A tabu search heuristic for the vehicle routing problem with conflicting categories for distribution to nanostores in emerging countries
Objectives

- Last-mile Deliver to Nanostores
  - Propose a mathematical formulation
  - Develop an efficient algorithm
  - Perform computational experiments in modified instances of the literature
Introduction

Reaching 50 million nanostores

Photo by Edgar Blanco
Problem Description

Location
- High pop. density

Economics
- Informal credit

Products
- < 1.000 SKUs

Area
- Reduced

Low Stock + Difficult Access

Many Deliveries + Small Vehicles
Problem Description – Categories

Cap = 1.000 Kg

[Diagram showing a network of points labeled A, B, C, D, and 1, 2, 3, 4, 5, 6, with connections and weights indicated.]
Problem Description – Service Radius

Cap = 1.000 Kg

Shortest Total Distance
Problem Description – Service Radius

Cap = 1.000 Kg  Many retailers adopt a simplified manner of compensations for the logistics providers: The distance to the client that is farthest from the depot (Service Radius)

Shortest Total Service Radius
Problem Description – Max Angle

Cap = 1.000 Kg

Diagram showing points A and B with weights of 500 Kg each and a depot marked with an X.
Problem Description — Max Angle

\[ \Theta_{\text{max}} = 90^\circ \]

\[ \phi = 90^\circ \]

\[ \phi = 120^\circ \]

\[ \phi = 110^\circ \]

\[ \Theta = 120^\circ - 90^\circ = 30^\circ \]
**Mathematical Formulation**

- **Decision Variables:**
  - $x_{ijv}$ binary variable
    - 1 if, and only if, the vehicle $v$ uses the arc $[i, j]$ going from $i$ to $j$
  - $u_{ijv}$ continuous variable
    - dummy demand of the vehicle at vertex $i$ when the vehicle uses the arc $[i, j]$ going from $i$ to $j$.  
  - $f_{cv}$ binary variable
    - 1 if, and only if, vehicle $v$ delivers at least one order of category $c$.  
  - $\phi_{max_v} / \phi_{min_v}$ continuous variable
    - Max/Min azimuth between all the customers that the vehicle serves.
  - $d_v$ continuous variable
    - Service radius of vehicle $v$
Objective Function:

\[ \min \sum_{v} d^v \]

Shortest Total Service Radius
Mathematical Formulation

○ Constrains:

\[
\sum_{i \in N} \sum_{v \in V} x_{ij}^v = 1, \forall j \in N^+ \tag{1}
\]

\[
\sum_{j \in N} x_{ij}^v \leq 1, \forall v \in V, i = 0 \tag{2}
\]

\[
\sum_{i \in N^+} x_{ij}^v = \sum_{h \in N^+} x_{jh}^v, \forall j \in N^+, \forall v \in V \tag{3}
\]

Guarantee that the Demand is delivered
Mathematical Formulation

Constrains:

\[ \sum_{i \in N} \sum_{v \in V} x_{ij}^v = 1, \forall j \in N^+ \]  \hspace{1cm} (1)

\[ \sum_{j \in N} x_{ij}^v \leq 1, \forall v \in V, i = 0 \]  \hspace{1cm} (2)

\[ \sum_{i \in N^+} x_{ij}^v = \sum_{h \in N^+} x_{jh}^v, \forall j \in N^+, \forall v \in V \]  \hspace{1cm} (3)

\[ \sum_{j \in N} \sum_{i \in N} Q_j x_{ij}^v \leq Q_{\text{max}}, \forall v \in V \]  \hspace{1cm} (4)
Mathematical Formulation

- **Constrains:**
  
  1. \[ \sum_{i \in N} \sum_{v \in V} x_{ij}^v = 1, \forall j \in N^+ \]  
  
  2. \[ \sum_{j \in N} x_{ij}^v \leq 1, \forall v \in V, i = 0 \]  
  
  3. \[ \sum_{i \in N^+} x_{ij}^v = \sum_{h \in N^+} x_{jh}^v, \forall j \in N^+, \forall v \in V \]  
  
  4. \[ \sum_{j \in N} \sum_{i \in N} x_{ij}^v \leq Q_{max}, \forall v \in V \]  
  
  5. \[ \sum_{v \in V} \sum_{j \in N^+} u_{ij}^v = |N^+|, i = 0 \]  
  
  6. \[ \sum_{v \in V} \sum_{i \in N} u_{ij}^v - \sum_{v \in V} \sum_{h \in N} u_{jh}^v = 1, \forall j \in N^+ \]  
  
  7. \[ u_{ij}^v \leq |N^+| \times x_{ij}^v, \forall v \in V, \forall j \in N, \forall i \in N \]  

  - Used to give an ordering to all nodes excluding the depot in order to prevent Sub-Tours

---

**Sub-Tour**

![Sub-Tour Diagram](Diagram)
Mathematical Formulation

○ Constrains:

\[
|N^+| f^v_k \geq \sum_{i \in N^+_k} \sum_{j \in N^+_k} x^v_{ij}, \forall v \in V, \forall k \in K
\]

\[
f^v_m - f^v_n \leq 1, \forall v \in V, \forall m \in K, \forall n \in K | C
\]

- \( f^v_k \) 1 when there is at least one order of categorie \( k \) in the route
- It is possible to have just 1 of the incompatible categories

(8)

(9)
Mathematical Formulation

 Constrains: 

\[
\left| N^+ \right| \mathbf{f}_k^v \geq \sum_{i \in N_k^+} \sum_{j \in N_k^+} x_{ij}^v, \forall v \in V, \forall k \in K
\]

\[
f_m^v - f_n^v \leq 1, \forall v \in V, \forall m \in K, \forall n \in K \mid CM_{mn} = 0, m > n
\]

\[
\sum_{i \in N} x_{ij}^v \mathbf{DDC}_j \leq d^v, \forall v \in V, \forall j \in N^+
\]

Makes \( d^v \) the distance of the client farthest from the depot
Mathematical Formulation

- ** Constrains: 

\[ |N^+| \cdot f_k^v \geq \sum_{i \in N^+_k} \sum_{j \in N^+_k} x_{ij}^v, \forall v \in V, \forall k \in K \]  

(8)

\[ f_m^v - f_n^v \leq 1, \forall v \in V, \forall m \in K, \forall n \in K | CM_{mn} = 0, m > n \]  

(9)

\[ \sum_{i \in N} x_{ij}^v \cdot DDC_j \leq d^v, \forall v \in V, \forall j \in N^+ \]  

(10)

\[ \sum_{i \in N} x_{ij}^v \cdot \varphi_j \leq maav, \forall v \in V, \forall j \in N^+ \]  

Makes \( \phi_{\text{max}}^v / \phi_{\text{min}}^v \) the max and min Azimuth  

(11)

\[ \sum_{i \in N} x_{ij}^v \cdot (360 - \varphi_j) \leq mia^v, \forall v \in V, \forall j \in N^+ \]  

Angle route can’t be larger than Max Angle  

(12)

\[ maav - (360 - mia^v) \leq \theta_{\text{max}}, \forall v \in V \]  

(13)
Proposed Method – multi-start heuristic

Algorithm 1: CWGT

Input: Orders and Matrix of compatible categories
Output: Routes

1. \( x \leftarrow x_0; \)
2. \( x_{best} \leftarrow x; \)
3. \( \text{while } it < it_{max} \) \( x \leftarrow x_0; \)
   \( \text{if } g(x) < g(x_{best}) \) \( x_{best} \leftarrow x; \) \( \text{while } TabuTime(x) \)
   \( \text{if } g(x) \)
   \( x_{best} \leftarrow x; \)
4. \( \text{end} \)
5. \( \text{end} \)
6. \( \text{end} \)
7. \( it ++ \)
8. \( \text{end} \)
9. \( \text{Return } \text{BestSol Routes;} \)
Proposed Method

- Clarke e Wright (1964) - Savings

- Modified Saving:

\[ \lambda \]

\[ mult = \frac{Q_j \times Q_j}{Cap} \]

\[ mult = \frac{dc^m \times dc^m}{K} \]
Proposed Method

- Clarke e Wright (1964) – GRASP adaptation

Algorithm 4: GRASP

```
Input: Instance
Output: Solution
1 while it < \( it_{max} \) do
2 | Build Greedy Solution
3 | Improve Current Solution with Local Search
4 | if Sol Quality > Best Sol Quality then
5 | | Best Sol ← Sol
6 | end
7 | it ++
8 end
9 Return Solution;
```

Savings List

1 route each time: **Sequential**
Multiple routes each time: **Parallel**
Proposed Method

- LOCAL SEARCH
- EVALUATION FUNCTION:
  - Delta = After – Before (Service Radius)

**REPLACE (2,3)**

**SWAP (4,5)**
Proposed Method

- TABU SEARCH

Tabu Time – 10 rounds

Local Min

Global Min
Proposed Method

- TABU SEARCH

\[ \Theta_{\text{max}} = 60^\circ \]

Step 1 – SR=22

Step 2 – SR=23

Step 3 – SR= 26

Step 4 – SR=26

Step 5 – SR =18

Step 6 – SR = 18

\[ 59^\circ \quad 48^\circ \]
Preliminary Results

- Sets: A, B and P proposed by Augerat et. al. (1995)
  - 42 Instances
  - Range: 16 to 78 Clients

- Modified version:
  - Each Client can order up to 6 categories (Random Distribution)
  - Size increase 6 times! Range: 96 to 468 orders

- $\Theta_{\text{max}} = 60^\circ$

<table>
<thead>
<tr>
<th>10</th>
<th>Cat</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-&gt;</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
Preliminary Results

- No optimal solutions – 5h

- Instance P—n16 (Reduced to 8 Clients)
  - Optimal Solution = Total Service Radius 168 / 5.100 sec
  - Heuristic Solution = Total Service Radius 168 / < 1 sec
Preliminary Results

- Parameter Tuning – Version and Type (Values = Sum of 42 Instances)

<table>
<thead>
<tr>
<th>Version</th>
<th>Type</th>
<th>Vehicles</th>
<th>T-SR</th>
<th>T-Dist</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>Normal</td>
<td>953</td>
<td>47.573</td>
<td>112.383</td>
<td>25</td>
</tr>
<tr>
<td>Sequential</td>
<td>Weight</td>
<td>960</td>
<td>47.644</td>
<td>113.831</td>
<td>25</td>
</tr>
<tr>
<td>Sequential</td>
<td>Cat</td>
<td>961</td>
<td>47.676</td>
<td>113.617</td>
<td>24</td>
</tr>
<tr>
<td>Parallel</td>
<td>Normal</td>
<td>1.037</td>
<td>52.041</td>
<td>118.092</td>
<td>5</td>
</tr>
<tr>
<td>Parallel</td>
<td>Weight</td>
<td>1.051</td>
<td>52.192</td>
<td>118.406</td>
<td>4</td>
</tr>
<tr>
<td>Parallel</td>
<td>Cat</td>
<td>1.055</td>
<td>52.376</td>
<td>117.458</td>
<td>3</td>
</tr>
</tbody>
</table>

\[ S = (d0_i + d0_j - \lambda \times d_{ij}) \times mult \]

\[ mult = \frac{Q_j \times Q_j'}{Cap} \quad mult = \frac{dc^m \times dc^m'}{K} \]

Weight \quad Cat
Preliminary Results

- Parameter Tuning – $\lambda$ (Values = Sum of 42 Instances)

$$S = (d0i + d0j - \lambda \times dij) \times mult$$
Preliminary Results

- Sequential GRASP Results – 100 re-starts / 10 random seeds (Values = Sum of 42 Instances)

<table>
<thead>
<tr>
<th>Version</th>
<th>Min</th>
<th>Avg</th>
<th>Max</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>45.453</td>
<td>45.964</td>
<td>46.282</td>
<td>1.8%</td>
</tr>
<tr>
<td>Parallel</td>
<td>46.334</td>
<td>46.668</td>
<td>47.005</td>
<td>1.4%</td>
</tr>
<tr>
<td>Delta</td>
<td>1.9%</td>
<td>1.5%</td>
<td>1.5%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version</th>
<th>Min</th>
<th>Avg</th>
<th>Max</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>0.6</td>
<td>48.4</td>
<td>208.8</td>
<td>-118%</td>
</tr>
<tr>
<td>Parallel</td>
<td>0.3</td>
<td>2.3</td>
<td>4.2</td>
<td>-2018%</td>
</tr>
<tr>
<td>Delta</td>
<td>-118%</td>
<td>-2018%</td>
<td>-4900%</td>
<td></td>
</tr>
</tbody>
</table>
## Preliminary Results

### Sequential GRASP Results – Evaluation Function (Service Radius x Distance)

<table>
<thead>
<tr>
<th>Dist</th>
<th>Version</th>
<th>Vehicles</th>
<th>T-SR</th>
<th>T-Dist</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sequential</td>
<td>937</td>
<td>46.542</td>
<td>108.396</td>
<td>2.162</td>
</tr>
<tr>
<td></td>
<td>Parallel</td>
<td>1.001</td>
<td>49.556</td>
<td>110.539</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Sequential</td>
<td>920</td>
<td>45.734</td>
<td>117.384</td>
<td>1.808</td>
</tr>
<tr>
<td></td>
<td>Parallel</td>
<td>947</td>
<td>46.671</td>
<td>117.381</td>
<td>74</td>
</tr>
</tbody>
</table>

### Delta

<table>
<thead>
<tr>
<th>Dist</th>
<th>Version</th>
<th>T-SR</th>
<th>T-Dist</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sequential</td>
<td>-1,8%</td>
<td>-1,7%</td>
<td>8,3%</td>
</tr>
<tr>
<td></td>
<td>Parallel</td>
<td>-5,4%</td>
<td>-5,8%</td>
<td>6,2%</td>
</tr>
</tbody>
</table>

![Diagram of Service Radius and Vehicle Variable Cost](image)
Preliminary Results

- Many softwares only minimize distance.

- It is usual to read that the parallel version yields to better results.

- Opportunity to create methods that take into account specific characteristics.
Next Activities

- Find optimal solutions for small instances;
  - HPC – Cluster Águia University of São Paulo
  - Lower Bounds

- Improvements of the proposed heuristic;

- Analysis of results;

- Completion of the text, general revision of the text and layout.
Thank you!

Master of Engineering in Logistics Systems Candidate

Renan Sallai Iwayama

Advisor: Dr. Prof. Claudio Barbieri da Cunha

2/10/2017