Optimizing Location, Routing and Scheduling Decisions Under Capacity Restrictions

Zeliha Akça[1]
Ted K. Ralphs[1] & Rosemary T. Berger

International Workshop on Column Generation,
Aussois, France, June 2008

[1] Industrial and Systems Engineering Dept.,
COR@L Lab, Lehigh University

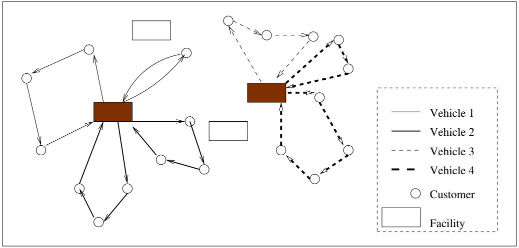
Outline

Location Routing & Scheduling (LRS) Problem

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.
- One-to-one relationship between vehicles and routes overestimates the number of vehicles used and costs.
  - Large fixed costs for vehicles and drivers.
  - Constant fleet size.
  - Working hour limit for drivers.
  - Time sensitive items.

LRSP in Literature

Lin et al. [2002]:

- Introduce the problem for a phone company.
- Divide the problem into 3 phases: facility location, vehicle routing and loading.
- Construct a heuristic algorithm which includes some metaheuristics.
- Solve a real instance with 27 customers and 4 candidate facilities.

Lin and Kwok [2005]:

- Extend the study to a multi-objective LRS problem.
- They develop a similar heuristic algorithm.
- Real instances with 27 customers and 85 customers,
- Random instances up to 200 customers.

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.

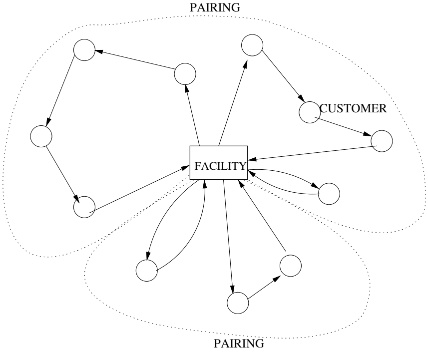
Constraints

- 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.

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

- each customer included in the pairing is visited once,
- each route included starts and ends at the same facility,
- total demand of each route $\leq$ vehicle capacity,
- total travel time of the pairing $\leq$ vehicle working hour limit

Set Partitioning-based model: Notation

Sets

$$I = \text{set of demand nodes}$$
$$J = \text{set of candidate facility locations}$$
$$P_j = \text{set of all feasible pairings for facility } j, \forall j \in J$$

Parameters

$$a_{ip} = \begin{cases} 1 & \text{if demand node } i \text{ is in pairing } p \text{ of facility } j, \forall i \in I, j \in J, p \in P_j \\ 0 & \text{otherwise} \end{cases}$$
$$C_p = \text{cost of pairing } p \text{ associated with facility } j, \forall p \in P_j, j \in J$$
$$F_j = \text{fixed cost of opening facility } j, \forall j \in J$$
$$C_j^F = \text{capacity of facility } j, \forall j \in J$$

Decision Variables

$$z_p = \begin{cases} 1 & \text{if pairing p is selected for facility j, } \forall p \in P_j \text{ and } j \in J \\ 0 & \text{otherwise} \end{cases}$$
$$t_j = \begin{cases} 1 & \text{if facility j is selected, } \forall j \in J \\ 0 & \text{otherwise} \end{cases}$$

Set Partitioning Formulation

(SPP-LRS)

$$\text{Minimize} \quad \sum_{j\in J} F_j t_j + \sum_{j\in J}\sum_{p\in P_j} C_p z_p \tag{1}$$

$$\text{subject to} \quad \sum_{j\in J}\sum_{p\in P_j} a_{ip} z_p = 1 \quad \forall i \in I \quad (\pi_{\mathbf{i}}), \tag{2}$$

$$\sum_{p\in P_j}\sum_{i\in I} a_{ip} D_i z_p \leq C_j^F t_j \quad \forall j \in J \quad (\mu_{\mathbf{j}}), \tag{3}$$

$$z_p \in \{0,1\} \quad \forall p \in P_j, \forall j \in J, \tag{4}$$

$$t_j \in \{0,1\} \quad \forall j \in J. \tag{5}$$

Simple Valid Inequalities

$$\sum_{p \in P_j} a_{ip} z_p \leq t_j \qquad \forall i \in I, \forall j \in J \quad (\sigma_{\mathbf{ji}}), \tag{6}$$

$$\sum_{j \in J} t_j \geq N^F, \tag{7}$$

$$\sum_{p \in P_j} z_p = v_j \qquad \forall j \in J \qquad (\nu_{\mathbf{j}}), \tag{8}$$

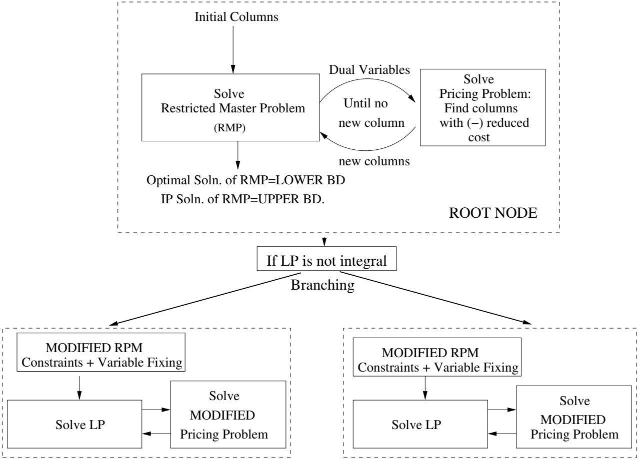
$$v_j \geq t_j \qquad \forall j \in J, \tag{9}$$

$$v_j \in \mathbb{Z}^+ \qquad \forall j \in J. \tag{10}$$

N^F is the minimum number of facilities required to be open:

$$N^F = \text{argmin}_{\{l=1..|J|\}} \left(\sum_{t=1}^{l} C^F_{j_t} \geq \sum_{i \in I} D_i \right) \text{ s.t. } C^F_{j_1} \geq C^F_{j_2} \geq \ldots \geq C^F_{j_n}$$

Branch and Price Algorithm



Describing Pricing Problem

Objective

To find a pairing (**a set of routes**) associated with a variable with minimum reduced cost:

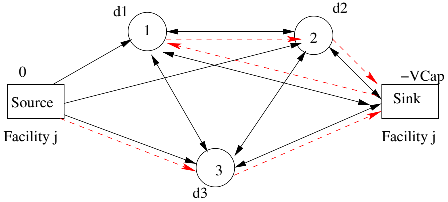
$$\hat{C}_p = C_p - \sum_{i \in N} a_{ip} \cdot \pi_i + \sum_{i \in N} a_{ip} \cdot d_i \cdot \mu_j + \sum_{i \in N} a_{ip} \cdot \sigma_{ji} - \nu_j \quad \forall p \in P_j, j \in J \tag{11}$$

Subject to

- All of the routes must start & end at the same facility.
- Each customer node can be visited at most once.
- Total demand of each route $\leq$ vehicle capacity.
- Total travel time of the pairing $\leq$ time limit.

Pricing Problem as a Network Problem: Cont.

- For each facility, construct a network with source and sink.



- Cost of arc (i,k) in network for facility j:

$$\hat{c}_{ik} = \begin{cases} c_{ik} - \pi_k + d_k \mu_j + \sigma_{jk} & \text{if } k \text{ is a customer node} \\ c_{ik} & \text{otherwise} \end{cases}$$

- Elementary (wrt. customer nodes), resource constraint shortest path problem (ESPPRC).
- Not elementary wrt sink $\rightarrow$ multiple vehicle routes.
- Sink is the end of a route and possibly beginning of another one.

Solving Pricing Problem

- Label setting algorithm by Feillet et al. [2004] to solve ESPPRC.
- All possible paths from source to sink are assigned a label.
- To reduce the number of labels, only the labels that are not dominated by any other label are considered.

2-phase ESPPRC Algorithm to Find Pairings

Phase 1: Network includes a source, a sink and customer nodes.

- Run ESPPRC algorithm with vehicle capacity and time limit resources.
- Run domination algorithm for the labels of sink.

Phase 2: Network includes a source, a sink and sink labels from phase 1.

- Each node starts with the label from Phase 1.
- Run ESPPRC algorithm with time limit resource.

Branching Rules

Rule 1 Facility location variables → **OPEN** \ **CLOSED**.
Simple, no need to update the pricing problems.

Rule 2 Total # of vehicles at each facility → **INTEGER**
Just a fixed cost change in the total reduced cost of a column.

Rule 3 A customer can only be assigned to 1 facility
→ **FORCE** customer i must be served by facility j:

$$\sum_{p \in P_j} a_{ip} Z_p \geq 1 \quad (\gamma_{ji}) \tag{12}$$

→ **FORBID** customer i cannot be served by facility j:

$$\sum_{p \in P_j} a_{ip} Z_p \leq 0 \tag{13}$$

Easy to incorporate into pricing problem.

Rule 4 Branching on flow on single arcs.

Heuristic Solutions For Pricing Problem

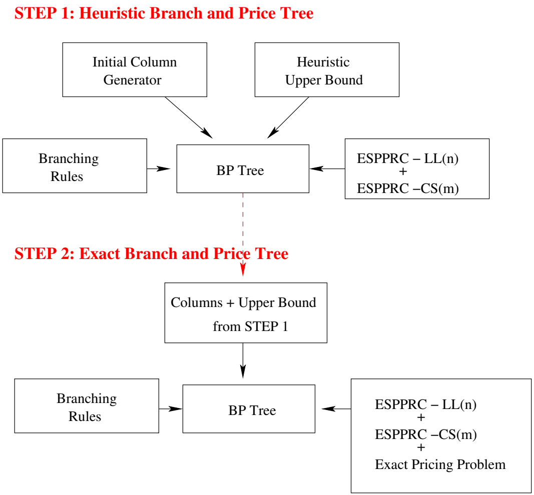
ESPPRC with Label Limit (ESPPRC - LL(n))

- Set a label limit, n.
- Sort the labels of node on reduced cost.
- Keep at most n of the non processed labels with smallest reduced cost.
- For small values of n, it is very quick.
- Label limit can be gradually increased.

ESPPRC for a Subset of Customers (ESPPRC - CS(n))

- Choose n based on the average demand and vehicle capacity.
- Choose a subset of customers C_S with size n based on reduced costs of arcs in the network.
- Apply ESPPRC algorithm to customer set C_S.

2-Step Branch & Price Algorithm



Implementation and Test Problems

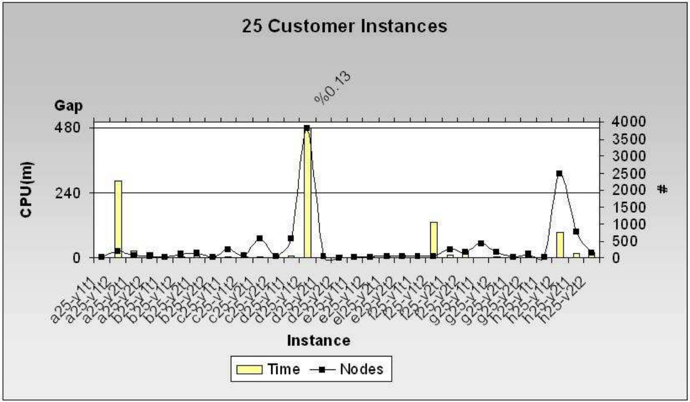
- MINTO 3.1 and CPLEX 9.1.
- Customer and candidate facility locations, customer demands are generated using MDVRP benchmark problems developed by Cordeau et al. [1995].
- Instances with 25 and 40 customers and 5 facilities.
- For each set of locations, 2 possible vehicle capacity, 2 possible time limit values are used.
- For 25 customer instances, 1 step branch & price (8 CPU hours).
- For 40 customer instances, 2 step branch & price: Step 1 (Heuristic BP) is run for 2 CPU hours.
- Step 2 of the algorithm (Exact BP) is run for 6 CPU hours.

Effect of ESPPRC - CS(n)

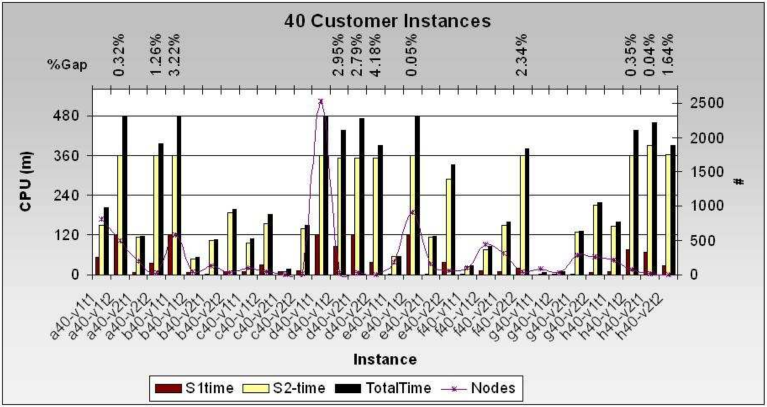
Data	**With out SS**		**With SS**		Data	**With out SS**		**With SS**	
	UB-S1	**Gap-S2**	**UB-S1**	**Gap-S2**		**UB-S1**	**Gap-S2**	**UB-S1**	**Gap-S2**
a40-v1t1	7131	0.01%	7131	0	e40-v1t1	7167	0	7167	0
a40-v1t2	**6986**	0.14%	7170	0.32%	e40-v1t2	6950	0.07%	6950	**0.05%**
a40-v2t1	7017	0	**6866**	0	e40-v2t1	6857	0	**6849**	0
a40-v2t2	6821	1.99%	6821	**1.26%**	e40-v2t2	6846	3.19%	6846	**0**
b40-v1t1	7252	4.14%	**7183**	**3.22%**	f40-v1t1	7167	0	7167	0
b40-v1t2	6934	0	6934	0	f40-v1t2	7001	0	7001	0
b40-v2t1	6823	0	6823	0	f40-v2t1	6925	0.01%	**6881**	**0**
b40-v2t2	6634	0	**6633**	0	f40-v2t2	6851	2.55%	**6845**	**2.34%**
c40-v1t1	8780	0	8780	0	g40-v1t1	7343	0	7343	0
c40-v1t2	8756	0	**8753**	0	g40-v1t2	**7117**	0	7124	0
c40-v2t1	8663	0	8663	0	g40-v2t1	**7000**	0	7005	0
c40-v2t2	8456	0	**8438**	0	g40-v2t2	6944	0	6944	0
d40-v1t1	7793	0.23%	**7740**	**0**	h40-v1t1	**7012**	0.01%	7013	**0**
d40-v1t2	7544	6.32%	**7236**	**2.95%**	h40-v1t2	6972	1.22%	6972	**0.35%**
d40-v2t1	7140	2.29%	**7132**	**2.79%**	h40-v2t1	6869	3.01%	**6857**	**0.04%**
d40-v2t2	7430	8.52%	**6947**	**4.18%**	h40-v2t2	6667	1.78%	**6658**	**1.64%**

Step 1 time $\rightarrow$ 3.4, step 2 time $\rightarrow$ 1.57, and total time $\rightarrow$ 1.61.

1 Step Branch & Price Algorithm for 25 Customer Instances



2 Step Branch & Price Algorithm for 40 Customer Instances



Comparison with Literature: Lin et. al (2002) Instances

Instances	Branch & Bound		Best Heuristic	
	Obj.	CPU (s)	Obj.	CPU (s)
1: 4 depots, 10 custs.	309,817	1155	309,817	0.44
2: 4 depots, 10 custs.	309,808	982	309,808	0.49
3: 4 depots, 12 custs.	312,036*	$>$ 10,000	312,036	0.82
4: 4 depots, 27 custs.	-	-	625,752.5	6

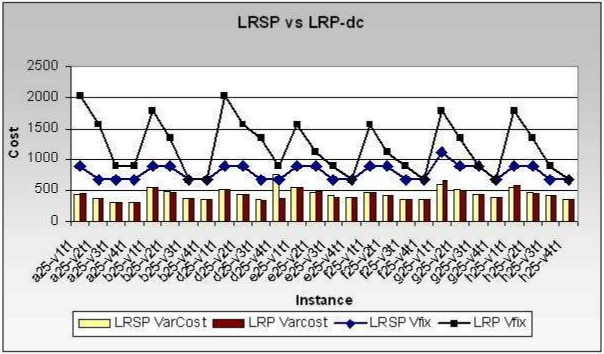
* Best solution in 10000 CPU seconds in a Pentium III machine.

a. Lin et. al (2002)

Instances	Obj.	CPU (s)
1: 4 depots, 10 custs.	309,817	0.70
2: 4 depots, 10 custs.	309,808	0.26
3: 4 depots, 12 custs.	312,036	0.13
4: 4 depots, 27 custs.	625,750.167	135.20

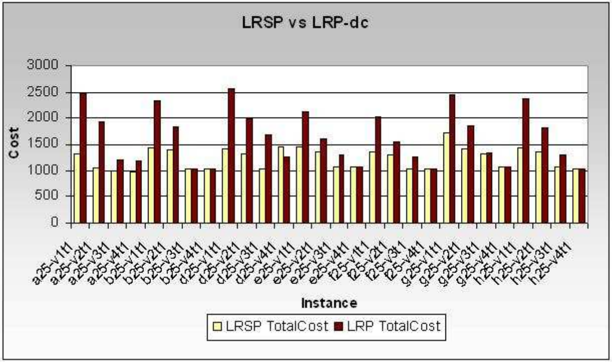
b. 1 Step Branch and Price Algorithm

Effect of Scheduling Constraint: LRSP vs LRP-DC



One of the open facilities is different in 3 of the instances.

Effect of Scheduling Constraint: LRSP vs LRP-DC



Conclusions

- Formulated and designed a branch and price algorithm to solve the LRSP to optimally.
- Enhanced the algorithm by a heuristic pricing problems and by a heuristic step.
- Solved instances up to 40 customers and 5 facilities.
- Solved LRP-DC and ompare the costs with LRSP.

J.F. Cordeau, M. Gendreau, and G. Laporte. A Tabu Search Heuristic for Periodic and Multi-Depot Vehicle Routing Problems, 1995. Technical Report 95-75, Center for Research on Transportation, Montréal. Available at neo.lcc.uma.es/radi-aeb/WebVRP/Problem_Instances/CordeauFilesDesc.html.

D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen. An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints: Application to Some Vehicle Routing Problems. *Networks*, 44(3):216–229, 2004.

C.K. Lin, C.K. Chew, and A. Chen. A Location-Routing-Loading Problem for Bill Delivery Services. *Computers and Industrial Engineering*, 43:5–25, 2002.

C.K. Lin and R.C.W. Kwok. Multi-Objective Metaheuristics for a Location-Routing Problem withMultiple Use of Vehicles on Real Data and Simulated Date. *European Journal of Operational Research*, In Press, 2005.