An Iterated Greedy Local Search Algorithm with Variable Neighborhood Descent for the Capacitated Vertex p-Center Problem[1]

Roger Z. Ríos

Graduate Program in Systems Engineering
Universidad Autónoma de Nuevo León
San Nicolás de los Garza, NL

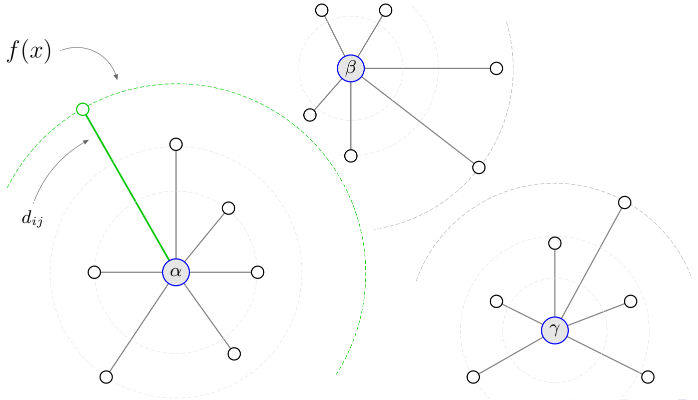
IV Congreso de la Sociedad Mexicana de Investigación de Operaciones
Cd. Juárez, 08 Octubre 2015

[1] Joint work with Dagoberto R. Quevedo Orozco (SINTEC)

Outline

1. Problem Statement and Motivation
2. Proposed Heuristic
3. Empirical Work
4. Wrap-up

Locating p facilities and assigning customers to them so as to minimize the longest distance between any customer and its assigned facility (bottleneck). It is required that the total customer demand assigned to each facility does not exceeded its given capacity. The CpCP is $\mathcal{NP}$-hard.



- Sets and parameters

$$
\begin{aligned}
V &= \{1, 2, ..., n\}, \text{ Set of nodes} \\
K &= \{1, 2, ..., p\}, \text{ Set of facility indices} \\
d_{ij} &= \text{Distance between nodes } i \text{ and } j,\ i, j \in V \\
w_j &= \text{Demand of customer } j \in V \\
s_i &= \text{Capacity of location } i \in V
\end{aligned}
$$

- Variables sets

$$
\begin{aligned}
P &= \{c(1), ..., c(p)\}, \text{ Set of centers} \\
X &= \{X_1, ..., X_p\}, \ p\text{-partition of } V
\end{aligned}
$$

Objective

$$\min_{X \in \Pi} \max_{k \in K} f(X_k) \tag{1}$$

where Π is the collection of all p-partitions of V. For a given territory X_k its cost function is $f(X_k) = \max_{j \in X_k} \{d_{j,c(k)}\}$ where the center $c(k)$, is given by,

$$c(k) = \arg\min_{i \in X_k} \left\{ \max_{j \in X_k} \left\{ d_{ij} : \sum_{j' \in X_k} w_{j'} \leq s_i \right\} \right\} \tag{2}$$

By convention, if for a given X_k there is not any $i \in X_k$ such that $\sum_{j \in X_k} w_j \leq s_i$ then $f(X_k) = \infty$.

Applications

(a) Public school district planning

(b) Medical facility system design

(c) Emergency facility location

And any other system that naturally impose the presence of limits to the total demand that each facility can supply and its cost is linked to the "worst" possible service time/length/etc.

State of the Art

- M. Scaparra, S. Pallottino, and M. Scutellà. Large-scale local search heuristics for the capacitated vertex p-center Problem. *Networks*, 43(4): 241–255, 2004.
- F. A. Özsoy, and M. Ç. Pınar. An exact algorithm for the capacitated vertex p-center problem. *Computers & Operations Research*, 33(5):1420–1436, 2006.
- M. Albareda-Sambola, J. A. Díaz, and E. Fernández. Lagrangean duals and exact solution to the capacitated p-center problem, *European Journal of Operational Research* 201(1): 71–81, 2010.

State of the Art

- M. Scaparra, S. Pallottino, and M. Scutellà. Large-scale local search heuristics for the capacitated vertex p-center Problem. *Networks*, 43(4): 241–255, 2004.
- F. A. Özsoy, and M. Ç. Pınar. An exact algorithm for the capacitated vertex p-center problem. *Computers & Operations Research*, 33(5):1420–1436, 2006.
- M. Albareda-Sambola, J. A. Díaz, and E. Fernández. Lagrangean duals and exact solution to the capacitated p-center problem, *European Journal of Operational Research* 201(1): 71–81, 2010.

- The CpCP has received less attention in the literature. To our knowledge, there is only one heuristic proposed by Scaparra et al. (2004).

Proposed Heuristic

Algorithm 1 GVND

procedure GVND$(V, p, \alpha, Iter_{\max})$
  $X \leftarrow$ CONSTRUCTION(p)
  $X \leftarrow \mathrm{VND}(X)$
  $X^{\text{best}} \leftarrow X$
  while $\neg$(stopping criteria) **do**
    $X \leftarrow \mathrm{IGLS}(\alpha, X)$
    $X \leftarrow \mathrm{VND}(X)$
    if X is better that X^{best} **then**
      $X^{\text{best}} \leftarrow X$
    else
      $X \leftarrow$ SHAKE(X)
    end if
    $Iter_{\max} \leftarrow Iter_{\max} - 1$
  end while
  return X^{best}
end procedure

A metaheuristic framework with a greedy randomized adaptive procedure with probability selection in its construction phase. The improvement phase applies a Iterated Greedy Local Search (IGLS) followed by Variable Neighborhood Descent (VND) with two neighborhoods based on insertion and exchange.

Choose the first center randomly ($P \leftarrow r$). Then, iteratively choose the next center seeking a node whose weighted distance from its nearest center is relatively large. For each $j \in V \setminus P$, its nearest center is given by $i^* = \arg\min_{i \in P}\{d_{ij}\}$. The greedy function is:

Greedy Function Phase 1

$$\gamma(j) = s_j d_{i^* j} \tag{3}$$

For each node $j \in V \setminus P$, the probability of choosing j, can be computed as:

$$\pi(j) = \frac{\gamma(j)}{\sum_{j' \in V \setminus P} \gamma(j')} \tag{4}$$

Stop when $|P| = p$.

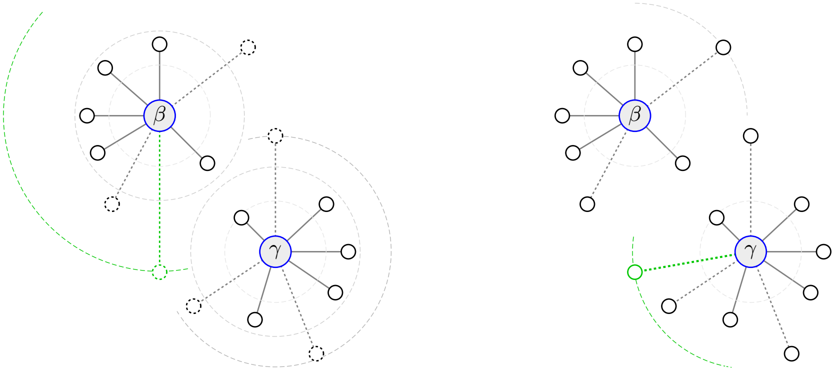
Consists of allocating the customers to these centers. The customers are defined by the remaining nodes $j \in V \setminus P$. We define a greedy function that measures the cost of assigning a customer j to a center k as follows:

Greedy Function Phase 2

$$\phi(j,k) = \max \left\{ \frac{d_{jc(k)}}{\bar{d}}, -\left(s_{c(k)} - \sum_{j' \in X_k} w_{j'} \right) + w_j \right\} \quad (5)$$

where $\bar{d} = \max_{i,j \in V}\{d_{ij}\} + 1$ is a normalization factor. Then each node j is assigned to its nearest center, namely $X_{k^*} \leftarrow X_{k^*} \cup \{j\}$ where $k^* = \arg\min_{k \in K} \phi(j,k)$.

Deallocating the $\alpha\%$ of nodes in X_k, with high values of $\rho(j) = d_{jc(k)} / \sum_{j' \in X_k} d_{jc(k)}$. Then each disconnected node is reassigned to its nearest center. A priority assignment is given to the bottleneck nodes.



(a) Destruction (b) Reconstruction

Variable Neighborhood Descent

Local Search

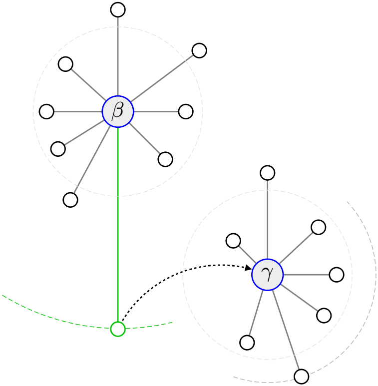
This method is formed by two neighborhoods based on reinsertion and exchange movements. The neighborhoods are denoted as $\mathcal{N}_k, k = 1, ..., k_{\max}$, in this case $k_{\max} = 2$.

Algorithm 2 Variable Neighborhood Descent

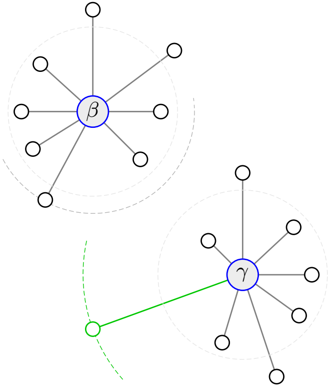
procedure $\mathrm{VND}(X)$
    while $k \leq k_{\max}$ **do**
        $X' \leftarrow \arg\min_{y \in \mathcal{N}_k(X)} f(y)$
        if X' is better that X **then**
            $X' \leftarrow X$
            $k \leftarrow 1$
        else
            $k \leftarrow k + 1$
        end if
    end while
    return X
end procedure

$\mathcal{N}_1$: Neighborhood by Reinsertion

Considers moves where a node i (currently assigned to center of set X_q) is assigned to set X_k, i.e., given X $move(i,k) = \{X_1, \ldots, X_q \setminus \{i\}, \ldots, X_k \cup \{i\}, ..., X_p\}$ where i must be a bottleneck node for the move to be attractive.



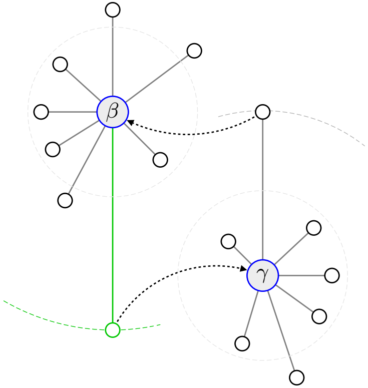
(a) Selection



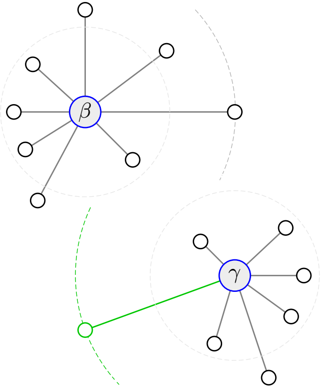
(b) Application

$\mathcal{N}_2$: Neighborhood by Exchange

Considers moves where two nodes i and j in different subsets are swapped, i.e., given X, $move(i,j) = \{X_1, ..., X_q \cup \{j\} \setminus \{i\}, ..., X_k \cup \{i\} \setminus \{j\}, ..., X_p\}$, where either i or j must be a bottleneck node for the move to be attractive.



(a) Selection



(b) Application

Improvement Criteria

We use an effective improvement criteria proposed in Scaparra et al. (2004) which includes the reduction of bottleneck elements, this is defined as

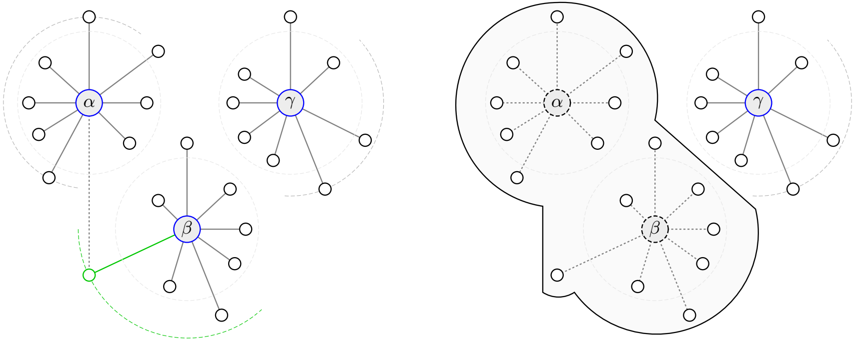
Improvement Criteria

$$f(x') < f(x) \vee f(x') = f(x), \mathcal{B}(x') \subseteq \mathcal{B}(x), \mathcal{J}(x') \subset \mathcal{J}(x) \tag{6}$$

where $\mathcal{B}(X)$ denote the set of bottleneck subsets in X, i.e., $\mathcal{B}(X) = \{k \in K : f(X_k) = f(X)\}$ and $\mathcal{J}(X) = \{j \in X_k : d_{jc(k)} = f(X), k \in \mathcal{B}(X)\}$. The incumbent solution X^{best} is updated if a better feasible solution is found according to the criterion (6) otherwise a shake of the solution X is applied.

Shake

We define an auxiliary mechanism that performs a partial shake of the current solution through an aggressive removal and reconstruction of several subsets, which diversifies the structure of the solution.



(a) Selection (b) Destruction

The heuristic was coded in C++, compiled with g++ 4.2 with the -O3 optimization level on 64 bits. ILOG CPLEX 12.5 is used for the exact method. Platform: Intel Core i5 2.4 GHz, 4 GiB RAM under OS X 10.7.5. A comparison of the proposed approach (QR) with the heuristic by Scaparra et al. (SP) and the exact method by Özsoy and Pınar (OP) is performed. For the experiments, we used seven different data sets:

- A: Beasley: 20 instances, with 50 and 100 demand nodes and 5 and 10 facilities to be located.
- B: Galvão and ReVelle: 8 instances, with 100 to 150 demand nodes and 5 to 15 facilities to be located.
- C: Lorena and Senne: 8 instances, with 100 to 402 demand nodes and 10 to 40 facilities to be located.
- $\alpha - \delta$: OR-Library, 40 instances for each subset, with 50 to 200 demand nodes and 5 to 80 facilities to be located.

Assessment of each component over all data sets, in terms of average relative optimality gap. Each column indicates the component omitted for experimentation.

Data set	Average relative gap (%)				Average time (s)			
	All	IGLS	VND	Shake	All	IGLS	VND	Shake
A	0.23	11.06	3.95	1.04	0.53	0.14	0.32	0.37
B	3.56	10.09	4.13	4.95	2.80	0.21	2.15	2.28
C	4.56	24.89	8.89	14.19	11.56	0.41	11.26	11.48
ORL-α	2.99	20.10	10.93	5.08	0.73	0.14	0.53	0.62
ORL-β	14.96	69.21	37.27	23.45	7.68	0.47	7.14	7.87
ORL-γ	13.71	121.09	55.84	34.62	12.42	0.73	11.65	12.55
ORL-δ	36.62	192.46	99.10	76.94	15.18	1.05	14.63	15.80

Table : Summary of component behavior over all data sets

Computational Results

n	p	Instance	Optimal	OP		SP		QR	
				gap %	Time (s)	gap %	Time (s)	gap %	Time (s)
50	5	cpmp01	29	0.00	0.19	0.00	0.45	0.00	0.30
		cpmp02	33	0.00	1.13	0.00	0.72	0.00	0.34
		cpmp03	26	0.00	0.20	0.00	0.56	0.00	0.31
		cpmp04	32	0.00	0.53	0.00	0.61	0.00	0.34
		cpmp05	29	0.00	1.02	0.00	0.69	0.00	0.33
		cpmp06	31	0.00	1.62	3.23	0.75	0.00	0.31
		cpmp07	30	0.00	0.51	0.00	0.91	0.00	0.37
		cpmp08	31	0.00	0.61	0.00	0.73	0.00	0.29
		cpmp09	28	0.00	0.74	3.57	0.91	0.00	0.33
		cpmp10	32	0.00	2.14	12.50	1.74	0.00	0.30
		Average		0.00	0.87	1.93	0.81	0.00	0.32
100	10	cpmp11	19	0.00	2.91	21.05	5.4	0.00	0.76
		cpmp12	20	0.00	2.91	10.00	5.74	0.00	0.80
		cpmp13	20	0.00	3.46	5.00	5.46	0.00	0.74
		cpmp14	20	0.00	2.15	10.00	5.28	0.00	0.70
		cpmp15	21	0.00	4.06	9.52	5.9	0.00	0.77
		cpmp16	20	0.00	6.96	10.00	7.04	0.00	0.82
		cpmp17	22	0.00	30.14	9.09	6.03	4.55	0.75
		cpmp18	21	0.00	6.50	4.76	4.74	0.00	0.67
		cpmp19	21	0.00	9.30	9.52	6.25	0.00	0.72
		cpmp20	21	0.00	12.25	0.00	5.93	0.00	0.70
		Average		0.00	8.06	8.90	5.78	0.45	0.74
		Overall average		0.00	4.47	5.41	3.29	0.23	0.53

Table : Comparison of methods on data set A

Computational Results

n	p	Instance	Optimal	OP		SP		QR	
				gap %	Time (s)	gap %	Time (s)	gap %	Time (s)
100	5	G1	94	0.00	4.49	3.19	4.71	1.06	1.56
100	5	G2	94	0.00	5.90	3.19	4.48	0.00	1.44
100	10	G3	83	0.00	121.44	9.64	8.01	6.02	2.06
100	10	G4	84	0.00	25.03	8.33	8.28	5.95	2.08
150	10	G5	95	0.00	190.95	5.26	22.61	2.11	3.16
150	10	G6	96	0.00	120.46	5.21	21.21	2.08	3.03
150	15	G7	89	0.00	60.62	8.99	28.31	5.62	4.61
150	15	G8	89	0.00	213.61	10.11	26.52	5.62	4.48
		Overall average		0.00	92.81	6.74	15.52	3.56	2.80

n	p	Instance	Optimal	OP		SP		QR	
				gap %	Time (s)	gap %	Time (s)	gap %	Time (s)
100	10	SJC1	364	0.00	195.16	26.67	8.79	7.48	0.89
200	15	SJC2	304	0.00	74.30	10.48	39.60	4.23	2.80
300	25	SJC3a	278	0.00	136.49	38.73	125.03	3.60	9.16
300	30	SJC3b	253	0.00	152.20	35.59	119.65	2.48	12.22
402	30	SJC4a	284	0.00	522.63	30.99	283.18	5.57	17.09
402	40	SJC4b	239	0.00	157.52	44.12	241.68	4.02	27.21
		Overall average		0.00	206.38	31.10	136.32	4.58	11.56

Table : Comparison of methods on data sets B–C

Comparison among methods for the all data sets in terms of their average relative optimality gap, running time, and memory usage. The memory statistic indicates the maximum resident set size used, in bits. That is, the maximum number of bits of physical memory that each approach used simultaneously.

Data set	Average gap (%)			Average time (s)			Average memory (bits)		
	OP	SP	QR	OP	SP	QR	OP	SP	QR
A	0.00	5.41	0.23	4.47	3.29	0.53	2.5E+7	4.3E+7	5.1E+5
B	0.00	6.74	3.56	92.81	16.37	2.80	4.7E+7	2.1E+8	5.6E+5
C	0.00	31.10	4.56	206.38	141.82	11.56	1.3E+8	4.7E+8	9.7E+5
ORL-α	0.00	12.89	2.99	144.48	14.54	0.73	7.9E+7	1.2E+8	5.5E+5
ORL-β	0.00	17.89	14.96	67.72	13.31	7.68	8.2E+7	6.6E+7	5.6E+5
ORL-γ	0.00	15.83	13.71	74.81	14.86	12.53	4.2E+7	6.8E+7	5.7E+5
ORL-δ	0.00	17.55	36.62	328.73	22.44	16.48	6.7E+7	1.0E+8	5.7E+5

Table : Summary of comparison among methods on all data sets

Asymptotic Analysis

Computational Results

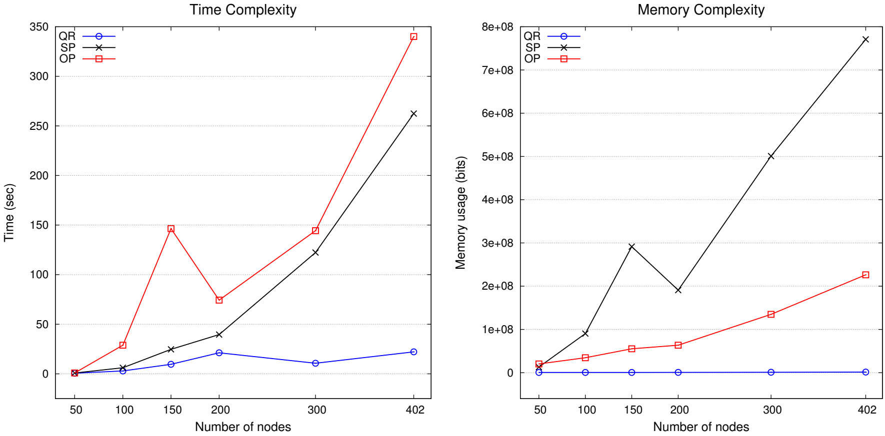


Figure : Comparison of the methods (asymptotic running time and used memory resources).

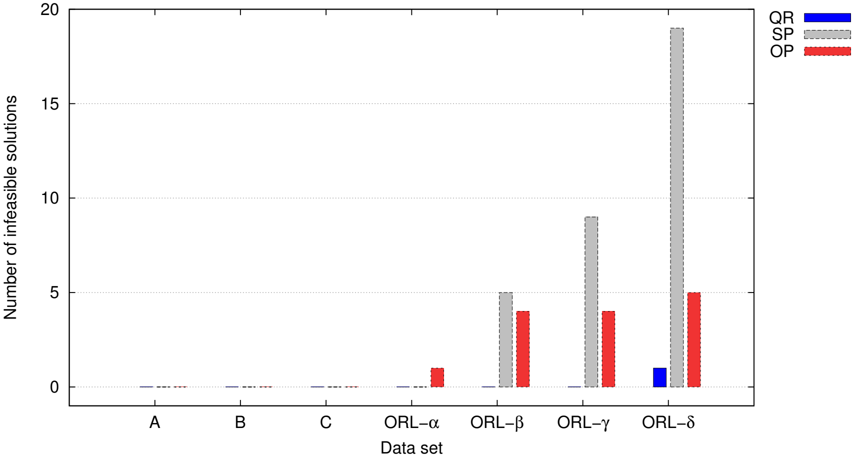


Figure : Comparison of the methods (number infeasible solutions).

Wrap-up

Conclusions

- The results clearly indicate that the proposed heuristic outperforms the best heuristic in terms of both solution quality and running time.
- The performance of the proposed heuristic is more robust than that of the exact method, requiring less time and memory to solve each instance obtaining solutions of reasonably good quality.

Acknowledgements

- Thank you!!
- Financial Support
  - CONACyT (grant CB-2011-01-166397 and scholarship for graduate studies).
  - Universidad Autónoma de Nuevo León (grant UANL-PAICYT CE728-11 and tuition waiver).
  - FIME-UANL (fee waiver).
- People
  - Juan A. Díaz, UDLAP, Mexico.
  - Maria Scaparra, Kent University, UK.
  - Elisa Schaeffer, UANL, Mexico.

- E-mail: roger@yalma.fime.uanl.mx
- URL: http://yalma.fime.uanl.mx/~roger

References

M. Scaparra, S. Pallottino, and M. Scutellà. Large-scale local search heuristics for the capacitated vertex p-center Problem. *Networks*, 43(4): 241–255, 2004.

F. A. Özsoy, and M. Ç. Pınar. An exact algorithm for the capacitated vertex p-center problem. *Computers & Operations Research*, 33(5):1420–1436, 2006.

M. Albareda-Sambola, J. A. Díaz, and E. Fernández. Lagrangean duals and exact solution to the capacitated p-center problem, *European Journal of Operational Research* 201(1): 71-81, 2010.

R. Ruiz and Stützle, T.: A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177(3): 2033–2049, 2007.

P. Hansen, N. Mladenović: Variable neighborhood search: Principles and applications. *European Journal of Operational Research* 130(3): 449–467, 2001.

M.E. Dyer, A. Frieze: A simple heuristic for the p-center problem. *Operations Research Letters* 3(6), 285–288, 1985.