Investigating the Dynamic Pickup and Delivery Problem with Time Windows

Penny Holborn, Jonathan Thompson and Rhyd Lewis

Cardiff University

OR 54 September, 2012
Outline of the talk

1. Introduction
2. Initial investigations
3. The Dynamic Problem
4. Results
5. Future Work
The pickup and delivery problem with time windows

Penny Holborn  Investigating the DPDPTW
The pickup and delivery problem with time windows

- **Problem**
  - Requests
  - Fleet of vehicles
  - Depot
The pickup and delivery problem with time windows

- **Problem**
  - Requests
  - Fleet of vehicles
  - Depot
- **Request**
  - Pickup location
  - Delivery location
  - Load
  - Time window
  - Service time
The pickup and delivery problem with time windows

- **Problem**
  - Requests
  - Fleet of vehicles
  - Depot
- **Request**
  - Pickup location
  - Delivery location
  - Load
  - Time window
  - Service time
- **Constraints**
  - Capacity
  - Precedence
  - Planning horizon

Penny Holborn
Investigating the DPDPTW
The pickup and delivery problem with time windows

- **Problem**
  - Requests
  - Fleet of vehicles
  - Depot
- **Request**
  - Pickup location
  - Delivery location
  - Load
  - Time window
  - Service time
- **Constraints**
  - Capacity
  - Precedence
  - Planning horizon
- **Objective**
  - Minimise the total travel cost
The dynamic problem

The dynamic problem is one where requests are not completely known in advance, but become available during the planning period. To solve the problem:

1. Split the planning horizon into set intervals.
2. For each interval have a static sub-problem to solve.
3. At the start of each interval, restart the algorithm taking into account all known information.

Start by solving the static pickup and delivery problem with time windows (PDPTW).
The dynamic problem

The **dynamic problem** is one where requests are not completely known in advance, but become available during the planning period.
The dynamic problem

The **dynamic problem** is one where requests are not completely known in advance, but become available during the planning period.

To solve the problem:
The dynamic problem

- The **dynamic problem** is one where requests are not completely known in advance, but become available during the planning period.

- To solve the problem:
  - Split the planning horizon into set intervals
  - For each interval have a static sub-problem to solve
  - At the start of each interval restart the algorithm taking into account all known information
The dynamic problem

- The **dynamic problem** is one where requests are not completely known in advance, but become available during the planning period.

- To solve the problem:
  - Split the planning horizon into set intervals
  - For each interval have a static sub-problem to solve
  - At the start of each interval restart the algorithm taking into account all known information

- Start by solving the static pickup and delivery problem with time windows (PDPTW)
Initial investigations

An initial solution was constructed using a combination of both a random and greedy heuristic. Three new reconstruction heuristics were developed to improve on the local search operators of Li & Lim, 2001:

1. Single move of a request within a route
2. Single route reconstruction
3. Multiple route reconstruction

Penny Holborn
Investigating the DPDPTW
Initial investigations

An initial solution was constructed using a combination of both a random and greedy heuristic.
Initial investigations

- An initial solution was constructed using a combination of both a random and greedy heuristic.

- 3 new reconstruction heuristics were developed to improve on the local search operators of Li & Lim, 2001.
Initial investigations

- An initial solution was constructed using a combination of both a random and greedy heuristic.

- 3 new **reconstruction heuristics** were developed to improve on the local search operators of **Li & Lim, 2001**.
  - Single move of a request within a route
  - Single route reconstruction
  - Multiple route reconstruction
Tabu Search

A tabu search heuristic was added to a simple move operator.

The length of the tabu list is based on the number of requests.

Attribute chosen to store in the tabu list:

- The pickup and delivery edges removed and inserted in a solution.

The search is stopped when no more feasible moves can be made.
A tabu search heuristic was added to a simple move operator.
Tabu Search

- A tabu search heuristic was added to a simple move operator
- The length of the tabu list is based on the number of requests
Tabu Search

- A tabu search heuristic was added to a simple move operator

- The length of the tabu list is based on the number of requests

Attribute chosen to store in the tabu list:
Tabu Search

- A tabu search heuristic was added to a simple move operator

- The length of the tabu list is based on the number of requests

Attribute chosen to store in the tabu list:
- The pickup and delivery edges removed and inserted in a solution
Tabu Search

- A tabu search heuristic was added to a simple move operator
- The length of the tabu list is based on the number of requests

Attribute chosen to store in the tabu list:
- The pickup and delivery edges removed and inserted in a solution
- The search is stopped when no more feasible moves can be made
Tabu search attribute

Edges stored in the tabu list

Route 1
Depot → A → B → C → D → Depot
Route 2
Depot → E → F → G → H → Depot

Route 1
Depot → A → B → C → D → Depot
Route 2
Depot → E → F → G → H → Depot
Having dealt with the grouping aspect of the problem it is now the ordering of the locations within a route which is to be considered. A branch and bound strategy is implemented to work within each route. The method is computationally expensive so is applied to sub-sections of routes. Results show sub-sections with up to 14 locations require less than a second of computational time. The size of the search tree can be limited by: time window constraints, pickup/delivery precedence constraints, using heuristic methods to search the branches.
Having dealt with the grouping aspect of the problem it is now the ordering of the locations within a route which is to be considered.
Branch and bound strategy

- Having dealt with the grouping aspect of the problem it is now the ordering of the locations within a route which is to be considered.

- A **branch and bound** strategy is implemented to work within each route.

Results show sub-sections with up to 14 locations require less than a second of computational time.
Branch and bound strategy

- Having dealt with the grouping aspect of the problem it is now the ordering of the locations within a route which is to be considered.

- A **branch and bound** strategy is implemented to work within each route.
  - The method is computationally expensive so is applied to sub-sections of routes.
Branch and bound strategy

Having dealt with the grouping aspect of the problem it is now the ordering of the locations within a route which is to be considered.

A **branch and bound** strategy is implemented to work within each route:
- The method is computationally expensive so is applied to sub-sections of routes.
- Results show sub-sections with up to 14 locations require less than a second of computational time.
Branch and bound strategy

- Having dealt with the grouping aspect of the problem it is now the ordering of the locations within a route which is to be considered.

- A **branch and bound** strategy is implemented to work within each route:
  - The method is computationally expensive so is applied to sub-sections of routes.
  - Results show sub-sections with up to 14 locations require less than a second of computational time.

- The size of the search tree can be limited by:
Having dealt with the grouping aspect of the problem it is now the ordering of the locations within a route which is to be considered.

A branch and bound strategy is implemented to work within each route:
- The method is computationally expensive so is applied to sub-sections of routes.
- Results show sub-sections with up to 14 locations require less than a second of computational time.

The size of the search tree can be limited by:
- Time window constraints, pickup/delivery precedence constraints, using heuristic methods to search the branches.
Benchmark data sets for the PDPTW

56 static instances provided by Li & Lim, 2001
Based on Solomons 56 100-customer instances (Solomon, 1987)
Instances organised into 6 classes
Locations are clustered, clustered/random or random
Some have a short scheduling horizon, others a longer scheduling horizon
51/56 best know solutions were found
Equals the number of best known solutions found by Dergis et al., 2004
Results provided in Holborn et al., 2012
BENCHMARK DATA SETS FOR THE PDPTW

- 56 static instances provided by Li & Lim, 2001
- Based on Solomons 56 100-customer instances (Solomon, 1987)
Benchmark data sets for the PDPTW

- 56 static instances provided by Li & Lim, 2001
- Based on Solomons 56 100-customer instances (Solomon, 1987)
- Instances organised into 6 classes
  - Locations are clustered, clustered/random or random
  - Some have a short scheduling horizon, others a longer scheduling horizon
Benchmark data sets for the PDPTW

- 56 static instances provided by Li & Lim, 2001
- Based on Solomons 56 100-customer instances (Solomon, 1987)
- Instances organised into 6 classes
  - Locations are clustered, clustered/random or random
  - Some have a short scheduling horizon, others a longer scheduling horizon

- 51/56 best known solutions were found
- Equals the number of best known solutions found by Dergis et al., 2004
Benchmark data sets for the PDPTW

- 56 static instances provided by Li & Lim, 2001
- Based on Solomons 56 100-customer instances (Solomon, 1987)
- Instances organised into 6 classes
  - Locations are clustered, clustered/random or random
  - Some have a short scheduling horizon, others a longer scheduling horizon
- 51/56 best known solutions were found
- Equals the number of best known solutions found by Dergis et al., 2004
- Results provided in Holborn et al., 2012
The dynamic problem

Things to consider for the dynamic problem:
- What is the optimal time between intervals?
- How should the new requests be inserted?
- Which locations of a route should be considered fixed?
- Which improvement methods should be applied?
The dynamic problem

- Things to consider for the dynamic problem:
  - What is the optimal time between intervals?
The dynamic problem

- Things to consider for the dynamic problem:
  - What is the optimal time between intervals?
  - How should the new requests be inserted?
The dynamic problem

- Things to consider for the dynamic problem:
  - What is the optimal time between intervals?
  - How should the new requests be inserted?
  - Which locations of a route should be considered fixed?
The dynamic problem

- Things to consider for the dynamic problem:
  - What is the optimal time between intervals?
  - How should the new requests be inserted?
  - Which locations of a route should be considered fixed?
  - Which improvement methods should be applied?
Benchmark datasets

Datasets of Mitrovic-Minic et al., 2004 were generated to emulate real-life data. Instances contain 100, 500 and 1000 requests, 30 instances of each problem size. Vehicle capacity is now negligible and return distance to the depot is no longer considered.
Datasets of Mitrovic-Minic et al., 2004 were generated to emulate real-life data.
Benchmark datasets

- Datasets of Mitrovic-Minic et al., 2004 were generated to emulate real-life data

- Instances contain 100, 500 and 1000 requests, 30 instances of each problem size
Benchmark datasets

- Datasets of Mitrovic-Minic et al., 2004 were generated to emulate real-life data
- Instances contain 100, 500 and 1000 requests, 30 instances of each problem size
- Vehicle capacity is now negligible and return distance to the depot is no longer considered
Benchmark datasets

- Datasets of Mitrovic-Minic et al., 2004 were generated to emulate real-life data

- Instances contain 100, 500 and 1000 requests, 30 instances of each problem size

- Vehicle capacity is now negligible and return distance to the depot is no longer considered

- 10 hour day partitioned into time intervals of 15 minutes
Our algorithm

- Static requests
  - Start of planning horizon
  - Initial solution

- New requests
  - Start of interval
  - Update solution
  - Output solution
  - End of interval
    - Restart algorithm
  - End of planning horizon
Insertion heuristics

In total 9 different insertion heuristics were examined. The method of insertion included:

- Random
- Greedy
- Slack

The criteria for insertion included:

- New requests
- New requests and non-fixed requests
- New requests and all non-fixed locations
Insertion heuristics

- In total 9 different insertion heuristics were examined
Insertion heuristics

- In total 9 different insertion heuristics were examined
- The method of insertion included:
Insertion heuristics

- In total 9 different insertion heuristics were examined

- The method of insertion included:
  - Random
  - Greedy
  - Slack
Insertion heuristics

- In total 9 different insertion heuristics were examined

- The method of insertion included:
  - Random
  - Greedy
  - Slack

- The criteria for insertion included:
Insertion heuristics

- In total 9 different insertion heuristics were examined

- The method of insertion included:
  - Random
  - Greedy
  - Slack

- The criteria for insertion included:
  - New requests
  - New requests and non-fixed requests
  - New requests and all non-fixed locations
Results of insertion heuristics
Observations

Overall, a simple random insertion seems to work best and is most practical in real-life. Characteristics of good solutions include:

- Less vehicles
- Less waiting time
- Lower number of fixed locations at each interval

It is interesting to note that the best solution at an interval early within the planning horizon does not always lead to the best solution at the end of the planning horizon.
Observations

- Overall a simple random insertion seems to work best and is most practical in real-life
Overall a simple random insertion seems to work best and is most practical in real-life.

Characteristics of good solutions include:

- Less vehicles
- Less waiting time
- Lower number of fixed locations at each interval

It is interesting to note that the best solution at an interval early within the planning horizon does not always lead to the best solution at the end of the planning horizon.
Observations

- Overall a simple random insertion seems to work best and is most practical in real-life.

- Characteristics of good solutions include:
  - Less vehicles
  - Less waiting time
  - Lower number of fixed locations at each interval
Observations

- Overall a simple random insertion seems to work best and is most practical in real-life.

- Characteristics of good solutions include:
  - Less vehicles
  - Less waiting time
  - Lower number of fixed locations at each interval

- It is interesting to note that the best solution at an interval early within the planning horizon does not always lead to the best solution at the end of the planning horizon.
## Results

<table>
<thead>
<tr>
<th>No. requests</th>
<th>Best found</th>
<th>Mitrovic-Minic</th>
<th>%Diff. dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>2626.53</td>
<td>2755.70</td>
<td>5%</td>
</tr>
<tr>
<td>500</td>
<td>8496036</td>
<td>1308.09</td>
<td>13%</td>
</tr>
<tr>
<td>1000</td>
<td>14140.69</td>
<td>17610.45</td>
<td>20%</td>
</tr>
</tbody>
</table>

![Bar chart showing total distance travelled vs number of requests](chart.png)

*Investigating the DPDPTW*
Health Courier Service

Pickup and delivery
Mail, equipment, samples, x-rays
Between hospitals, GP health centres, clinics

Time windows
Blood has a 2 hour life span, this is prioritised
Toxicology sample can be refrigerated over night

Objective
Minimise cost function, employee costs, mileage, fuel costs

Penny Holborn
Investigating the DPDPTW
Health Courier Service

- Pickup and delivery

- Objective
  - Minimise cost function, employee costs, mileage, fuel costs

Penny Holborn
Investigating the DPDPTW
Health Courier Service

- **Pickup and delivery**
  - Mail, equipment, samples, x-rays
  - Between hospitals, GP health centres, clinics
Health Courier Service

- **Pickup and delivery**
  - Mail, equipment, samples, x-rays
  - Between hospitals, GP health centres, clinics

- **Time windows**
Health Courier Service

- **Pickup and delivery**
  - Mail, equipment, samples, x-rays
  - Between hospitals, GP health centres, clinics

- **Time windows**
  - Blood has a 2 hour life span, this is prioritised
  - Toxicology sample can be refrigerated over night
Health Courier Service

- **Pickup and delivery**
  - Mail, equipment, samples, x-rays
  - Between hospitals, GP health centres, clinics

- **Time windows**
  - Blood has a 2 hour life span, this is prioritised
  - Toxicology sample can be refrigerated over night

- **Objective**
Health Courier Service

- **Pickup and delivery**
  - Mail, equipment, samples, x-rays
  - Between hospitals, GP health centres, clinics

- **Time windows**
  - Blood has a 2 hour life span, this is prioritised
  - Toxicology sample can be refrigerated over night

- **Objective**
  - Minimise cost function, employee costs, mileage, fuel costs
HCS Problems

Penny Holborn Investigating the DPDPTW
HCS Problems

- Static problem
HCS Problems

- **Static problem**
  - 80% of daily locations scheduled in advance and manually
  - Approximately 160 locations per day
  - Repeated routes around major hospitals and clinics
Daily schedules
HCS Problems

- **Static problem**
  - 80% of daily locations scheduled in advance and manually
  - Approximately 160 locations per day
  - Repeated routes around major hospitals and clinics

- **Dynamic Problem**
HCS Problems

- **Static problem**
  - 80% of daily locations scheduled in advance and manually
  - Approximately 160 locations per day
  - Repeated routes around major hospitals and clinics

- **Dynamic Problem**
  - 24/7 ad-hoc service
  - Have 1 vehicle whose sole use is for 24/7 calls mainly for priority calls
  - All other requests need to be scheduled into the existing routes
  - Not all requests are met
Thank you

U. Dergis and T. Dohmer.
Indirect search for the vehicle routing problem with pickup and delivery and time windows.

P. L. Holborn, J.M. Thompson, and R. Lewis.
Combining heuristic and exact methods to solve the vehicle routing problem with pickups, deliveries and time windows.

H. Li and A. Lim.
A metaheuristic for the pickup and delivery problem with time windows.

Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows.

M. M. Solomon.
Algorithms for the vehicle routing and scheduling problems with time window constraints.

holbornpl@cardiff.ac.uk