Integrated Location Routing and Scheduling Problems

Zeliha Akça  
zelihaakca@lehigh.edu
Rosemary T. Berger  
rosemary.berger@verizon.net
Theodore K. Ralphs  
tkr2@lehigh.edu

Department of Industrial and Systems Engineering  
Lehigh University

CORAL Seminar Series
Outline

1. Introduction
2. Problem Definition and Formulation
3. Solution Methodology
4. Future Work
LRS problem integrates the decisions of determining

- the optimal number and locations of facilities,
- an optimal set of vehicle routes from facilities to customers
- an optimal assignment of routes to vehicles subject to scheduling constraints.

The objective is to minimize the total fixed costs and operating costs of facilities and vehicles.
Motivation

- For multiple customer routes, location and routing are interdependent.
- Assuming one-to-one relationship between vehicles and routes overestimates the required number of vehicles and costs.
  - Fixed costs of vehicles and drivers are high.
  - Companies might have fleets with constant size.
  - Delivery of items might be time sensitive and drivers might have working hour limits.
Integrates other NP-Hard problems such as CFLP, CVRP and MDVSP.
Generalizes problems such as LRP, MDVRP, VRPMT.
Lin et al. [2002] introduce the LRS problem.
They divide the problem into 3 phases: facility location, vehicle routing and loading, and construct a heuristic algorithm which includes some metaheuristics.
Lin and Kwok [2005] extend the study to a multi-objective LRS problem. They use a similar heuristic algorithm.
Problem Definition

Objective

to select a subset of the facilities, construct a set of delivery routes and to assign routes to vehicles with minimum total cost.

Constraints

- Capacitated facilities.
- Capacitated vehicles.
- Time limit for the vehicles.
- Each customer must be visited exactly once.
- Each route and vehicle must start at a facility and return to the same facility.

2 Formulations: Edge-based linear mixed integer model, set-partitioning based linear integer model
Pairing Concept

**Pairing:**
A set of routes that can be served sequentially by one vehicle within the vehicle’s working hour limit.

**A pairing is feasible if**
- total demand of each route $\leq$ vehicle capacity,
- total travel time of the pairing $\leq$ vehicle working hour limit,
- each customer included in the pairing is visited once.

*Figure: Example of pairings*
Set Partitioning-based model: Notation

Sets

\[ N = \text{set of demand nodes} \]
\[ M = \text{set of candidate facility locations} \]
\[ P_j = \text{set of all feasible pairings for facility } j, \forall j \in M \]

Parameters

\[
a_{ipj} = \begin{cases} 
1 & \text{if demand node } i \text{ is in pairing } p \text{ of facility } j, \forall i \in N, j \in M, p \in P_j \\
0 & \text{otherwise}
\end{cases}
\]
\[ C_{jp} = \text{cost of pairing } p \text{ associated with facility } j, \forall j \in M, p \in P_j \]
\[ FC_j = \text{fixed cost of opening facility } j, \forall j \in M \]
\[ CapF_j = \text{capacity of facility } j, \forall j \in M \]

Decision Variables

\[
Z_{jp} = \begin{cases} 
1 & \text{if pairing } p \text{ is selected for facility } j, \forall p \in P_j \text{ and } j \in M \\
0 & \text{otherwise}
\end{cases}
\]
\[ T_j = \begin{cases} 
1 & \text{if facility } j \text{ is selected, } \forall j \in M \\
0 & \text{otherwise}
\end{cases}
\]
Set Partitioning Formulation

(SPP-LRS)

Minimize
\[ \sum_{j \in M} FC_j T_j + \sum_{j \in M} \sum_{p \in P_j} C_{jp} Z_{jp} \]  
subject to
\[ \sum_{j \in M} \sum_{p \in P_j} a_{ipj} Z_{jp} = 1 \quad \forall i \in N \]  
\[ \sum_{p \in P_j} \sum_{i \in N} a_{ipj} d_{ij} Z_{jp} \leq Cap_{Fj} T_j \quad \forall j \in M \]  
\[ Z_{jp} \in \{0, 1\} \quad \forall j \in M, \forall p \in P_j \]  
\[ T_j \in \{0, 1\} \quad \forall j \in M \]
Simple Valid Inequalities

\[ \sum_{j \in M} T_j \geq nREQ \quad (6) \]

\[ \sum_{p \in P_j} a_{ipj}Z_{jp} \leq T_j \quad \forall j \in M, i \in N \quad (\sigma_{ji}) \quad (7) \]

\( nREQ = \) minimum number of required (selected) facilities
Branch and Price Algorithm

**Introduction**

**Problem Definition and Formulation**

**Solution Methodology**

**Future Work**

**References**

**Branch and Price Algorithm**

**ROOT NODE**

- **Initial Columns**
  - Solve Restricted Master Problem (RMP)
  - Dual Variables
    - Until no new column
      - new columns
    - Optimal Soln. of RMP=LOWER BD
    - IP Soln. of RMP=UPPER BD.

- If LP is not integral
  - Branching
    - MODIFIED RPM
      - Constraints + Variable Fixing
        - Solve LP
          - Solve MODIFIED Pricing Problem
    - MODIFIED RPM
      - Constraints + Variable Fixing
        - Solve LP
          - Solve MODIFIED Pricing Problem

**Figure: Branch-and-Price Tree**
Pricing Problem as a Network Problem

Objective: Find a column with minimum reduced cost.

Constraints:
- All of the routes in the pairing start and end at the same facility.
- Total demand of each route \( \leq \) vehicle capacity.
- Total travel time of the pairing \( \leq \) time limit.
- Each customer node can be visited at most once.

Can define the problem as an ESPPRC (not elementary wrt sink).

**Exact solution:** labeling algorithm by Feillet et al. [2004] for ESPPRC.

**Heuristic solution:** labeling algorithm with a LIMIT on the number of labels for each node.

Figure: Constructed Network for facility \( j \)
Branching Rules

1. Branching on the facility location variables, OPEN \ CLOSED. *Simple, no need to update the pricing problems.*

2. Integrality of total # of vehicles at each facility. *Just a fixed cost change in the total reduced cost of a column.*

3. A customer can only be assigned to 1 facility, FORCE \ FORBID a customer for a facility. *Add constraints dual variables of which can be easily incorporated into the pricing problem.*

4. Modified *Ryan and Foster* branching rule suggested by Desrochers and Soumis [1989]. *Update the arc costs in pricing problem.*
Branch and Price Algorithm

**STEP 1: Heuristic Branch and Price Tree**

- Initial Column Generator
- Heuristic Upper Bound
- Branching Rules
- BP Tree
- Pricing Problem with LABEL LIMIT

**STEP 2: Exact Branch and Price Tree**

- Columns + Upper Bound from STEP 1
- Branching Rules
- BP Tree
- Pricing Problem with LABEL LIMIT
- Limit increased gradually + Exact Pricing Problem

Figure: Branch and Price Algorithm
Implementation and Test Problems

- MINTO 3.1 and CPLEX 9.1.
- Initial columns and upper bound for heuristic BP are generated using a facility location heuristic, some VRP heuristics and a bin packing heuristic.
- Step 1 of the algorithm (Heuristic BP) is run for 3 CPU hours.
- Step 2 of the algorithm (Exact BP) is run for 6 CPU hours.
- 6 set of customer and candidate facility locations, customer demands generated from MDVRP benchmark problems developed by Cordeau et al. [1995].
- For each set of locations, 2 possible vehicle capacity, 2 possible time limit values are used. Therefore, 24 instances with 30 customers.
- Facility and vehicle fixed costs, vehicle operating cost are estimated.
## Computational Results: 2-step Branch and Price Algorithm

<table>
<thead>
<tr>
<th>Data</th>
<th>LP</th>
<th>IP</th>
<th>% Int. Gap</th>
<th>CPU Time</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Step 1</td>
<td>Step 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p01-f-11</td>
<td>4821.07</td>
<td>4877.33</td>
<td>0.00%</td>
<td>1.36 hr</td>
<td>5.49 hr</td>
</tr>
<tr>
<td>p01-f-12</td>
<td>4715.3</td>
<td>4987.17</td>
<td>2.83%</td>
<td>3 hr</td>
<td>6 hr</td>
</tr>
<tr>
<td>p01-f-21</td>
<td>4582.27</td>
<td>4684.67</td>
<td>0.01%</td>
<td>3 hr</td>
<td>6 hr</td>
</tr>
<tr>
<td>p01-f-22</td>
<td>4482.2</td>
<td>4885</td>
<td>8.25%</td>
<td>3 hr</td>
<td>6 hr</td>
</tr>
<tr>
<td>p03-f-11</td>
<td>6467.35</td>
<td>6504.83</td>
<td>0.00%</td>
<td>1.76 hr</td>
<td>0.89 hr</td>
</tr>
<tr>
<td>p03-f-12</td>
<td>6370.12</td>
<td>6497.33</td>
<td>0.94%</td>
<td>3 hr</td>
<td>6 hr</td>
</tr>
<tr>
<td>p03-f-21</td>
<td>6263.17</td>
<td>6420.67</td>
<td>0.55%</td>
<td>2.5 hr</td>
<td>6 hr</td>
</tr>
<tr>
<td>p03-f-22</td>
<td>6193.52</td>
<td>6196.5</td>
<td>0.05%</td>
<td>0.82 hr</td>
<td>6 hr</td>
</tr>
<tr>
<td>p03-l-11</td>
<td>4896.04</td>
<td>5014.67</td>
<td>0.00%</td>
<td>1.58 hr</td>
<td>0.54 hr</td>
</tr>
<tr>
<td>p03-l-12</td>
<td>4756.95</td>
<td>5026.83</td>
<td>4.12%</td>
<td>3 hr</td>
<td>6 hr</td>
</tr>
<tr>
<td>p03-l-21</td>
<td>4679.23</td>
<td>4730.17</td>
<td>0.00%</td>
<td>1.43 hr</td>
<td>0.67 hr</td>
</tr>
<tr>
<td>p03-l-22</td>
<td>4562.29</td>
<td>4706.33</td>
<td>1.28%</td>
<td>2.4 hr</td>
<td>6 hr</td>
</tr>
<tr>
<td>p07-f-11</td>
<td>4765.63</td>
<td>4808.33</td>
<td>0.00%</td>
<td>1.82 hr</td>
<td>1.06 hr</td>
</tr>
<tr>
<td>p07-f-12</td>
<td>4654.19</td>
<td>4760.17</td>
<td>0.01%</td>
<td>2.65 hr</td>
<td>6 hr</td>
</tr>
<tr>
<td>p07-f-21</td>
<td>4558.21</td>
<td>4683.5</td>
<td>0.00%</td>
<td>1.74 hr</td>
<td>2.56 hr</td>
</tr>
<tr>
<td>p07-f-22</td>
<td>4453.2</td>
<td>4745</td>
<td>5.77%</td>
<td>3 hr</td>
<td>6 hr</td>
</tr>
<tr>
<td>p11-f-11</td>
<td>7915.87</td>
<td>8054.67</td>
<td>0.00%</td>
<td>1 min</td>
<td>1.19 min</td>
</tr>
<tr>
<td>p11-f-12</td>
<td>7791.1</td>
<td>8036.33</td>
<td>0.00%</td>
<td>2.32 min</td>
<td>4.17 min</td>
</tr>
<tr>
<td>p11-f-21</td>
<td>7531.81</td>
<td>7634.67</td>
<td>0.00%</td>
<td>0.4 min</td>
<td>0.21 min</td>
</tr>
<tr>
<td>p11-f-22</td>
<td>7469.84</td>
<td>7634.67</td>
<td>0.00%</td>
<td>0.85 min</td>
<td>0.47 min</td>
</tr>
<tr>
<td>p11-l-11</td>
<td>9929.75</td>
<td>10043</td>
<td>0.00%</td>
<td>0.25 min</td>
<td>0.18 min</td>
</tr>
<tr>
<td>p11-l-12</td>
<td>9837.09</td>
<td>9937.33</td>
<td>0.00%</td>
<td>0.42 min</td>
<td>0.31 min</td>
</tr>
<tr>
<td>p11-l-21</td>
<td>9645.66</td>
<td>9758.17</td>
<td>0.00%</td>
<td>0.8 min</td>
<td>0.74 min</td>
</tr>
<tr>
<td>p11-l-22</td>
<td>9416.78</td>
<td>9500.67</td>
<td>0.00%</td>
<td>1.08 min</td>
<td>0.75 min</td>
</tr>
</tbody>
</table>
Future Work

- Improve the algorithm to be able to solve larger instances.
- Investigate how to incorporate cuts into the algorithm, develop a branch, cut and price algorithm.
- Update the algorithm to solve the special cases of the LRS problem such as LRP, MDVRP, VRPMT.


