

A Local Search procedure for the Open vehicle Routing Problem

Majid Salari, Paolo Toth, Andrea Tramontani
DEIS, University of Bologna, Italy.

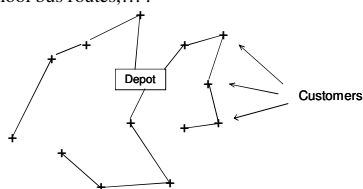
October 12-15, 2008
Washington, DC

Outline

- The Open Vehicle Routing Problem (OVRP).
- An ILP improvement procedure for OVRP.
- Computational results.
- Conclusions and Future directions.

Open Vehicle Routing Problem(OVRP)

- A variant of the “classical” Vehicle Routing Problem (VRP) in which the vehicles are not required to return to the depot after completing their service.
- Application in the companies which do not own a vehicle fleet
- In the problems consist of pick up and delivery, planning of a set of school bus routes,



Open Vehicle Routing Problem(OVRP)

General Constraints:

- Each path associated with one vehicle.
- Each route must have a total duration not exceeding the given limit D of the vehicles.
- Each route must have a total capacity not exceeding the given limit Q of the vehicles.

Objectives:

- Minimize the number of used vehicles as the first objective and
- Minimize the traveling cost.

An ILP improvement procedure for OVRP

Given a feasible initial solution z for the OVRP,

1) Selection phase:

Select a set of customers F , each customer with a probability p .

2) Extraction phase:

Extract the customers in F and build a restricted solution $z(F)$ by short-cutting the extracted nodes .

In $z(F)$, consider the edges incident at the depot as directed arcs leaving the depot, and add artificial arcs (with cost 0) connecting the last customer of each route with the depot.

Each edge/arc in $z(F)$ is then viewed as an insertion point i which can allocate one or more nodes in F . Denote with I the set of all the insertion point.

General description of the algorithm

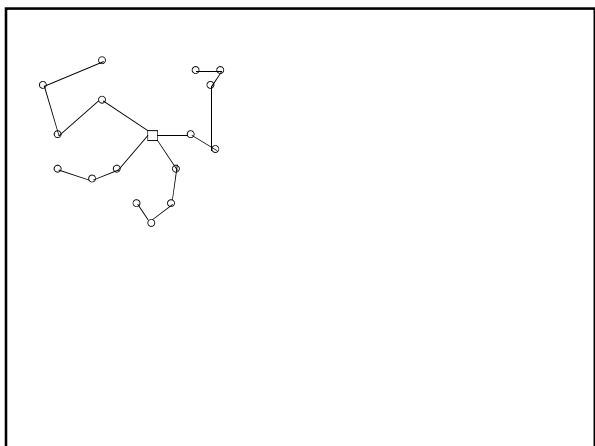
3) Recombination phase:

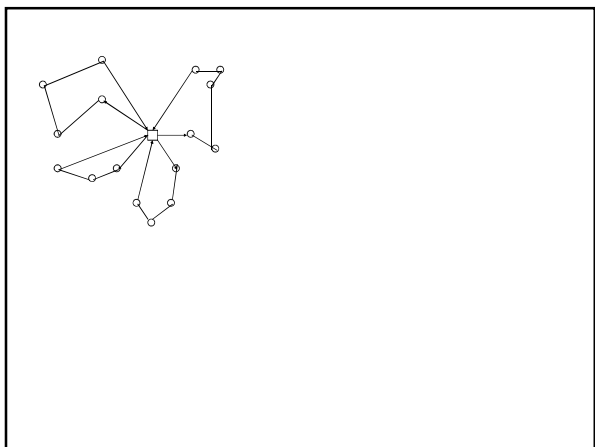
Generate a pool of sequences S through the recombination of the nodes in F (i.e., a pool of elementary paths through the nodes of F).

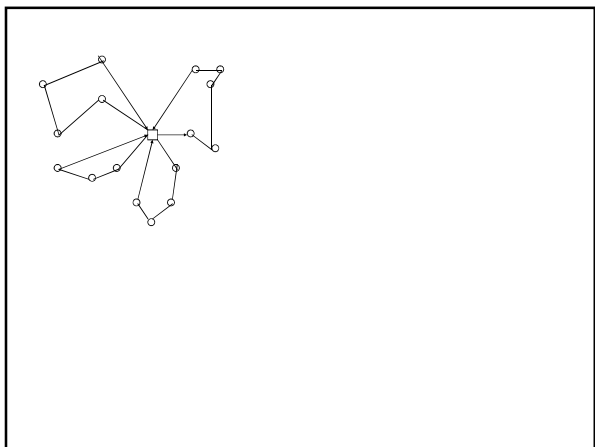
4) Reallocation phase:

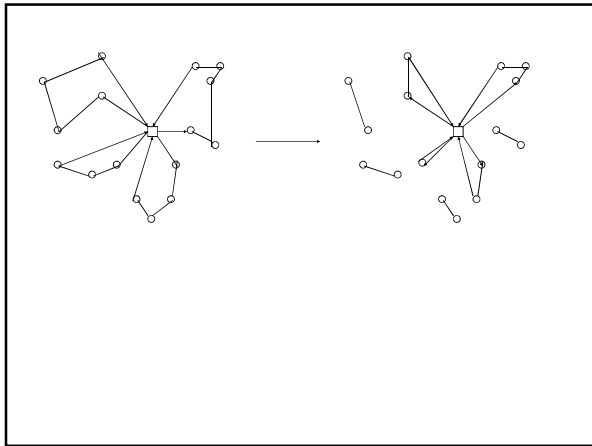
Reallocate all the extracted nodes to the restricted solution (through some sequences in S and some insertion points in I) in an optimal way, by solving an ILP model (*Reallocation Model*)

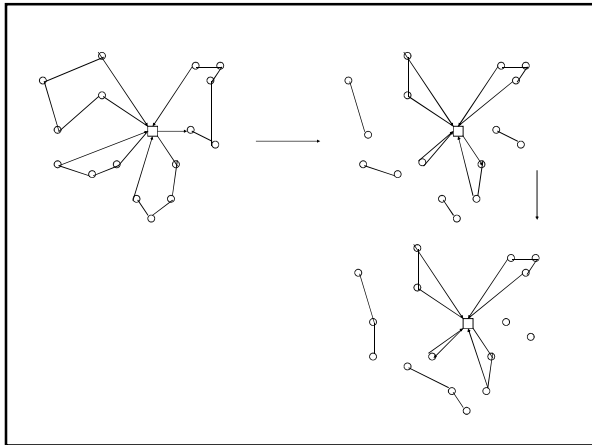
Approach proposed by De Franceschi et al. [4] and deeply investigated by Toth and Tramontani [22] in the context of the classical VRP.

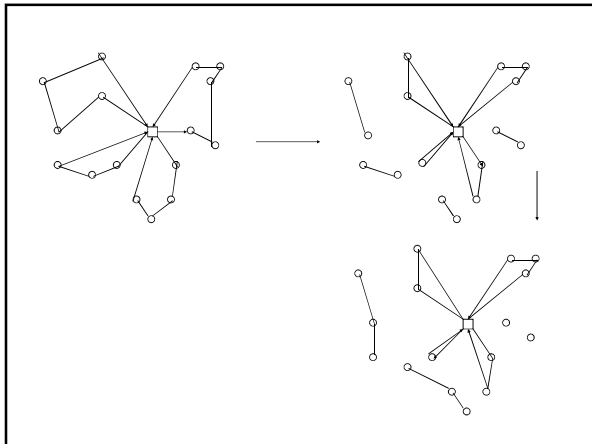


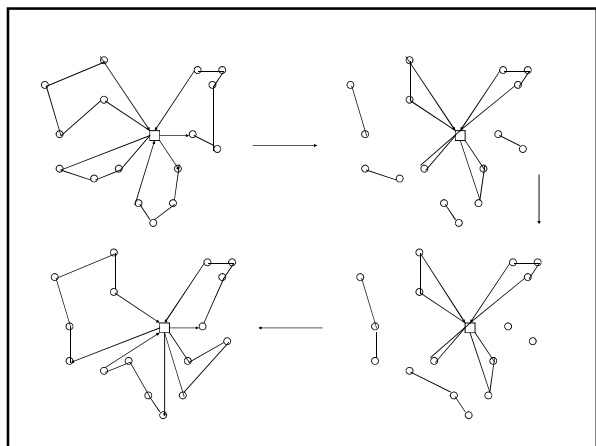


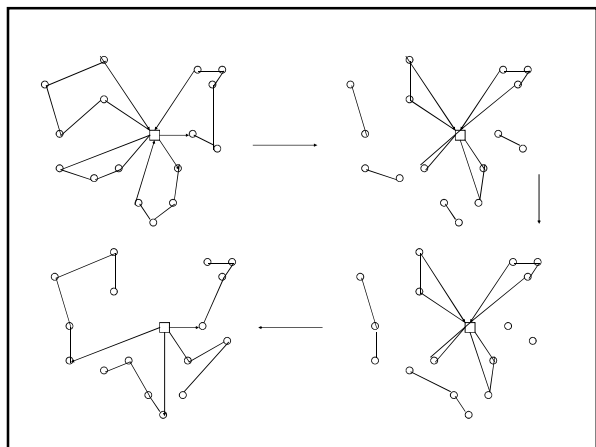












Reallocation Model:

Some notations and definitions.

- $z(F)$: Restricted solution obtained by extracting some nodes from the initial solution.
- \mathcal{R} : the set of routes in the restricted solution.
- $I = I(z, F)$: the set of all edges or the set of insertion points in $z(F)$.
- S_i (for each insertion point i): the subset of the *sequences* S which can be allocated to the insertion point i in I
- $c(s)$ and $d(s)$, overall cost and request of the sequence $s \in S$.

Reallocation Model:

Some notations and definitions.

- γ_s : Extra cost for assigning sequence s to insertion point i .
- $I(r)$: Set of all the insertion points associated with $r \in \mathfrak{R}$.
- Q and D maximum capacity and travel length of the vehicles.
- $\bar{q}(r)$ and $\bar{c}(r)$ are the total request and demand computed for route $r \in \mathfrak{R}$.

Reallocation Model:

$$x_{si} = \begin{cases} 1 & \text{if sequence } s \in S_i \text{ is allocated to insertion point } i \in I, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$\sum_{s \in \mathfrak{R}} \bar{c}(r) + \min_{i \in I} \sum_{s \in S_i} \gamma_s x_{si} \quad (2)$$

Subject to:

$$\sum_{i \in I} \sum_{s \in S_i} x_{si} = 1 \quad v \in F, \quad (3)$$

$$\sum_{s \in S_i} x_{si} \leq 1 \quad i \in I, \quad (4)$$

$$\sum_{i \in I(r)} \sum_{s \in S_i} q(s) x_{si} \leq Q - \bar{q}(r) \quad r \in \mathfrak{R}, \quad (5)$$

$$\sum_{i \in I(r)} \sum_{s \in S_i} \gamma_s x_{si} \leq D - \bar{c}(r) \quad r \in \mathfrak{R}, \quad (6)$$

$$x_{si} \in \{0, 1\} \quad s \in S_i, \quad (7)$$

Recombination phase

1) Initialization

- Initialize each subset S_i with the *basic* sequence extracted from i and
- The feasible singleton sequence (v) with the minimum insertion cost.
- Initialize Linear Programming relaxation of the Reallocation Model (LRM) with the initial set of variables corresponding to the current subsets, and solve LRM.

Column generation

- For each insertion point $i \in I$ add to the pool of variables all the sequences $\langle v_1, v_2 \rangle$, $(v_1, v_2 \in F)$ such that it's reduced cost is less than a given threshold RC_{\max} .
- For each insertion point $i \in I$, solve the column generation problem associated with i , adding to S_i all the sequences corresponding to elementary path in F whose associated variables have a reduced cost under a given threshold RC_{\max} .

Column generation

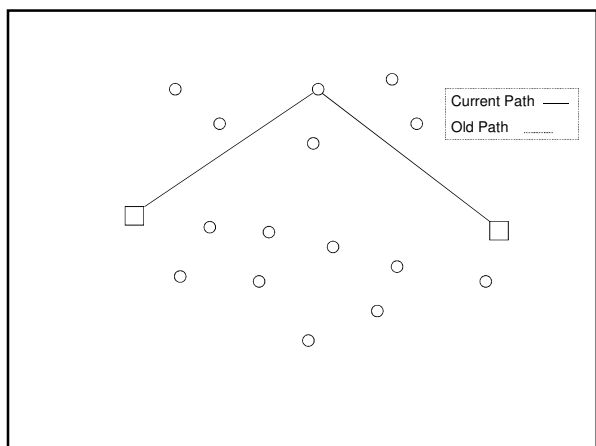
- For any fixed insertion point $i \in I$, the column generation problem associated with i in LRM is in practice a Resource Constrained Elementary Shortest Path Problem (RCESPP), which usually arises in the Set Partitioning formulation of the classical VRP.
- Here, for each insertion point, we solve the corresponding RCESPP through a simple greedy heuristic, with the aim of finding as many variables with small reduced cost as possible.

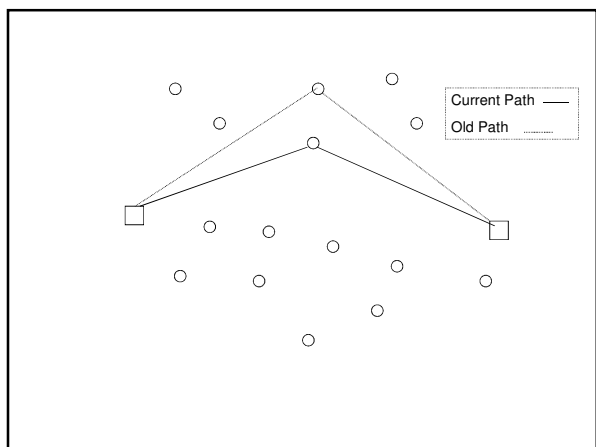
Column generation (*Heuristic Algorithm*)

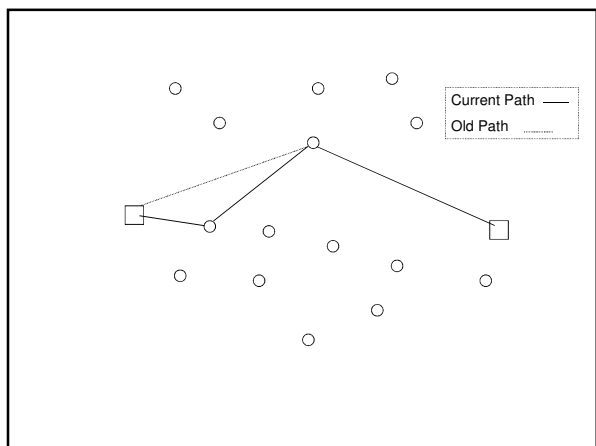
Given a graph $G(x)$ and a starting feasible path $P = \langle a_i, v, b_i \rangle$, with $v \in F$:

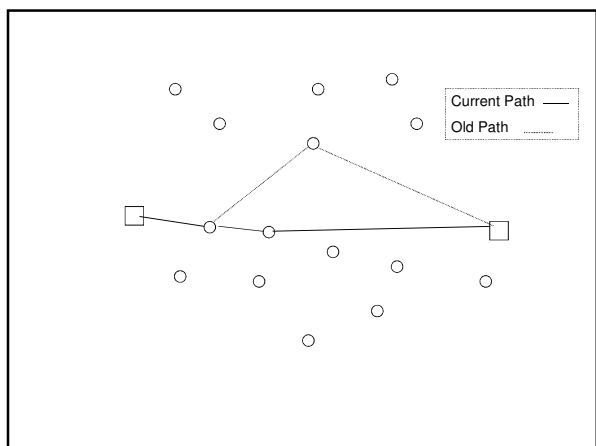
- 1) Evaluate all the 1-1 feasible exchanges between all the nodes $w \in P$ and all the nodes $v \in P$ and select the best one, if such an exchange leads to an improvement perform it and repeat from 1.
- 2) Evaluate all the feasible insertions of all the nodes $w \in P$ in all the edges $\{v_1, v_2\} \in P$ and select the best one, force such an insertion even if it leads to a worsening of the current path and repeat from 1.

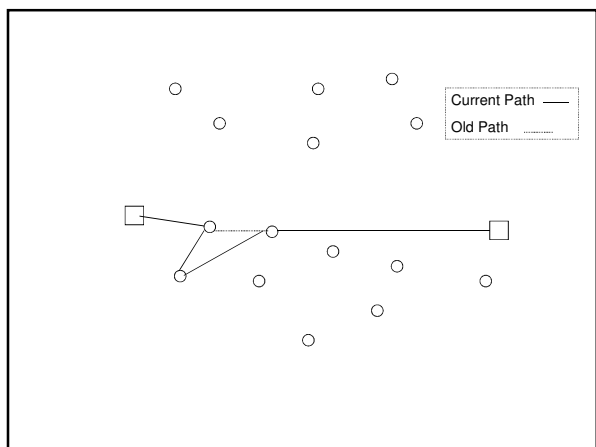
At any time a new path is generated, the corresponding variable is added to the variable pool V if its reduced cost is smaller than a certain threshold RC_{\max} .

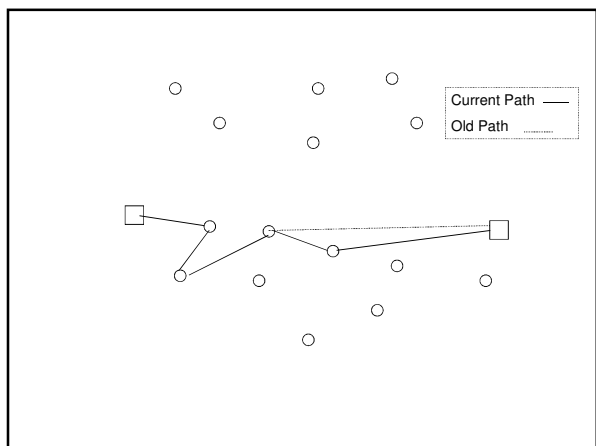


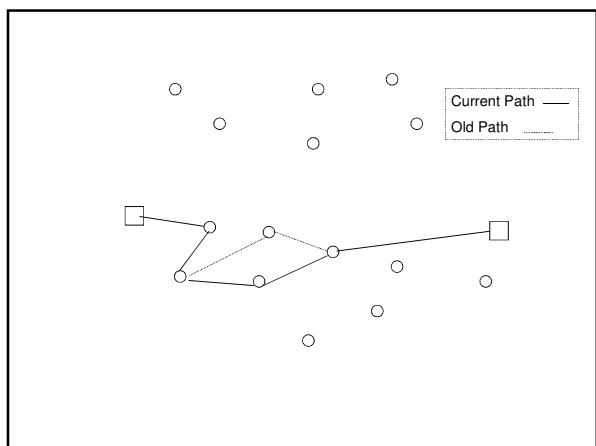


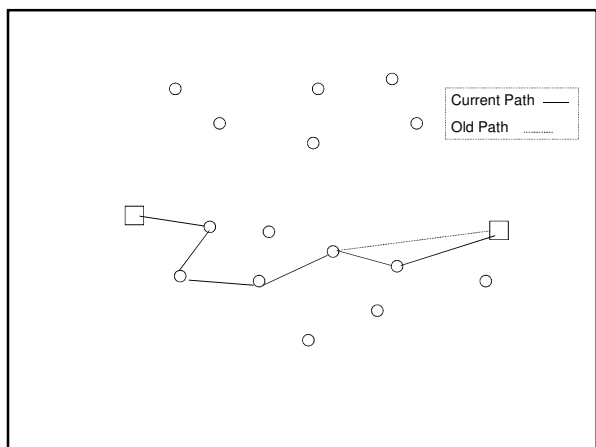


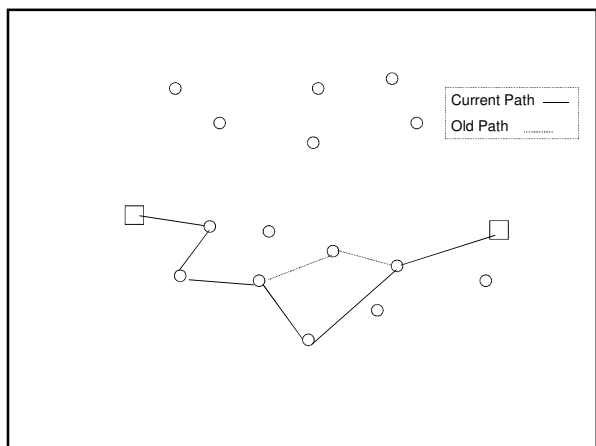


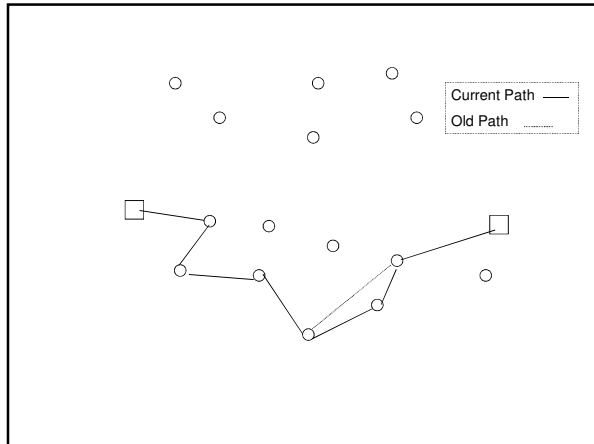


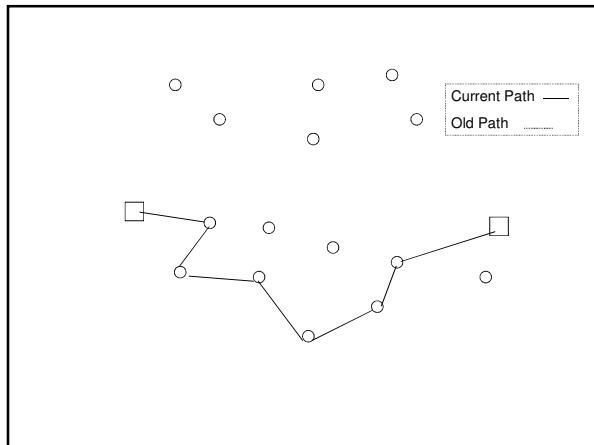












Algorithm:

1. (Initialization) Set $kt := 0$ and $kp := 0$. Take z_0 as the incumbent solution and initialize the current solution z_c as $z_c := z_0$.
2. (Node selection) Build the set F by selecting each customer with a probability p .
3. (Node extraction) Extract the nodes selected in the previous step from the current solution z_c and construct the corresponding restricted OVRP solution $z_r(F)$, obtained by shortcutting the extracted nodes.
4. (Reallocation) Define the subsets $S(i \in I(z_c, F))$ as described. Build the corresponding Reallocation Model (2)-(7) and solve the model by using a general-purpose ILP solver. Once an optimal ILP solution has been found, construct the corresponding new OVRP solution and possibly update z_c and z_0 .
5. (Termination) Set $kt := kt + 1$. If $kt = kt_{max}$, terminate.
6. (Perturbation) If z_c has been improved in the last iteration, set $kp := 0$; otherwise set $kp := kp + 1$. If $kp = kp_{max}$, "perturb" the current solution and set $kp := 0$. In any case, repeat from step 2.

Perturbation:

- If the current solution is not improved after a given number of consecutive iterations we do the random perturbation.
- Randomly extract np customers from the solution.
- Reinsert each extracted customer, in turn, in its best feasible position.
- If a customer cannot be inserted in any currently non-empty route (due to the capacity and/or distance restrictions), a new route is created to allocate the customer.

Computational Results:

- Benchmark instances, taken from Christofides et al. (instances C1-C14), Fisher (instances F11-F12) and Golden.
- The number of customers ranges from 50 to 480.
- C1-C5, C11-C12 and F11-F12 and Golden instances have only capacity constraints, while C6-C10 and C13-C14 are the same instances as C1-C5 and C11-C12, respectively, but with both capacity and distance constraints.
- High quality initial solutions taken from Fu et al. [9, 10].

Computational Results:

- Test on a Pentium IV 3.4 GHz with 1 GByte RAM using ILOG Cplex 10.0 as ILP solver.
- Parameters: $RC_{\max} = 1$, $p = 0.5$, $KP_{\max} = 50$, $KT_{\max} = 5000$.
- $np \in [np_{\min}, np_{\max}]$ (randomly) in which $np_{\min} = 15$, $np_{\max} = 25$.

Table 1: Computational results on benchmark instances starting from the solutions by Fu et al.

Instances	Prev.best sol.	Initial solution		Final solution	Best time	Final time
		Cost	Time			
C1	416.06*	416.06*	32.0	-	-	-
C2	567.14	567.14	312.0	-	-	84.2
C3	639.74*	641.88	928.0	369.74*	106.0	119.9
C4	733.13	738.94	272.0	733.13	21.2	156.6
C5	869.25	878.95	2476.0	868.81	10.3	220.3
C6	412.96	412.96	24.0	-	-	45.1
C7	568.49	568.49	240.0	-	-	83.1
C8	644.63	646.31	416.0	644.63	0.1	136.2
C9	756.14	761.28	1864.0	756.14	102.7	255.5
C10	875.07	903.10	2076.0	878.54	323.9	460.2
C11	682.12	717.15	924.0	683.64	165.8	198.8
C12	534.24*	534.71	168.0	534.24*	1.6	94.0
C13	896.50	917.90	3284.0	894.19	475.0	1165.3
C14	591.87	600.66	100.0	591.87	293.8	354.7
F11	177.00*	177.00*	16.0	-	-	-
F12	769.66	770.07	1136.0	769.55	77.8	148.2
			Pentium IV 3 GHz	Pentium IV 3.4 GHz		

*: Provably optimal solutions
 "._": Final solution costs equal to the previously best known ones.
 new best solutions are in bold face.

Table 1: results on 8 large instances of Golden.

Problem Name	Best known	Starting Value	final	Best Time	Final Time
Golden201	6018.52	6018.52	---	---	182.23
Golden241	4584.55	4584.69	4573.53	34.59	283.96
Golden281	7731.46	7731.46	---	--	304.60
Golden321	7260.59	7260.59	7253.91	94.45	438.93
Golden361	9167.20	9167.20	9165.40	41.37	499.62
Golden401	9803.80	9805.45	---	---	581.31
Golden441	10348.57	10348.57	10344.37	8.56	652.96
Golden481	12420.16	12420.16	---	---	623.56

Different tuning of parameters:

- Considering the different probabilities of extracting the nodes:

- $z_{avg}(\bar{p})$ is the average of final value obtained on 14 instances.
- $total_time_{avg}(\bar{p})$ is the corresponding average CPU time in seconds.

P	0.3	0.5	0.7
$z_{avg}(\bar{p})$	684.55	681.65	683.32
$total_time_{avg}(\bar{p})$	71.9	251.6	460.0

Conclutions and Future directions :

- The proposed method is very effective in improving the starting solution, even if it is of very-good quality.
- More sophisticated criteria for removing customers from the current solution.
- More sophisticated algorithms for solving the column generation problem related to the ILP model.
- The overall procedure can be considered as a general framework and it could be extended to cover other variants of Vehicle Routing Problems such as :Vehicle Routing Problems with heterogenous vehicles and multidepot Vehicle Routing Problems.

References:

1. Christofides, N., Mingozi, A., Toth, P.: The vehicle routing problem. In: Christofides, N., Mingozi, A., Toth, P., Sandi, C. (eds.), Combinatorial Optimization, pp.313-338. Wiley, Chichester (1979)
2. De Franceschi, R., Fischetti, M., Toth, P.: A new ILP-based refinement heuristic for vehicle routing problems. Mathematical Programming 105, 471-499 (2006)
3. Toth, P., Tramontani, A.: An integer linear programming local search for capacitated vehicle routing problems. To appear in: Golden, B., Raghavan S., Wasil, E.(eds.), The Vehicle Routing Problem: Latest Advances and New Challenges. Springer Verlag, New York (2008).
4. Fu, Z., Eglese, R., Li, L.Y.O.: A new tabu search heuristic for the open vehicle routing problem. Journal of the Operational Research Society 56, 267-274 (2005).
5. Fu, Z., Eglese, R., Li, L.Y.O.: Corrigendum: A new tabu search heuristic for the open vehicle routing problem. Journal of the Operational Research Society 57, 1018 (2006).

Thanks for your attention

Majid Salari

DEIS, University of Bologna,

Majid.salari2@unibo.it