Sparsity-Constrained Transportation Problem

Annie I. Chen (anniecia@mit.edu)
Stephen C. Graves (sgraves@mit.edu)
Massachusetts Institute of Technology
Inventory Positioning for Online Retailers

• **Large scale:**
  – Fulfillment centers (FCs): tens
  – Customer demand zones: hundreds
  – Items (SKUs): millions

• **Integrated and centralized control:**
  – Customers served by FC not limited to geographic location
  – Not all items have to be stocked everywhere
Sparsity Constraints

• Limit the number of FCs that can carry an item
  – There may be other types of sparsity constraints, e.g., limit the number of items in each FC.

• Benefits:
  – Reduce operational complexity (fewer FCs/items to keep track of)
  – Reduce fixed costs (e.g. transportation to FC)
  – Reduce variability (risk pooling of demand zones)

• Trade-off: fulfillment costs
Sparsity-Constrained Transportation Problem

• **Given:**
  – Demand
    • Extension: item affinity
  – FC capacity
    • Throughput, storage
  – Fulfillment cost
    • Distance, weight, time

• **Decision:**
  – Flow amount
Sparsity-Constrained Transportation Problem

**Given:**
- Demand
  - Extension: item affinity
- FC capacity
  - Throughput, storage
- Fulfillment cost
  - Distance, weight, time
- Item sparsity
  - # FCs carrying the item

**Decision:**
- Flow amount
- FC usage

What if we only used 2 FCs?
Sparsity-Constrained Transportation Problem

- **Given:**
  - Demand
    - Extension: item affinity
  - FC capacity
    - Throughput, storage
  - Fulfillment cost
    - Distance, weight, time
  - Item sparsity
    - # FCs carrying the item

- **Decision:**
  - Flow amount
  - FC usage
Sparsity-Constrained Transportation Problem

- **Given:**
  - Demand $d_v^i$
  - FC capacity $b_u$
  - Fulfillment cost $c_{uv}^i$
  - Item sparsity $s_i$

- **Decision:**
  - Flow amount $x_{uv}^i$
  - FC usage $y_u^i \in \{0, 1\}$

---

**Problem Formulation**
1. Sparsify-Improve
2. Column Generation
3. Capacity Allocation
Conclusion
Problem Formulation

1. Sparsify-Improve
2. Column Generation
3. Capacity Allocation

MIP Formulation

(MIP) \[ \min_x \sum_{i \in I} \sum_{(u,v) \in E} c_{uv}^i x_{uv}^i \]

subject to

\[ \sum_{u:(u,v) \in E} x_{uv}^i = d_v^i \quad \forall i \in I, \forall v \in V \quad \text{(demand satisfaction)} \]

\[ \sum_{i} x_{uv}^i \leq b_u \quad \forall u \in U \quad \text{(inbound capacity)} \]

\[ \sum_{v:(u,v) \in E} x_{uv}^i \leq M y_u^i \quad \forall i \in I, \forall u \in U \quad \text{(inbound node usage)} \]

\[ x_{uv}^i \geq 0 \quad \forall i \in I, \forall (u,v) \in E \quad \text{(nonnegativity)} \]

\[ \sum_{u:(u,v) \in E} y_u^i \leq s_i \quad \forall i \in I \quad \text{(inbound flow sparsity)} \]

\[ y_u^i \in \{0, 1\} \quad \forall i \in I, \forall u \in U \quad \text{(inbound node usage)} \]
Solution Approaches

1. Sparsify-Improve
   - Decompose sparsity constraint
   - Conceptually similar to subgradient projection

2. Column Generation
   - Branch-and-price for large scale MIP

3. Capacity Allocation
   - Decompose by item
   - Conceptually similar to primal decomposition
Decompose sparsity constraint

(MIP) \[
\begin{align*}
\min_x & \quad \sum_{i \in I} \sum_{(u,v) \in E} c^i_{uv} x^i_{uv} \\
\text{subject to} & \quad \sum_{v: (u,v) \in E} x^i_{uv} = d^i_v \\
& \quad \sum_{i} \sum_{v: (u,v) \in E} x^i_{uv} \leq b_u \\
& \quad \sum_{v: (u,v) \in E} x^i_{uv} \leq M y^i_u \\
& \quad x^i_{uv} \geq 0 \\
& \quad \sum_{v: (u,v) \in E} y^i_u \leq s_i \\
& \quad y^i_u \in \{0, 1\}
\end{align*}
\]

\[
\begin{align*}
f(y) = \min_x & \quad \sum_{i \in I} \sum_{(u,v) \in E} c^i_{uv} x^i_{uv} \\
\text{subject to} & \quad \sum_{v: (u,v) \in E} x^i_{uv} = d^i_v & \forall i \in I, \forall v \in V \\
& \quad \sum_{i} \sum_{v: (u,v) \in E} x^i_{uv} \leq b_u & \forall u \in U \\
& \quad \sum_{v: (u,v) \in E} x^i_{uv} \leq 0 & \forall i \in I, u : y^i_u = 0 \\
& \quad x^i_{uv} \geq 0 & \forall i \in I, \forall (u,v) \in E
\end{align*}
\]

(Master Problem)

\[
\begin{align*}
\min_{y^i_u \in \{0,1\}} & \quad f(y) \\
\text{subject to} & \quad \sum_{v: (u,v) \in E} y^i_u \leq s_i & \forall i \in I
\end{align*}
\]
The Sparsify-Improve Algorithm

**Initialize:** start with all FCs active for all items

**Step 1: Sparsify**
Generate a sparse solution by progressively eliminating the least active FC-item pair

**Step 2: Improve**
Explore other sparse solutions by heuristically swapping FCs

**Output:** sparse near-optimal solution for Master Problem

---

**Problem Formulation**

1. **Sparsify-Improve**
2. Column Generation
3. Capacity Allocation
4. Conclusion

---

**Master Problem**

\[
    f(y) = \min \sum_{i \in I} \sum_{u : (u, v) \in E} c^i_{uv} x^i_{uv} \\
    \text{subject to} \quad \sum_{u : (u, v) \in E} x^i_{uv} = d^i_v \quad \forall i \in I, \forall v \in V \\
    \sum_{v : (u, v) \in E} x^i_{uv} \leq 0 \quad \forall i \in I, u : y^i_u = 0 \\
    \sum_{u : (u, v) \in E} x^i_{uv} \leq b_u \quad \forall u \in U \\
    x^i_{uv} \geq 0 \quad \forall i \in I, (u, v) \in E
\]

\[
    \min_{y^i_u \in \{0, 1\}} f(y) \\
    \text{subject to} \quad \sum_{u : (u, v) \in E} y^i_u \leq s_i \quad \forall i \in I
\]
Sparsify-Improve in Pictures

1. Sparsify
2. Improve

\[ \sum_{u: (u, v) \in E} y_u^i \leq s_i \quad \forall i \in I \]
Numerical Experiment: Setup

• # of items: 1, 2, 4, 8, 16, 32, 64
• For each item setting, 10 random graphs:
  – 30 fulfillment centers (each with random capacity)
  – 100 demand zones (each with random demand)
  – Sparsity limit = 5 for all items
• Implemented in Python + Gurobi
• Comparison: MIP formulation
• Results: ~5% sub-optimal in 30-40% time
Numerical Experiment: Results

Table 1: Computation time and optimality of Sparsify-Improve and MIP

<table>
<thead>
<tr>
<th></th>
<th>Average computation time</th>
<th>Average optimality (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MIP</td>
<td>Sparsify-Improve</td>
</tr>
<tr>
<td>1</td>
<td>0.917</td>
<td>1.523</td>
</tr>
<tr>
<td>2</td>
<td>5.46</td>
<td>4.013</td>
</tr>
<tr>
<td>4</td>
<td>17.26</td>
<td>11.01</td>
</tr>
<tr>
<td>8</td>
<td>90.13</td>
<td>39.68</td>
</tr>
<tr>
<td>16</td>
<td>376.15</td>
<td>111.97</td>
</tr>
<tr>
<td>32</td>
<td>997.63</td>
<td>338.17</td>
</tr>
<tr>
<td>64</td>
<td>-</td>
<td>1058.281</td>
</tr>
</tbody>
</table>

Results: \(~5\%\) sub-optimal in 30-40% time
Solution Approaches

1. Sparsify-Improve
   - Decompose sparsity constraint
   - Conceptually similar to subgradient projection

2. Column Generation
   - Branch-and-price for large scale MIP

3. Capacity Allocation
   - Decompose by item
   - Conceptually similar to primal decomposition
Column Generation

- Select **best pattern** from a discrete set
- Simplifying assumption: each demand zone served by **only one FC**
- **Pattern**: a vector specifying which FC serves each demand zone
- **Master problem**: choose best pattern for each item
- **Restricted master problem**: choose best pattern from a subset of patterns
- **Pricing subproblem**: generate new patterns (columns) and add to subset
Solution Approaches

1. **Sparsify-Improve**
   - Decompose sparsity constraint
   - Conceptually similar to subgradient projection

2. **Column Generation**
   - Branch-and-price for large scale MIP

3. **Capacity Allocation**
   - Decompose by item
   - Conceptually similar to primal decomposition
Capacity Allocation

- Model a continuous spectrum of capacity allocations (rather than discrete usage patterns)
- View each item (and each FC) as a decision-making agent and decompose by item.

Problem Formulation

1. Sparsify-Improve
2. Column Generation
3. Capacity Allocation

Conclusion
Conclusion

• **Sparsity** is practical in online retail inventory positioning.

• We proposed algorithms that solve the *sparsity-constrained transportation problem*:
  1. **Sparsify-Improve**: decomposes sparsity constraint; gives near-optimal solutions.
  2. **Column Generation**
  3. **Capacity Allocation**

• Related problems:
  – What is the right sparsity level?  
    (Sensitivity and cost-benefit analysis)
  – Scaling up: Decomposition by items? Aggregation?
  – Network design: flexibility and risk

Questions? Comments? [anniecia@mit.edu](mailto:anniecia@mit.edu)
References