110</td>
<td>50-140</td>
<td>102-150</td>
<td>0</td>
<td>0</td>
<td>110-200</td>
<td>142-200</td>
</tr>
<tr>
<td>$a_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>20-30</td>
<td>10-30</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>10-30</td>
<td>30</td>
</tr>
</tbody>
</table>
Choose R1 after R4

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>0,4,8</td>
<td>3</td>
<td>5</td>
<td>1,6</td>
<td>9</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0,1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>51-150</td>
<td>32-65</td>
<td>62-110</td>
<td>50-140</td>
<td>102-150</td>
<td>0</td>
<td>0</td>
<td>110-200</td>
<td>142-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>0-30</td>
<td>10-30</td>
<td>20-30</td>
<td>10-30</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>10-30</td>
<td>30</td>
</tr>
</tbody>
</table>

Propagate successor implications
Choose R1 after R4

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>9</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>70-139</td>
<td>32-65</td>
<td>62-110</td>
<td>50-119</td>
<td>102-150</td>
<td>0</td>
<td>0</td>
<td>131-200</td>
<td>142-200</td>
</tr>
<tr>
<td>$q_i$</td>
<td>20-30</td>
<td>10-30</td>
<td>20-30</td>
<td>10-30</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>20-30</td>
<td>30</td>
</tr>
</tbody>
</table>

Update load and time, and route var
Choose R1 after R4
Choose R1 after R4

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s_i)</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>9</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(r_i)</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(t_i)</td>
<td>70-139</td>
<td>32-65</td>
<td>62-110</td>
<td>50-119</td>
<td>102-150</td>
<td>0</td>
<td>0</td>
<td>131-200</td>
<td>142-200</td>
</tr>
<tr>
<td>(q_i)</td>
<td>20-30</td>
<td>10-30</td>
<td>20-30</td>
<td>10-30</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>20-30</td>
<td>30</td>
</tr>
</tbody>
</table>

Update load and time, and route var
Choose R1 after R4

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>9</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_i$</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t_i$</td>
<td>70</td>
<td>32</td>
<td>62</td>
<td>50</td>
<td>102</td>
<td>0</td>
<td>0</td>
<td>131</td>
<td>142</td>
</tr>
<tr>
<td>$q_i$</td>
<td>20</td>
<td>10</td>
<td>20</td>
<td>10</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

FINISHED! Bind remaining vars to min val
Data

We know (note uppercase)

- $V_i$: The ‘value’ of customer $i$
- $D_{ik}$: Demand by customer $i$ for commodity $k$
- $E_i$: Earliest time to start service at $i$
- $L_i$: Latest time to start service at $i$
- $S_i$: Service time of visit at $i$
- $Q_{jk}$: Capacity of vehicle $j$ for commodity $k$
- $T_{ij}$: Travel time from visit $i$ to visit $j$
- $C_{ij}$: Cost (w.r.t. objective) of travel from $i$ to $j$
Insertion

• Allows for maximum propagation
• Allows constraints to influence solution progressively

• e.g. Blood delivery constraints
  – Delivery within 20 minutes of pickup
  – As soon as one is fixed implication flows to other

• e.g. Driver break
  – Extra request (*a-priori*) with time constraints that relate to other breaks
  – Special propagator that removes requests that cannot be inserted in a route without violating rules
Local Search

- Can apply std local search methods VRPs
  - k-opt
  - Or-opt
  - exchange
  - ...
- Step from solution to solution, so
- CP is only used as rule-checker
  - Little use of propagation

- (Constraint-based local search (e.g. Invariants) not yet widely available)
Large Neighbourhood Search

• LNS destroys then re-creates

• Creation methods can leverage propagation

• LNS *can* use the full power of CP
Large Neighbourhood Search revisited
Large Neighbourhood Search

Destroy & Re-create

- Destroy part of the solution
  - Remove visits from the solution
- Re-create solution
  - Use favourite construct method to re-insert customers
- If the solution is better, keep it
- Repeat
Large Neighbourhood Search

Destroy part of the solution (Select method)

In CP terms, this means:

• Relax some variable assignments

In CP-VRP terms, this means

• Relax some successor assignments, i.e.
• ‘Unassign” some visits.
Large Neighbourhood Search

Destroy part of the solution (*Select* method)

Examples

- Remove a sequence of visits
Large Neighbourhood Search

Destroy part of the solution (Select method)

Examples

- Choose longest (worst) arc in solution
  - Remove visits at each end
  - Remove nearby visits
- Actually, choose \( r^{th} \) worst
- \( r = n \ast (\text{uniform}(0,1))^y \)
- \( y \sim 6 \)
  - \( 0.5^6 = 0.016 \)
  - \( 0.9^6 = 0.531 \)
Large Neighbourhood Search

Destroy part of the solution (*Select* method)

Examples

- Dump visits from $k$ routes ($k = 1, 2, 3$)
  - Prefer routes that are close,
  - Better yet, overlapping
Large Neighbourhood Search

Destroy part of the solution (*Select* method)

Examples

- Choose first visit randomly
- Then, remove “related” visits
  - Based on distance, time compatibility, load

\[ R_{ij} = \varphi C_{ij} + \chi(|a_i - a_j|) + \psi(|q_i - q_j|) \]
Large Neighbourhood Search

Destroy part of the solution (*Select* method)

Examples

- Dump visits from the smallest route
  - Good if saving vehicles
  - Sometimes fewer vehicles = reduced travel
Large Neighbourhood
Search

Destroy part of the solution (*Select* method)

- Parameter: Max to dump
  - As a % of $n$?
  - As a fixed number e.g. 100 for large problems

- Actual number is uniform rand (5, $max$)
Large Neighbourhood Search

Re-create solution

• Use complete search
  – for small $n$, or highly-constrained problems
  – Works for assign-to-successor search

• Use semi-complete search
  – If you have a heuristic you trust, use Limited Discrepancy Search
  – Depth-bounded search
  – Fail-bounded
  – Time-bounded
Large Neighbourhood Search

Re-create solution

- Insertion methods are good creation methods
- Also make good re-create methods for LNS

- Use insert method to guide Limited Discrepancy Search

- Use a portfolio of insert methods
  - Diversify search
Large Neighbourhood Search

Re-create solution

- Insert methods differ by choice of:
  - Which visit to insert
  - Where to insert it

- Where to insert:
  - Usually in position that increases cost by least
  - Also consider ‘spare time’ – choose position that maintains the maximum spare time

- Which to insert:
  - Many choices
Recreate solution

Which to insert

• Examples:
  – Nearest Neighbour
    • Unassigned visit closest to last visit
  – Random
    • Choose visit to insert at random
  – Minimum insert cost
    • Choose visit that increases cost by the least
  – Regret, 3-Regret, 4-Regret
    • Choose visit with maximum difference between first and next best insert position
Regret
Regret
Regret
Regret
Regret
Regret

\[ \text{Regret} = C(\text{insert in 2^{nd}-best route}) - C(\text{insert in best route}) \]
\[ = f(2, i) - f(1, i) \]

\[ \text{K-Regret} = \sum_{k=1}^{K} (f(k, i) - f(1, i)) \]
Large Neighbourhood Search

If the solution is better, keep it

- Can use Hill-climbing
- Can use Simulated Annealing
- Can use Threshold Annealing
- ...

Repeat

- Can use fail limit (limit on number of infeasibilities found)
- Can use time limit
- Can use Restarts
- Can use limited number of iterations
Advanced techniques

Randomized Adaptive Decomposition

- Partition problem into subproblems
- Work on each subproblem in turn
- Decomposition ‘adapts’
  - Changes in response to incumbent solution
Solving VRPs

- CP is “natural” for solving vehicle routing problems
  - Real problems often have unique constraints
  - Easy to change CP model to include new constraints
  - New constraints don’t change core solve method
  - Infrastructure for complete (completish) search in subproblems

- LNS is “natural” for CP
  - Insertion leverages propagation

oh, and one more thing
Announcement

NICTA is hiring

• Seeking recent PhD with good publication record
• Work in Logistics and Supply Chain
• Work with Pascal van Hentenryck + me
• Research + Software development
• Lots of Constraint Programming
• Work on next-generation CP system
• Canberra-based

• See NICTA web pages, or talk to me later
  www.nicta.com.au