Pharmacy Duty Scheduling Problem  
Exact and Heuristic Approaches

Gökhan Ceyhan\textsuperscript{a}, Fatih Kocatürk\textsuperscript{b} and Özgür Özpeynirci\textsuperscript{b}

\textsuperscript{a} Middle East Technical University  
\textsuperscript{b} İzmir University of Economics

Laboratoire Genie Industriel, Ecole Centrale Paris  
November 4\textsuperscript{th}, 2014

This work is supported by TUBITAK with grant number 111M107
1 Introduction

2 Branch and Price Algorithm

3 VNS Heuristics

4 Computational Results

5 Conclusion
Pharmacies in Izmir

Around 1000 pharmacies and population ~ 3.6M
Pharmacy Duty Scheduling (PDS) Problem

- PDS deals with assignment of duty days to pharmacies during the planning horizon
- PDS is a multi-period facility location problem
  - pharmacies are the facilities
  - opening a facility is assigning a duty to a pharmacy
  - multiple days in the planning horizon
Notation

Sets

- $i$: demand points (customer nodes), $i \in \{1, \ldots, I\}$
- $j$: facility sites (pharmacies), $j \in \{1, \ldots, J\}$
- $t$: time periods (days), $t \in \{1, \ldots, T\}$
- $k$: regions, $k \in \{1, \ldots, K\}$
  - $J_k$: Set of pharmacies in region $k$

Parameters

- $h_i$: demand at customer node $i$
- $d_{ij}$: distance between customer node $i$ and pharmacy $j$
- $n_j$: # of duties for pharmacy $j$
- $r_j$: region of pharmacy $j$
Decision variables

\[ y_{jt} = \begin{cases} 
1 & \text{if pharmacy } j \text{ is on duty on day } t \\
0 & \text{otherwise} 
\end{cases} \]

\[ x_{ijt} = \begin{cases} 
1 & \text{if pharmacy } j \text{ serves customer } i \text{ on day } t \\
0 & \text{otherwise} 
\end{cases} \]
Mathematical Model

Minimize

\[ F = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{t=1}^{T} h_i d_{ij} x_{ijt} \]  

Subject to

\[ \sum_{j=1}^{J} x_{ijt} \geq 1 \quad \forall i, t \]  

\[ \sum_{t=1}^{T} y_{jt} = n_j \quad \forall j \]  

\[ x_{ijt} \leq y_{jt} \quad \forall i, j, t \]  

\[ \sum_{j \in J_k} y_{jt} = 1 \quad \forall t, k \]
Dantzig-Wolfe Decomposition

MASTER PROBLEM:

Sets:

\[ S \]: set of feasible single day schedules
\[ S_j \]: set of feasible single day schedules which include pharmacy \( j \)

Parameters:

\[ a_{ij}^s \]: 1 if pharmacy \( j \) is the closest pharmacy to customer \( i \) in schedule \( s \), 0 otherwise

Decision Variables:

\[ \lambda_s \]: number of times schedule \( s \) is used
Dantzig-Wolfe Decomposition

(MP):

Minimize

\[
F = \sum_{s \in S} \lambda_s \sum_{i \in I} \sum_{j \in J} a_{ij}^s d_{ij} h_i
\]  \hspace{1cm} (6)

Subject to

\[
\sum_{s \in S} \lambda_s \geq T \hspace{1cm} (7)
\]

\[
\sum_{s \in S_j} \lambda_s \leq n_j \hspace{1cm} \forall j \hspace{1cm} (8)
\]

\[
\lambda_s \in \mathbb{Z}^+ \hspace{1cm} \forall s \in S \hspace{1cm} (9)
\]
Pricing Problem

Sets:

\[ J_k \]: set of pharmacies in region \( k \)

Parameters:

\( v \): dual variable associated with (7) in MP
\( u_j \): dual variable associated with (8) in MP

Decision Variables:

\[ y_j = \begin{cases} 
1 & \text{if pharmacy } j \text{ is on duty} \\
0 & \text{otherwise}
\end{cases} \]

\[ x_{ij} = \begin{cases} 
1 & \text{if pharmacy } j \text{ serves customer } i \\
0 & \text{otherwise}
\end{cases} \]
(PP):

Minimize

\[ F = \sum_{j=1}^{J} \left( \sum_{i=1}^{I} x_{ij} d_{ij} h_i \right) - u_j y_j \] - v \quad (10)

Subject to

\[ \sum_{j=1}^{J} x_{ij} \geq 1 \quad \forall i \] \quad (11)

\[ x_{ij} \leq y_j \quad \forall i, j \] \quad (12)

\[ \sum_{j \in J_k} y_j = 1 \quad \forall k \] \quad (13)
\( \lambda^*_s \): the optimal value of decision variable \( \lambda_s \)

\[
\beta_{gh} = \sum_{s \in S_g \cap S_h} \lambda^*_s
\]  

(15)

Pick a pair \((g, h)\) such that \(|\beta_{gh} - \lfloor \beta_{gh} \rfloor - 0.5|\) is minimum. 
Generate the left child

\[
\sum_{s \in S_g \cap S_h} \lambda_s \leq \lfloor \beta_{gh} \rfloor,
\]  

(16)

and the right child.

\[
\sum_{s \in S_g \cap S_h} \lambda_s \geq \lceil \beta_{gh} \rceil
\]  

(17)
Branching

- Select the left child first in a depth-first manner.
- Update the objective function of (PP).
- $\rho_{gh}$: the dual variable of the branching constraint
- $F$: the objective function of the current node

For the left child:

$$F_{g,h} = F - \rho_{gh}(y_g + y_h)$$  \hspace{1cm} (18)

For the right child:

$$F_{g,h} = F + \rho_{gh}(y_g + y_h)$$  \hspace{1cm} (19)
At the end of the root node:
- Let $S^* = \{ \lambda_s | \lambda_s^* > 0 \}$
- Solve (MP) as an IP problem: $\lambda_s \in S^*, \lambda_s \in \mathbb{Z}^+$

At each node of the tree:
- Let $T'$ be the partial schedule formed as follows:
  \[ T' = \sum_{s \in S} \lfloor \lambda_s^* \rfloor \]  
  \[ (20) \]
- If $T' \geq \alpha T$, then run a heuristic for $T - T'$ days, $0 < \alpha < 1$. 

Ceyhan, Kocatürk and Özpeynirci
PDSP: Exact and Heuristic Approaches
Pricing Heuristics

Heuristic 1:

- Consider each region separately,
- Pick the best pharmacy in each region.

\[ D_j = \{ i \in l \mid d_{ij} \leq d_{ij'} \quad \forall j' \in J, r_j \neq r_{j'} \} , \quad \forall j \in J \tag{21} \]

\[ c_j = \left[ \sum_{i \in D_j} d_{ij} - u_j \right] / \left[ \sum_{i \in D_j} h_i \right] \tag{22} \]

\[ j_k^* = \min_{j \in J_k} c_j \tag{23} \]

Select \( j_k^* \) in each region \( k \).
Pricing Heuristics

Heuristic 2:

- Pick the best pharmacy considering the previously selected pharmacies

1. Determine a region order in priori: \( \{ k_1, k_2, ..., k_K \} \)
2. Suppose a partial schedule: \( j_1 \in J_{k_1}, j_2 \in J_{k_2}, ..., j_n \in J_{k_n} \)
3. Select \( j_{n+1} \in J_{k_{n+1}} \) as follows:
4. Let \( J' = \{ j_1, j_2, ..., j_n, j_{n+1} \} \)

\[
j_{n+1} = \arg\min_{j \in J_{k_{n+1}}} \left\{ \sum_{j \in J'} \sum_{i \in I} x_{ij} d_{ij} h_i - u_j \right\}
\]

(24)
**Step 1:** Solve *Heuristic 1*. Add the column and solve (RMP). If (RMP) objective function is not improved for last Q iterations, then go to *Step 2*.

**Step 2:** Solve *Heuristic 2*. Add the column and solve (RMP). If (RMP) objective function is improved, then go to *Step 1*. Else If (RMP) objective function is not improved for last Q iterations, then go to *Step 3*.

**Step 3:** Solve (PP). Add the column and solve (RMP). If (RMP) objective function is improved, then go to *Step 1*. 
PDS problem has similarities with the $p -$ median problem in terms of the objective function and the uncapacitated candidate facilities

Hansen and Mladenović applied VNS to the $p -$ median problem

Hansen et al. (2001) showed that Variable Neighborhood Decomposition Search (VNDS) can be very useful for large scale problems

---

1Kocaturk and Özpeynirci, 2014, Variable neighborhood search for the pharmacy duty scheduling problem, C&OR
Neighborhood Structure

- A feasible solution \( z \) is a \( K \times T \) matrix, \( z \in \mathbb{N}^{K \times T} \)
- The set of feasible solutions is \( Z \), \( Z = \{z^1, z^2, \ldots\} \)
- Swap algorithm to move from one feasible solution to another
- The distance between two feasible solutions \( z^1 \) and \( z^2 \) is the number of swaps applied
- The distance metric is,
  \[
  \rho(z^1, z^2) = z^1 \ominus z^2 = z^2 \ominus z^1 = \sum_{t=1}^{T} \frac{|OD^1_t \setminus OD^2_t|}{2}, \forall z^1, z^2 \in Z
  \]
  where \( OD^s_t \) is the set of on duty pharmacies on day \( t \) in solution \( z^s \), \( s = 1, 2 \)
Algorithm 1 BVNS Algorithm

**Initialization:** Generate an initial solution (schedule) $z$, set the neighborhood structures $N_k$, $k = \{1, \ldots, k_{max}\}$ and define a stopping condition rule.

**Main Step:** Repeat the following steps until the stopping condition is satisfied,

1. $k = 1$,
2. Until $k = k_{max}$ repeat the following steps,
   a. *Shaking:* Generate randomly a solution, $z'$, from the $k^{th}$ neighborhood of $z$, $(z' \in N_k(z))$,
   b. *Local Search:* Find the local minimum, $z''$, around the solution $z'$ using the swap algorithm,
   c. *Move:* If $f(z'') < f(z)$ then change the incumbent solution ($z = z''$) and return $N_1$, $(k = 1)$ and continue to search from there. Otherwise, increase the neighborhood ($k = k + 1$).
Shaking Strategy

- We generate random solutions in the shaking step 2(a),
- In order to obtain \( z' \) from \( z \),
  - Pick \( k \) regions,
  - In each region pick two different pharmacies,
  - If \( n_j > 1 \) for a selected pharmacy \( j \), select one of the days that pharmacy \( j \) is on duty,
  - Apply \( k \) swaps, one in each region to obtain \( z' \).
- We use a controlled random procedure. Give greater importance to
  - the regions with a high number of pharmacies,
  - the pharmacies with higher \( n_j \) values,
  - the days that increase objective function more,
Variable Neighborhood Decomposition Search (VNDS)

- VNDS randomly selects $k$ regions,
- Consider the customers, $I_k$, that can be affected by duty changes in the selected regions,
- VNDS applies a heuristic search in the decomposed solution space,
- Combine the obtained solution for the decomposed space and the rest of the solution,
**Algorithm 3 VNDS Algorithm**

**Initialization:** Generate an initial solution (schedule) \( z \), set the neighborhood structures \( N_k, k = \{2, \ldots, k_{\text{max}}\} \), set customer subsets \( I_k, k \in K \) and define a stopping condition rule.

**Main Step:** Repeat the following steps until the stopping condition is satisfied,

1. \( k = 2 \),
2. Until \( k = k_{\text{max}} \) repeat the following steps,
   a. **Shaking:** Generate randomly a solution, \( z' \), from the \( k^{\text{th}} \) neighborhood of \( z \), \( (z' \in N_k(z)) \), let \( w \) be the solution formed by randomly selected regions \( k \in K^* \) of \( z' \) and customers \( I^* = \bigcup_{k \in K^*} I_k \) such that \( w \in N^{k \times T} \),
   b. **Local Search:** Find the local minimum, \( w' \), in the space of \( w \) using the swap algorithm, and denote the corresponding local minimum with \( z'' \) in the whole space \( Z \), \( (z'' = (z' \setminus w) \cup w') \),
   c. **Move:** If \( f(z'') < f(z) \) then change the incumbent solution \( (z = z'') \) and return \( N_2, (k = 2) \) and continue to search from there. Otherwise, increase the neighborhood \( (k = k + 1) \).
In step 2(a), VNDS has to select $k$ regions,

It first selects an unselected region randomly,

Then, iteratively adds the closest regions until $k$ regions are selected,

Reset "selection status" when all regions are selected,

Note that a selected region can not be the first region in the next iterations but still can be selected,
Variable Neighborhood Restricted Search (VNRS)

Three main differences between VNDS and VNRS algorithms:

- region selection strategy,
  - close regions vs random
- local search,
  - subset of customers vs all customers
- minimum $k$ value,
  - $k \geq 2$ vs $k \geq 1$
Ağlamaz and Özpeynirci (2011) developed a lower bound algorithm for the PDS problem,

There are three groups of test problems; small, large and real life,

- Small Size (9 problem sizes and 10 instances)
- Large Size (3 problem sizes and 5 instances)
- Real Life (2 instances)
Computational Results - BP Algorithm

- Optimal solution known: 86 instances (out of 90)
- Root node LP relaxation
  - $LP_{RMP}^* = F^*$: 80 instances
  - Integer optimal: 35 instances
Computational Results - BP Algorithm

Table 1: The performance of BP in comparison to IBM ILOG CPLEX

<table>
<thead>
<tr>
<th>Instance</th>
<th>BP</th>
<th>IBM ILOG CPLEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>I J T K</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>20 10 5 4</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>20 20 10 4</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>20 30 15 9</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>40 20 5 4</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>40 40 10 9</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>40 60 15 9</td>
<td>9</td>
<td>0.18</td>
</tr>
<tr>
<td>60 30 5 9</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>60 60 10 9</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>60 90 15 9</td>
<td>7</td>
<td>4.45</td>
</tr>
</tbody>
</table>

*The tests are conducted on a machine with Intel Core2Duo 3.00 GHz with 4 GB RAM.
### Table 2: Test Results for Small Scale Problems

<table>
<thead>
<tr>
<th>I</th>
<th>J</th>
<th>T</th>
<th>K</th>
<th>Gap (%)</th>
<th>CPU (Secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Random</strong></td>
<td><strong>BVNS</strong></td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>5</td>
<td>4</td>
<td>3.2</td>
<td>0.0</td>
</tr>
<tr>
<td>40</td>
<td>20</td>
<td>5</td>
<td>4</td>
<td>4.6</td>
<td>0.1</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>10</td>
<td>4</td>
<td>5.5</td>
<td>0.2</td>
</tr>
<tr>
<td>60</td>
<td>30</td>
<td>5</td>
<td>9</td>
<td>3.6</td>
<td>0.7</td>
</tr>
<tr>
<td>20</td>
<td>30</td>
<td>15</td>
<td>9</td>
<td>5.0</td>
<td>5.2</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
<td>10</td>
<td>9</td>
<td>5.8</td>
<td>13.8</td>
</tr>
<tr>
<td>60</td>
<td>60</td>
<td>10</td>
<td>9</td>
<td>7.9</td>
<td>29.9</td>
</tr>
<tr>
<td>40</td>
<td>60</td>
<td>15</td>
<td>9</td>
<td>7.8</td>
<td>5.4</td>
</tr>
<tr>
<td>60</td>
<td>90</td>
<td>15</td>
<td>9</td>
<td>8.9</td>
<td>13.8</td>
</tr>
</tbody>
</table>
### Table 3: Test Results for Large Scale Problems

<table>
<thead>
<tr>
<th>I</th>
<th>J</th>
<th>T</th>
<th>K</th>
<th>Random</th>
<th>Gap (%)</th>
<th>CPU (Secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BVNS</td>
<td>VNDS</td>
<td>VNRS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BVNS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VNDS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VNRS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BVNS</td>
<td>VNDS</td>
<td>VNRS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>350</td>
<td>250</td>
<td>25</td>
<td>49</td>
<td>9.1</td>
<td>1.0</td>
<td>1.6</td>
</tr>
<tr>
<td>350</td>
<td>250</td>
<td>25</td>
<td>49</td>
<td>7.9</td>
<td>1.1</td>
<td>1.3</td>
</tr>
<tr>
<td>350</td>
<td>250</td>
<td>25</td>
<td>49</td>
<td>8.2</td>
<td>1.1</td>
<td>1.6</td>
</tr>
<tr>
<td>350</td>
<td>250</td>
<td>25</td>
<td>49</td>
<td>7.4</td>
<td>0.9</td>
<td>1.2</td>
</tr>
<tr>
<td>350</td>
<td>250</td>
<td>25</td>
<td>49</td>
<td>8.3</td>
<td>1.0</td>
<td>1.3</td>
</tr>
<tr>
<td>350</td>
<td>500</td>
<td>50</td>
<td>49</td>
<td>10.2</td>
<td>1.3</td>
<td>2.1</td>
</tr>
<tr>
<td>350</td>
<td>500</td>
<td>50</td>
<td>49</td>
<td>10.6</td>
<td>1.2</td>
<td>2.0</td>
</tr>
<tr>
<td>350</td>
<td>500</td>
<td>50</td>
<td>49</td>
<td>11.0</td>
<td>1.2</td>
<td>1.9</td>
</tr>
<tr>
<td>350</td>
<td>500</td>
<td>50</td>
<td>49</td>
<td>10.6</td>
<td>1.0</td>
<td>1.6</td>
</tr>
<tr>
<td>350</td>
<td>500</td>
<td>50</td>
<td>49</td>
<td>10.2</td>
<td>1.2</td>
<td>2.1</td>
</tr>
<tr>
<td>350</td>
<td>1000</td>
<td>100</td>
<td>49</td>
<td>11.4</td>
<td>2.6</td>
<td>3.4</td>
</tr>
<tr>
<td>350</td>
<td>1000</td>
<td>100</td>
<td>49</td>
<td>11.6</td>
<td>2.4</td>
<td>3.4</td>
</tr>
<tr>
<td>350</td>
<td>1000</td>
<td>100</td>
<td>49</td>
<td>10.6</td>
<td>2.3</td>
<td>2.8</td>
</tr>
<tr>
<td>350</td>
<td>1000</td>
<td>100</td>
<td>49</td>
<td>12.0</td>
<td>2.5</td>
<td>2.7</td>
</tr>
<tr>
<td>350</td>
<td>1000</td>
<td>100</td>
<td>49</td>
<td>11.5</td>
<td>2.2</td>
<td>2.9</td>
</tr>
</tbody>
</table>
### Table 4: Test Results for Real Life Problems

<table>
<thead>
<tr>
<th>Planning Period</th>
<th>Comparison Type</th>
<th>Random</th>
<th>Current</th>
<th>BVNS</th>
<th>VNRS</th>
<th>VNDS</th>
<th>LB</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010/3</td>
<td>Cost (Million km)</td>
<td>309.27</td>
<td>309.01</td>
<td>294.76</td>
<td>293.46</td>
<td>295.34</td>
<td>291.89</td>
</tr>
<tr>
<td></td>
<td>Gap (%)</td>
<td>5.96</td>
<td>5.87</td>
<td>0.98</td>
<td>0.54</td>
<td>1.18</td>
<td>-</td>
</tr>
<tr>
<td>2011/1</td>
<td>Cost (Million km)</td>
<td>321.41</td>
<td>321.12</td>
<td>306.28</td>
<td>304.82</td>
<td>307.55</td>
<td>303.28</td>
</tr>
<tr>
<td></td>
<td>Gap (%)</td>
<td>5.98</td>
<td>5.88</td>
<td>0.99</td>
<td>0.51</td>
<td>1.41</td>
<td>-</td>
</tr>
</tbody>
</table>
Future work

- Further improvements of BP algorithm
- Re-clustering the pharmacies
- A multiobjective approach: public interest vs pharmacists interests
- Implementation!
Conclusions

- A real life problem
- Problem specific exact algorithm
- VNS heuristics
Izmir University of Economics
- A foundation university
- Izmir Chamber of Commerce
- Established in 2001

Department of Logistics Management
- 8 professors, (Industrial Engineering, Business Administration)
- Undergraduate, Masters, and PhD Programmes
- Industry collaboration
- Open faculty positions (visiting or full time)
Questions and comments

ozgur.ozpeynirci@ieu.edu.tr
dlm.ieu.edu.tr