The Pickup and Delivery Traveling Salesman Problem with FIFO Loading

Jean-François Cordeau, CIRRELT, HEC Montréal, Canada
Mauro Dell’Amico, DISMI, Università di Modena e Reggio Emilia, Italy
Manuel Iori, DISMI, Università di Modena e Reggio Emilia, Italy

ROUTE 2009
Outline

1 Introduction
2 Formulation and valid inequalities
3 Branch-and-cut algorithm
4 Computational results
The TSP with Pickup and Delivery (TSPPD)

Set of $n$ requests with

- origin $i$ where a load must be picked up
- destination $n + i$ where the load must be delivered
The TSP with Pickup and Delivery (TSPPD)

Set of $n$ requests with
- origin $i$ where a load must be picked up
- destination $n + i$ where the load must be delivered

Complete weighted directed graph $G = (N, A)$ where
- $P = \{1, \ldots, n\}$
- $D = \{n + 1, \ldots, 2n\}$
- $N = P \cup D \cup \{0, 2n + 1\}$
The TSP with Pickup and Delivery (TSPPD)

Set of $n$ requests with
- origin $i$ where a load must be picked up
- destination $n + i$ where the load must be delivered

Complete weighted directed graph $G = (N, A)$ where
- $P = \{1, \ldots, n\}$
- $D = \{n + 1, \ldots, 2n\}$
- $N = P \cup D \cup \{0, 2n + 1\}$

Problem: Find the shortest Hamiltonian path from 0 to $2n + 1$ such that node $i$ is visited before node $n + i$ for every request $i$
The TSPPD with LIFO Loading (TSPPD-L)

Pickups and deliveries must be performed according to a Last-In-First-Out (LIFO) policy:

- a load being picked up is always placed on top of the stack
- a delivery can be performed only if the associated load is at the top of the stack
The TSPPD with FIFO Loading (TSPPPDF)

Pickups and deliveries must be performed according to a First-In-First-Out (FIFO) policy:

- a load being picked up is always placed on top of the stack
- a delivery can be performed only if the associated load is at the bottom of the stack
The three problems in a glance

Several applications in practice

- Routing of vehicles to transport freight/persons in which loading requirements cause the additional constraints
The three problems in a glance

Several applications in practice

- Routing of vehicles to transport freight/persons in which loading requirements cause the additional constraints

Different combinatorial structure and difficulty

<table>
<thead>
<tr>
<th></th>
<th>TSPPD</th>
<th>TSPPDL</th>
<th>TSPPDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smallest unsolved instance ( (n) )</td>
<td>25</td>
<td>22</td>
<td>15</td>
</tr>
<tr>
<td>Largest instance solved ( (n) )</td>
<td>100</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>
The TSPPDF literature

Erdogan, Cordeau and Laporte (2007)

- Heuristic and compact mathematical model
- Largest instance solved by the model has $n = 11$
- Smallest instance unsolved by the model has $n = 11$
The TSPPDF literature

Erdogan, Cordeau and Laporte (2007)
- Heuristic and compact mathematical model
- Largest instance solved by the model has $n = 11$
- Smallest instance unsolved by the model has $n = 11$

Carrabs, Cerulli and Cordeau (2008)
- Additive Branch-and-Bound
- Largest instance solved has $n = 19$
- Smallest instance unsolved has $n = 13$
Motivations for a branch-and-cut algorithm

Other approaches are not promising

- Compact mathematical models seem slow
- Branch-and-bound approaches seem slow
- Dynamic programming is not likely to be successful
Motivations for a branch-and-cut algorithm

Other approaches are not promising
- Compact mathematical models seem slow
- Branch-and-bound approaches seem slow
- Dynamic programming is not likely to be successful

Branch-and-cut successful on several related problems
- TSPPD: Ruland and Rodin (1997), Dumitrescu et al. (2006)
Motivations for a branch-and-cut algorithm

Other approaches are not promising
- Compact mathematical models seem slow
- Branch-and-bound approaches seem slow
- Dynamic programming is not likely to be successful

Branch-and-cut successful on several related problems
- TSPPD: Ruland and Rodin (1997), Dumitrescu et al. (2006)

FIFO requirements lead to many (nice) valid inequalities
- In this work we propose 17 families of new inequalities
- This helps in improving LBs
TSPPD formulation

Min \sum_{(i,j) \in A} c_{ij}x_{ij}

subject to

x(\delta^{+}(i)) = 1 \quad \forall i \in P \cup D \cup \{0\}

x(\delta^{-}(i)) = 1 \quad \forall i \in P \cup D \cup \{2n + 1\}

x(S) \leq |S| - 1 \quad \forall S \subseteq P \cup D, |S| \geq 2

x(S) \leq |S| - 2 \quad \forall S \in S

x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A
Precedence constraints (Ruland, 1995)

\[ S = \{ S \subseteq N | 0 \in S, 2n + 1 \notin S \text{ and } \exists i \in P : n + i \in S, i \notin S \} \]

\[ x(S) \leq |S| - 2 \quad \forall S \in S \]
TSPPDF formulation

\[
\text{Min} \quad \sum_{(i,j) \in A} c_{ij} x_{ij}
\]

subject to

\[
\begin{align*}
  x(\delta^+(i)) &= 1 & \forall i \in P \cup D \cup \{0\} \\
  x(\delta^-(i)) &= 1 & \forall i \in P \cup D \cup \{2n + 1\} \\
  x(S) &\leq |S| - 1 & \forall S \subseteq P \cup D, |S| \geq 2 \\
  x(S) &\leq |S| - 2 & \forall S \in S \\
  x(i, S) + x(S) + x(S, n + i) &\leq |S| & \forall S \in \Omega, \forall i \in P : i, n + i \not\in S \\
  x_{ij} &\in \{0, 1\} & \forall (i, j) \in A
\end{align*}
\]
Imposing FIFO constraint on the $x_{ij}$ variables

$$\Omega = \{S \subset P \cup D \, | \exists j \in P : j, n+j \in S\}$$

$$x(i, S) + x(S) + x(S, n+i) \leq |S| \quad \forall S \in \Omega, \forall i \in P : i, n+i \notin S$$
Imposing FIFO constraint on the $x_{ij}$ variables

$$\Omega = \{S \subseteq P \cup D \mid \exists j \in P : j, n+j \in S\}$$

$$x(i, S) + x(S) + x(S, n+i) \leq |S| \quad \forall S \in \Omega, \forall i \in P : i, n+i \not\in S$$

The inequality may be lifted by:

$$x(i, S) + x(S) + x(S, n+i) + x_{i,n+i} \leq |S| \quad \forall S \in \Omega, \forall i \in P : i, n+i \not\in S$$
Valid inequalities for the TSPPDF

Known inequalities for the TSPPD are also valid here:

- $\pi$ and $\sigma$ inequalities [Balas, Fischetti and Pulleyblank, 1995]
- Generalized order constraints [Ruland and Rodin, 1997]
- Lifted $D^+$ and $D^-$ inequalities [Cordeau, 2006]
Valid inequalities for the TSPPDF

Known inequalities for the TSPPD are also valid here:
- $\pi$ and $\sigma$ inequalities [Balas, Fischetti and Pulleyblank, 1995]
- Generalized order constraints [Ruland and Rodin, 1997]
- Lifted $D^+$ and $D^-$ inequalities [Cordeau, 2006]

New inequalities:
- Alternative FIFO inequalities
- Simple FIFO inequalities
- Shutter (cycle) inequalities
Alternative FIFO inequalities

\[ \Omega' = \{ S \subset P \cup D \mid \exists i, j \in P : i, n + j \notin S \text{ and } |S \cap \{j, n + i\}| = 1 \} \]

\[ x(i, S) + x(S) + x(S, n+j) + x_i, n+j \leq |S| \quad \forall S \in \Omega', \forall i \in P : i, n+i \notin S \]
Alternative FIFO inequalities

\[ \Omega' = \{ S \subset P \cup D | \exists i, j \in P : i, n + j \notin S \text{ and } |S \cap \{j, n + i\}| = 1 \} \]

\[ x(i, S) + x(S) + x(S, n+j) + x_{i,n+j} \leq |S| \quad \forall S \in \Omega', \forall i \in P : i, n+i \notin S \]
Simple FIFO inequalities

This part contains 10 families of valid inequalities!
Here we provide just an example.
Simple FIFO inequalities

This part contains 10 families of valid inequalities!
Here we provide just an example.

Whenever an arc \((j, n + j)\) is used, it follows from the FIFO policy that the vehicle arrives empty at \(j\) and leaves empty from \(n + j\).
Simple FIFO inequalities

This part contains 10 families of valid inequalities!
Here we provide just an example.

Whenever an arc \((j, n+j)\) is used, it follows from the FIFO policy that the vehicle arrives empty at \(j\) and leaves empty from \(n+j\).

For any node \(j \in P\) and any set \(H \subseteq P \setminus \{j\}\) the following inequality holds for the TSPPDF:

\[
\sum_{i \in H} x_{ij} + x_{j,n+j} + \sum_{n+h \in \sigma(P \setminus H)} x_{n+h,n+j} \leq 1
\]
Shutter inequalities

\[ x_{ij} + x_{n+j,n+i} + x_{n+j,i} + x_{n+i,j} \leq 1 \]
Shutter inequalities

Consider a sequence of \( k \geq 3 \) requests \( i_1, i_2, \ldots, i_k \) and let \( i_{k+1} = i_1 \) then the following inequality holds for the TSPPDF:

\[
\sum_{h=1}^{k} (x_{i_h,i_{h+1}} + x_{n+i_{h+1},i_h} + x_{n+i_{h+1},n+i_h}) \leq k - 1
\]
Shutter inequalities

The inequality may be lifted into two different ways:
Shutter inequalities

The inequality may be lifted into two different ways: Consider a sequence of \( k > 3 \) requests \( i_1, i_2, \ldots, i_k \) and let \( i_{k+1} = i_1 \) then the following inequalities hold for the TSPPDF:

\[
\sum_{h=1}^{k} \left( x_{i_{h},i_{h+1}} + x_{n+i_{h+1},i_h} + x_{n+i_{h+1},n+i_h} + \sum_{l=h+2}^{h+k-2} x_{i_{h},i_l} \right) \leq k - 1
\]
Shutter inequalities

The inequality may be lifted into two different ways:
Consider a sequence of \( k > 3 \) requests \( i_1, i_2, \ldots, i_k \) and let \( i_{k+1} = i_1 \) then the following inequalities hold for the TSPPDF:

\[
\sum_{h=1}^{k} \left( x_{i_h, i_{h+1}} + x_{n+i_h, i_{h+1}} + x_{n+i_h, n+i_h} + \sum_{l=h+2}^{h+k-2} x_{i_h, i_l} \right) \leq k - 1
\]

\[
\sum_{h=1}^{k} \left( x_{i_h, i_{h+1}} + x_{n+i_h, i_{h+1}} + x_{n+i_h, n+i_h} + \sum_{l=h+2}^{h+k-2} x_{n+i_h, n+i_l} \right) \leq k - 1
\]
Branch-and-cut algorithm

- Cut pool: special cases with at most $O(n^2)$ constraints
  - most of the simple FIFO inequalities
  - some relevant cases of the other inequalities
- Separation procedures:
  - exact for subtour elimination and precedence inequalities
  - exact for FIFO and alternative FIFO inequalities
  - exact for simple FIFO inequalities
  - heuristic for $\pi$ and $\sigma$, $D^+$ and $D^-$, GOC, shutter and lifted shutter inequalities
- Upper bound: heuristic solution by Erdogan et al.
Exact separation procedures

- Subtour elimination constraints
  - solve maxflow from 0 to every other node in $P \cup D$
  - $O(n)$ maxflows

- Precedence constraints
  - solve maxflow from $\{0, n+i\}$ to $\{i, 2n+1\}$ for every $i \in P$
  - $O(n)$ maxflows

- FIFO and alternative FIFO constraints
  - solve two maxflow problems for every node pair $i, j \in P$
  - $O(n^2)$ maxflows
Heuristic separation procedures

- $\pi$ and $\sigma$ inequalities
  - set-oriented tabu search: add or remove a node from the set

- Lifted $D^+$ and $D^-$, GOCs, shutter and lifted shutter inequalities
  - sequence-oriented tabu search: add or remove a node, swap two nodes in the sequence
Branching Strategy

- We branch by extending a feasible path from the origin depot.
- We start by identifying the arc with largest flow among those leaving the depot, and we create the two branches to 1 and 0.
- At each node, we extend the path by branching on the outgoing arc with the largest flow.
Branching Strategy

- We branch by extending a feasible path from the origin depot.
- We start by identifying the arc with largest flow among those leaving the depot, and we create the two branches to 1 and 0.
- At each node, we extend the path by branching on the outgoing arc with the largest flow.
- We may sometimes branch on variables taking value 1.
Branching Strategy

- We branch by extending a feasible path from the origin depot
- We start by identifying the arc with largest flow among those leaving the depot, and we create the two branches to 1 and 0
- At each node, we extend the path by branching on the outgoing arc with the largest flow
- We may sometimes branch on variables taking value 1
- Every time a variable is set to 1 in a branch, we apply filters to eliminate incompatible arcs from the graph
- We have nine different filters, and they are very useful to consistently reduce the number of arcs
Branching Strategy

- We branch by extending a feasible path from the origin depot
- We start by identifying the arc with largest flow among those leaving the depot, and we create the two branches to 1 and 0
- At each node, we extend the path by branching on the outgoing arc with the largest flow
- We may sometimes branch on variables taking value 1
- Every time a variable is set to 1 in a branch, we apply filters to eliminate incompatible arcs from the graph
- We have nine different filters, and they are very useful to consistently reduce the number of arcs
- MUCH BETTER THAN THE OTHER STRATEGIES WE TESTED
Adding Cut to the Model

- We first call the heuristic separation procedures
- We then call the exact separation procedures
- We stop as soon as 10 inequalities have been added
- We do not call the separation procedures when the number of variables fixed to 1 is greater than or equal to \( n/2 \)
Adding Cut to the Model

- We first call the heuristic separation procedures
- We then call the exact separation procedures
- We stop as soon as 10 inequalities have been added
- We do not call the separation procedures when the the number of variables fixed to 1 is greater than or equal to \( n/2 \)
- We use local cuts instead of global cuts
- We use a pool of cuts that contains local cuts previously inserted. Every time we process a node, we first check if the pool contains violated cuts. In that case, we add to the model the most violated cuts (up to 10)
Adding Cut to the Model

- We first call the heuristic separation procedures
- We then call the exact separation procedures
- We stop as soon as 10 inequalities have been added
- We do not call the separation procedures when the the number of variables fixed to 1 is greater than or equal to $n/2$
- We use local cuts instead of global cuts
- We use a pool of cuts that contains local cuts previously inserted. Every time we process a node, we first check if the pool contains violated cuts. In that case, we add to the model the most violated cuts (up to 10)
- MUCH BETTER THAN THE OTHER STRATEGIES WE TESTED
Computational experiments

- TSPLIB instances used by Erdogan et al.
- Tests performed on 49 instances with $n \leq 25$ (52 nodes)
- Maximum CPU time of 3 hours on a 3 GHz Pentium IV
Computational experiments

- TSPLIB instances used by Erdogan et al.
- Tests performed on 49 instances with $n \leq 25$ (52 nodes)
- Maximum CPU time of 3 hours on a 3 GHz Pentium IV
- Root node results
  - Plain formulation has average gap 16.19% (smallest gap 3.65%, largest 32.30%)
  - Full formulation has average gap 10.92% (smallest gap 2.43%, largest 25.89%)
Computational experiments

- TSPLIB instances used by Erdogan et al.
- Tests performed on 49 instances with \( n \leq 25 \) (52 nodes)
- Maximum CPU time of 3 hours on a 3 GHz Pentium IV
- Root node results
  - Plain formulation has average gap 16.19% (smallest gap 3.65%, largest 32.30%)
  - Full formulation has average gap 10.92% (smallest gap 2.43%, largest 25.89%)
- Most useful inequalities:
  1. \( \pi \) and \( \sigma \) inequalities [Balas, Fischetti and Pulleyblank, 1995]
  2. One family of simple FIFO inequalities
  3. Alternative FIFO inequalities
  4. Lifted shutter inequalities
  5. …
## Computational experiments

### Integer results: best case

<table>
<thead>
<tr>
<th>Inst.</th>
<th>n</th>
<th>UB</th>
<th>Carrabs</th>
<th>Erdogan</th>
<th>Branch-and-cut</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>nodes</td>
<td>sec</td>
<td>LB</td>
</tr>
<tr>
<td>a280</td>
<td>9</td>
<td>394</td>
<td>16404</td>
<td>0.2</td>
<td>284.0</td>
</tr>
<tr>
<td>11</td>
<td>469</td>
<td>67000</td>
<td>67000</td>
<td>1.1</td>
<td>5130.0</td>
</tr>
<tr>
<td>13</td>
<td>535</td>
<td>1060629</td>
<td>1060629</td>
<td>25.9</td>
<td>n.d.</td>
</tr>
<tr>
<td>15</td>
<td>604</td>
<td>7153930</td>
<td>7153930</td>
<td>244.0</td>
<td>n.d.</td>
</tr>
<tr>
<td>17</td>
<td>686</td>
<td>18407525</td>
<td>18407525</td>
<td>897.8</td>
<td>n.d.</td>
</tr>
<tr>
<td>19</td>
<td>728</td>
<td>125794822</td>
<td>125794822</td>
<td>7863.3</td>
<td>n.d.</td>
</tr>
<tr>
<td>21</td>
<td>792</td>
<td>n.d.</td>
<td>n.d.</td>
<td>n.d.</td>
<td>n.d.</td>
</tr>
<tr>
<td>23</td>
<td>841</td>
<td>n.d.</td>
<td>n.d.</td>
<td>n.d.</td>
<td>n.d.</td>
</tr>
<tr>
<td>25</td>
<td>888</td>
<td>n.d.</td>
<td>n.d.</td>
<td>n.d.</td>
<td>n.d.</td>
</tr>
</tbody>
</table>
Computational experiments

### Integer results: worst cases

<table>
<thead>
<tr>
<th>Inst.</th>
<th>n</th>
<th>UB</th>
<th>Carrabs</th>
<th>Erdogan</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>nodes</td>
<td>sec</td>
</tr>
<tr>
<td>brd14051</td>
<td>9</td>
<td>4357</td>
<td>941600</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>4400</td>
<td>5618833</td>
<td>52.7</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>4847</td>
<td>268489184</td>
<td>3998.8</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>5236</td>
<td>n.d.</td>
<td>n.d.</td>
</tr>
<tr>
<td>d15112</td>
<td>9</td>
<td>78016</td>
<td>115896</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>87166</td>
<td>1321367</td>
<td>18.9</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>97921</td>
<td>62744515</td>
<td>1259.8</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>113268</td>
<td>n.d.</td>
<td>n.d.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LB</th>
<th>%gap</th>
<th>sec</th>
<th>Branch-and-cut</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>nodes</td>
</tr>
<tr>
<td>4357</td>
<td>0.00</td>
<td>3.3</td>
<td>10</td>
</tr>
<tr>
<td>4400</td>
<td>0.00</td>
<td>5.2</td>
<td>13</td>
</tr>
<tr>
<td>4847</td>
<td>0.00</td>
<td>1754.2</td>
<td>22278</td>
</tr>
<tr>
<td>4951</td>
<td>5.44</td>
<td>10800.0</td>
<td>84153</td>
</tr>
<tr>
<td>78016</td>
<td>0.00</td>
<td>25.4</td>
<td>947</td>
</tr>
<tr>
<td>87166</td>
<td>0.00</td>
<td>112.2</td>
<td>2171</td>
</tr>
<tr>
<td>97921</td>
<td>0.00</td>
<td>3779.7</td>
<td>58948</td>
</tr>
<tr>
<td>106775</td>
<td>5.73</td>
<td>10800.1</td>
<td>101259</td>
</tr>
</tbody>
</table>
Computational experiments

Integer results: summary

- Average gap equal to 0.01%
- Worst gap equal to 10.18%
Computational experiments

Integer results: summary

- Average gap equal to 0.01%
- Worst gap equal to 10.18%
- Number of nodes explored may exceed 200,000
- Number of cuts added may exceed 20,000
Computational experiments

Integer results: summary

- Average gap equal to 0.01%
- Worst gap equal to 10.18%
- Number of nodes explored may exceed 200,000
- Number of cuts added may exceed 20,000
- 42 optimal solutions out of 49
- No known optimal solution missed
- 15 new optimal solutions
Conclusions

- Branch-and-Cut provides the best results (also) for the TSPPDF
Conclusions

- Branch-and-Cut provides the best results (also) for the TSPPDF
- Size of TSPPDF smaller than that of the TSPPDL...
- Size of TSPPDL smaller than that of the TSPPD...
- Size of TSPPD is much smaller than that of the TSP...
- ... there is still a lot that should be done
Conclusions

- Branch-and-Cut provides the best results (also) for the TSPPDF
- Size of TSPPDF smaller than that of the TSPPDL...
- Size of TSPPDL smaller than that of the TSPPD...
- Size of TSPPD is much smaller than that of the TSP...
- ... there is still a lot that should be done
- FIFO (and LIFO) conditions appear in many real-life problems
- They strongly affect the combinatorial structure and are a nice field of research
THANK YOU!