Metaheuristics Network

Contract Number: HPRN-CT-1999-00106

Vehicle Routing Problems (VRPs)

Technische Universiteit Eindhoven.

Short Course; November 28-29 2000

Luca Maria Gambardella

IDSIA Istituto Dalle Molle di Studi sull'intelligenza Artificiale

Galleria 2, 6928 Manno-Lugano, CH

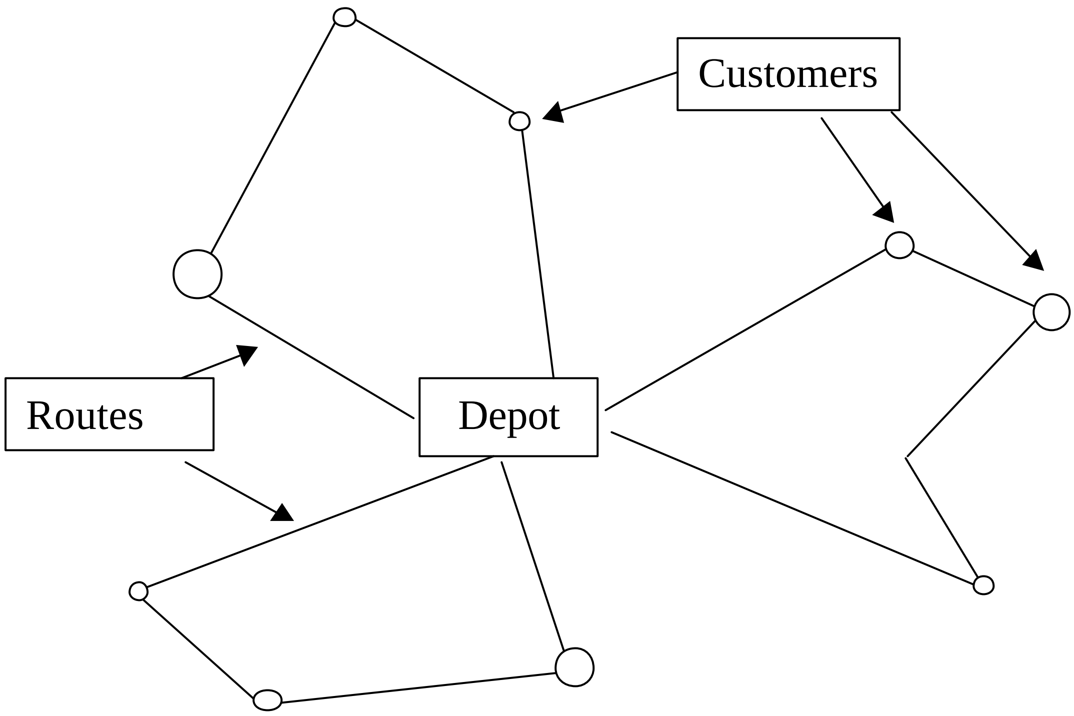
luca@idsia.ch

http://www.idsia.ch/luca

Vehicle Routing Problems (VRPs)

- VRP is a generic name given to a class of problems in which customers are visited by vehicles (first formulated by Dantzig and Ramser in 1950)

- The problem is to design routes for the vehicles so as to meet the given constraints and objectives minimising a given objective function.



Features

- *Depots* (number, location)
- *Vehicles* (capacity, costs, time to leave, driver rest period, type and number of vehicles, max time)
- *Customers* (demands, hard or soft time windows, pickup and delivery, accessibility restriction, split demand, priority)
- *Route Information* (maximum route time or distance, cost on the links)

Objective Functions (also multiple objectives)

- Minimise the total travel distance,
- Minimise the total travel time
- Minimise the number of vehicles

CVRP (capacitated) is defined on a graph G=(V,E) where V={0, …, n} is the vertex set where 0 is the depot.

To each vertex V\{0} is associated a non negative demand q_i and to each edge (i,j) is associated a cost c_{ij}.

Goal: design *m* vehicle routes of least cost, each starting and ending at the depot, such as:

- Each customer is visited only once
- The total demand of any route does not exceed the vehicle capacity *Q*
- The length of any route does not exceed a pre-set maximal route length *L*
- In some version *m* is fixed a priori in other is a decision variable

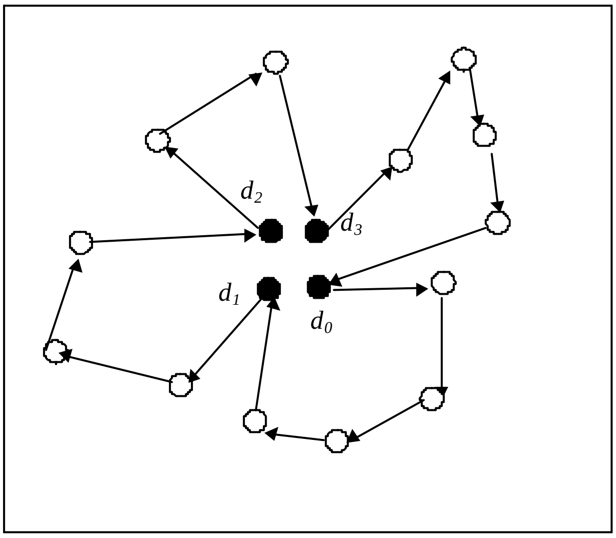
Relation with TSP

VRP is a generalisation of the Travelling Salesman Problem (TSP) therefore is NP-Hard.

TSP = VRP with one vehicle with no limits, no (any) depot, customers with no demand

Multiple TSP = one origin and *m* salesman = VRP with a depot, *m* vehicles with no limits, customers with no demand.

MTSP is transformable in a TSP by adding *m* identical copies of the origin and making costs between copies infinite.



Solution Techniques for VRP

Exact Approach (up to 100 nodes)
- Branch and bound (Fisher 1994)

Approximation
- Clark and Wright (1964)
- Hierarchical Approach (split + TSP)
  - Fisher & Jaikumur (1981)
  - Taillard (1993)
- Multi-route Improvement Heuristics
  - Kinderwater and Savelsbergh (1997)

MetaHeuristics
- Tabu search, Rochat and Taillard (1995)
- Constraint Programming, Shown (1998)
- Tabu search Kelly and Xu (1999)
- Granular Tabu, Toth & Vigo (1998)
- Ant System, Gambardella & al. (1999)

Tour Construction Heuristic

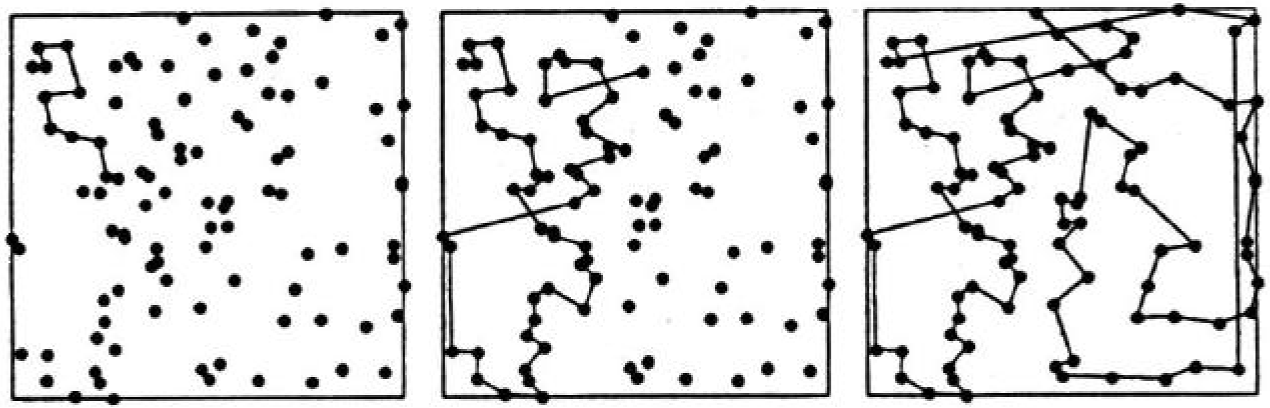


Figure 1. The Nearest Neighbor heuristic (Bentley 1992)

NN (Flood, 1956)

1. Randomly Select a starting node
2. Add to the last node the closest node until no more node is available
3. Connect the last node with the first node

Running time $O(N^2)$

Worst-Case Guarantee (WCG) = $NN(I)/OPT(I) \leq 0.5(\lceil \log_2 N \rceil + 1)$

Heuristics that grow tours

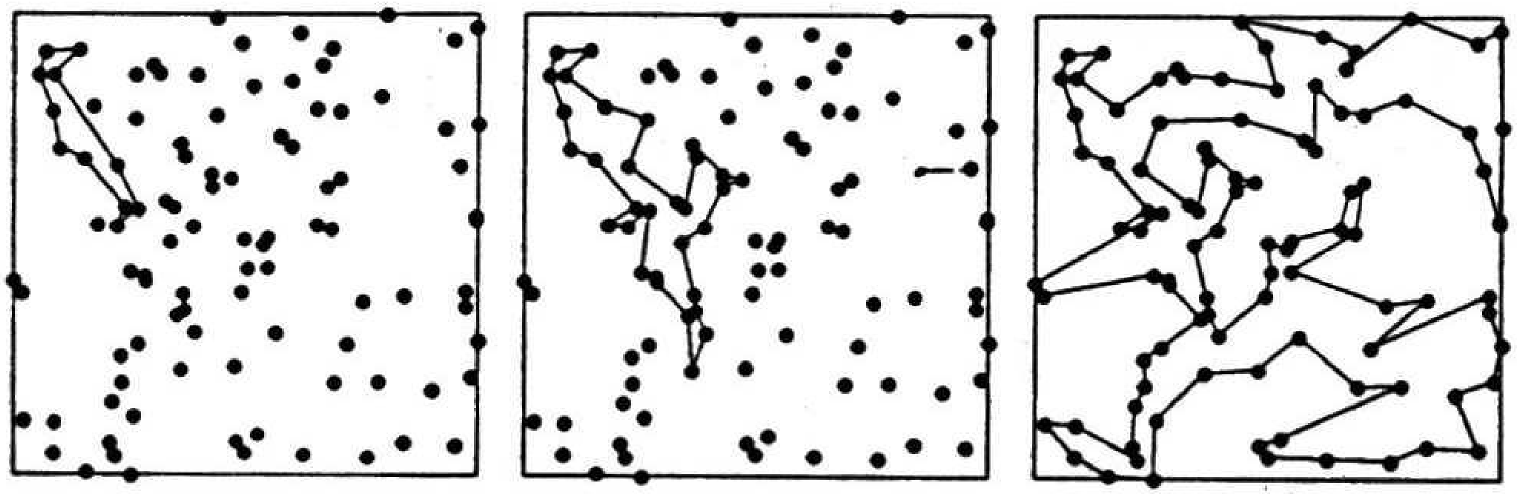
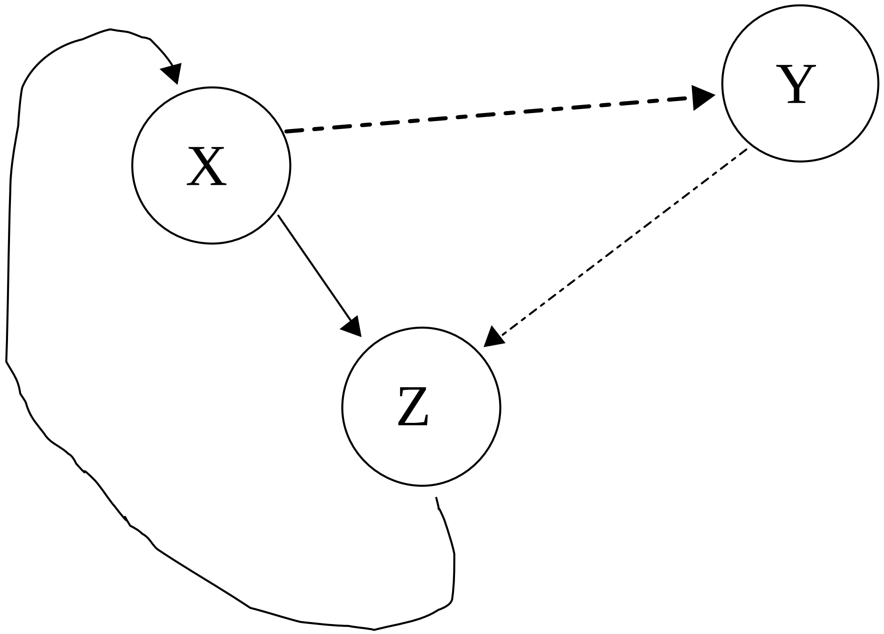


Figure 8. The Nearest Addition heuristic. (Bentley 1992)

NA

1. Select a node and is closest node and build a tour of two nodes
2. Insert in the tour the closest node Y until no more node is available

Running time $O(N^3)$



Y is the node that minimise

Dist(X,Y)+Dist(Y,Z)-DIST(X,Z)

(X,Z) is deleted and

(X,Y) and (Y,Z) are added to the tour

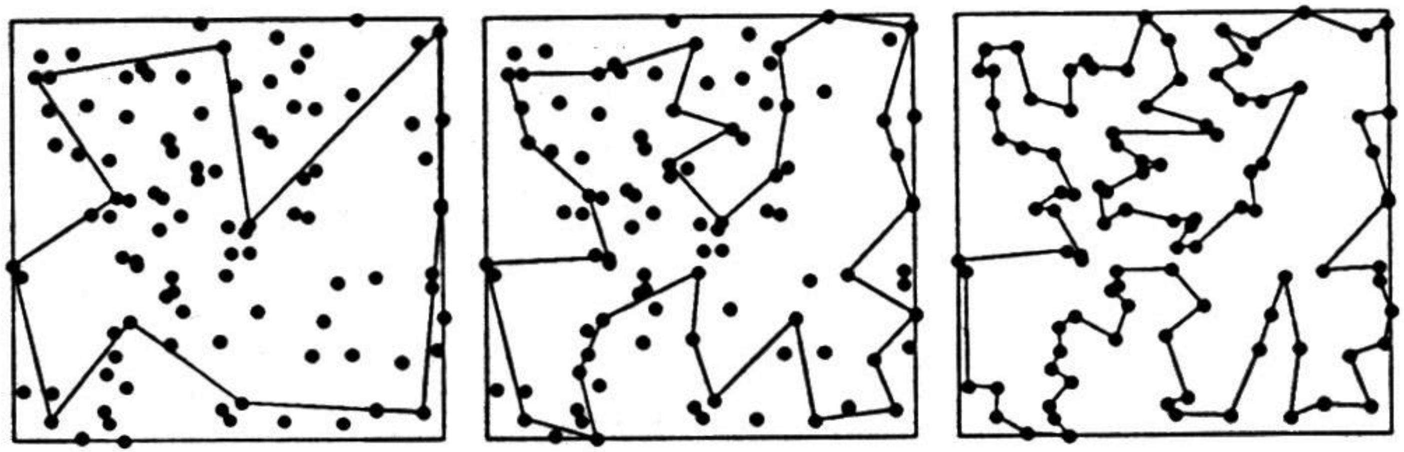


Figure 11. The Farthest Addition heuristic. (Bentley 1992)

FA O(N^3)

1. Select a node and is farthest and build a tour of two nodes
2. Insert in the tour the farthest node Y until no more node is available

FA is more efficient than NA because the first few farthest points sketch a broad outline of the tour that is refined after.

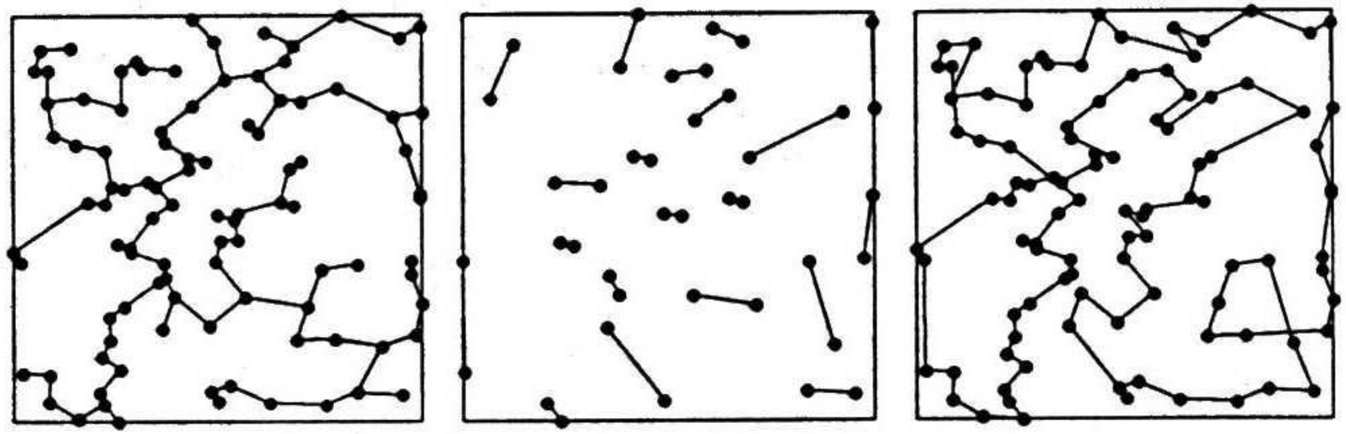


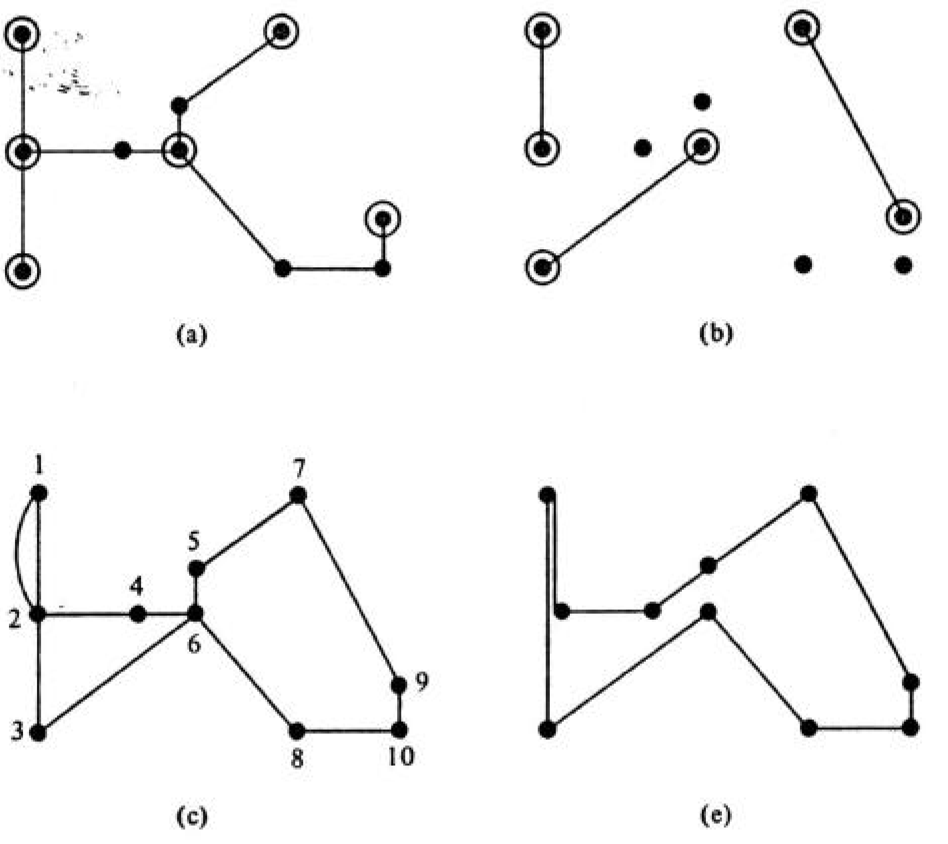
Figure 19. Christofides' heurist **1976**, (Bentley 1992)

1. Find the minimum spanning tree *T*. $O(N^2)$
2. Find nodes in *T* with odd degree and find the shortest complete matching *M* in the complete graph consisting of these nodes only. Let *G* be the multigraph all nodes and edges in *T* and *M*. $O(N^4)$
3. Find an Eulerian walk (each node appears at least once and each edge exactly once) on *G* and an embedded tour. O(N)

Running time $O(N^4)$

WCG = CH(I)/OPT(I) $\leq$ **2**

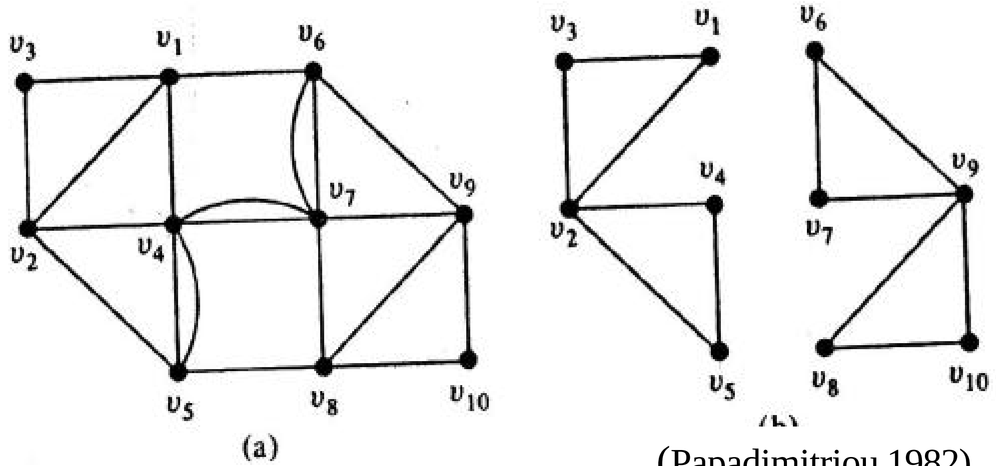
```
procedure Euler(v1)
(comment: it returns an Eulerian walk of the connected
          component of G containing v1)
  begin
  if v1 has no edges then return [v1] (comment: the empty walk)
    else
    begin
      starting from v1 create a walk of G, never visiting the
      same edge twice, until v1 is reached again;
      let [v1,v2, . . . ,vn,v1] be this walk;
      delete [v1,v2], . . . ,[vn,v1] from G;
      return [Euler(v1),Euler(v2), . . . ,Euler(vn),v1]†
    end
  end
```



[1, 2, 3, 6, 8, 10, 9, 7, 5, 6, 4, 2, 1] (Papadimitriou 1982)

(d)

Starting from this multigraph



(Papadimitriou 1982)

Euler(1)=[1,6,7,4,7,8,5,4,1] edges are deleted (b)
Euler(1)=[1,3,2,4,5,2,1]
Euler(6)=[6,7,9,6]
Euler(7)=[7], Euler(4)=[4], Euler(7)=[7]
Euler(8)=[8,9,10,8]
Euler(5)=[5], Euler(4)=[4], Euler(1)=[1]

Eulerian Walk =
[1,3,2,4,5,2,1,6,7,9,6,7,4,7,8,9,10,8,5,4,1]

Path = [1,3,2,4,5,2,1,6,7,9,6,7,4,7,8,9,10,8,5,4,1]

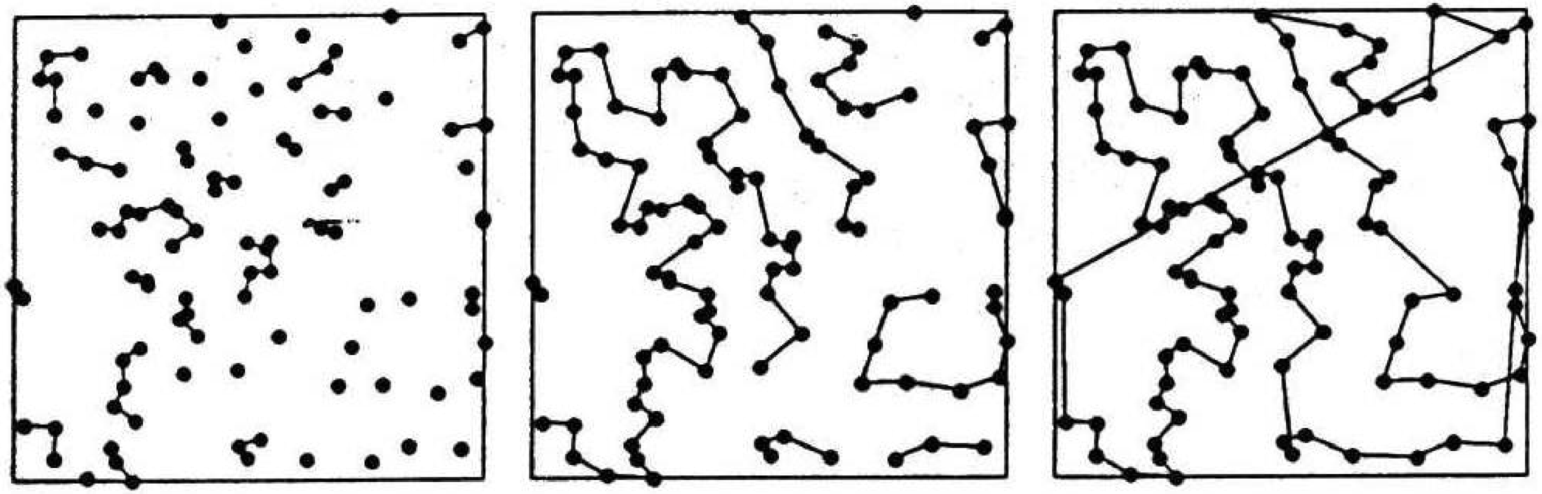
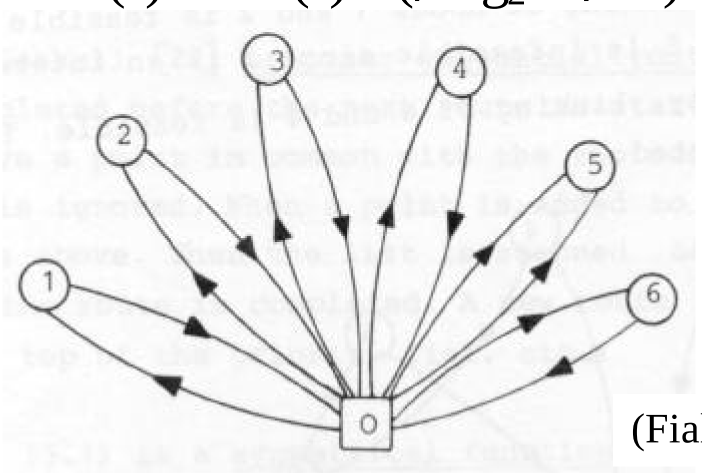


Figure 5. The Multiple Fragment heuristic (Bentley 1992)

1. The problem instance is seen as a complete graph
2. A tour is an Hamiltonian cycle in this graph (a connected collection of edges in which every city as degree 2)
3. We start with the shortest edge ad we add the edges in increasing order only if they do not create a 3-degree city

Clarke-Wright Saving Heuristic (1964) proposed for VRP

$$CW(I)/OPT(I) \leq (\lceil \log_2 N \rceil + 1)$$



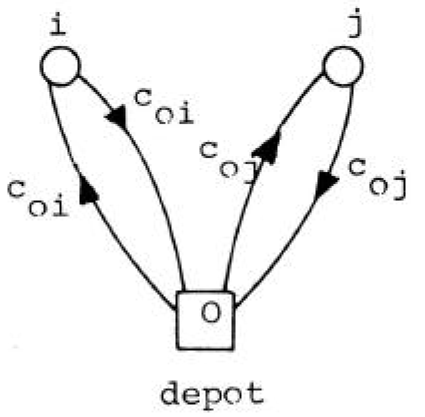
(Fiala 1978)

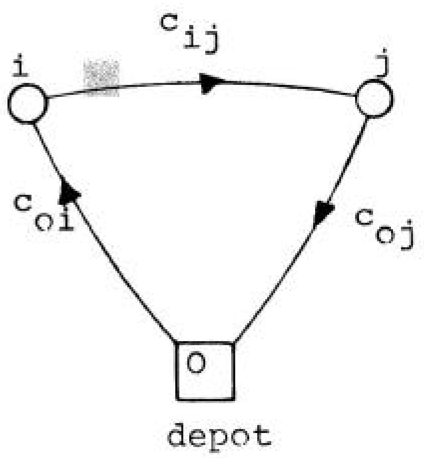
1. Start with an initial allocation of one vehicle to each customer (0 is the depot for VRP or any chosen city for TSP)

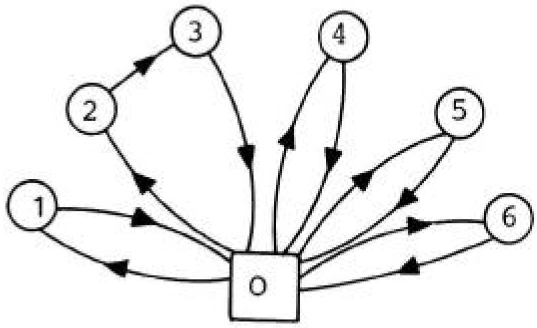
2. Calculate saving $s_{ij}=c_{0i}+c_{0j}-c_{ij}$ and order the saving in increasing order

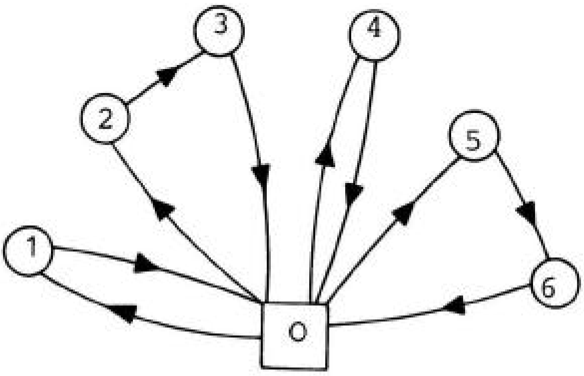
3. At each step find the largest saving s_{ij} where:
   i and j are not in the same tour
   neither i and j are interior to an existing route
   vehicle and time capacity are not exceeded

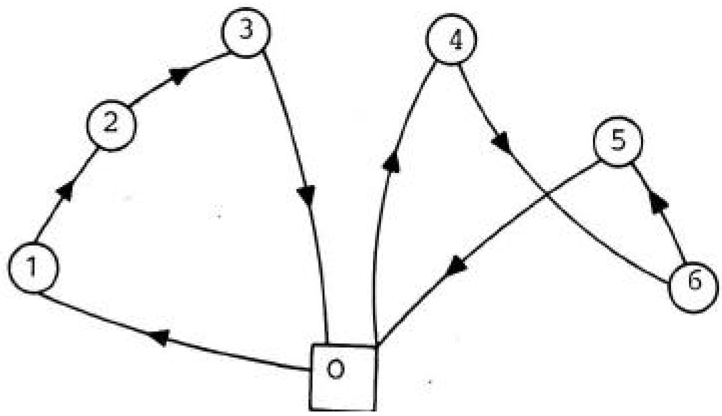
4. link *i* and *j* together to form a new tour (replacing to other routes)









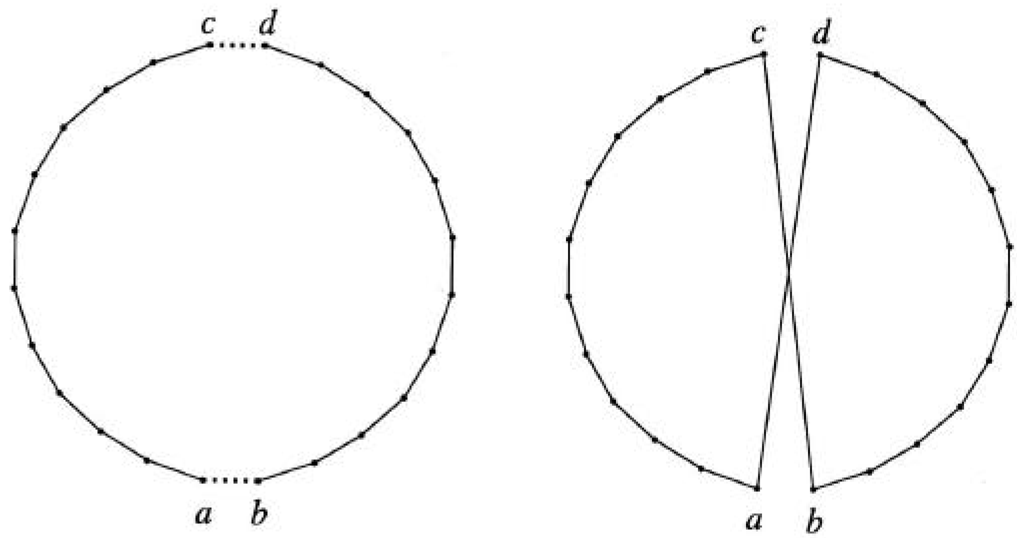


(Fiala 1978)

Tour Improvement Heuristics

- Start from an given tour and compute the best (the first) edge exchange that improves the tour.
- Execute this exchange and search for another exchange until no improvement is possible

2-OPT

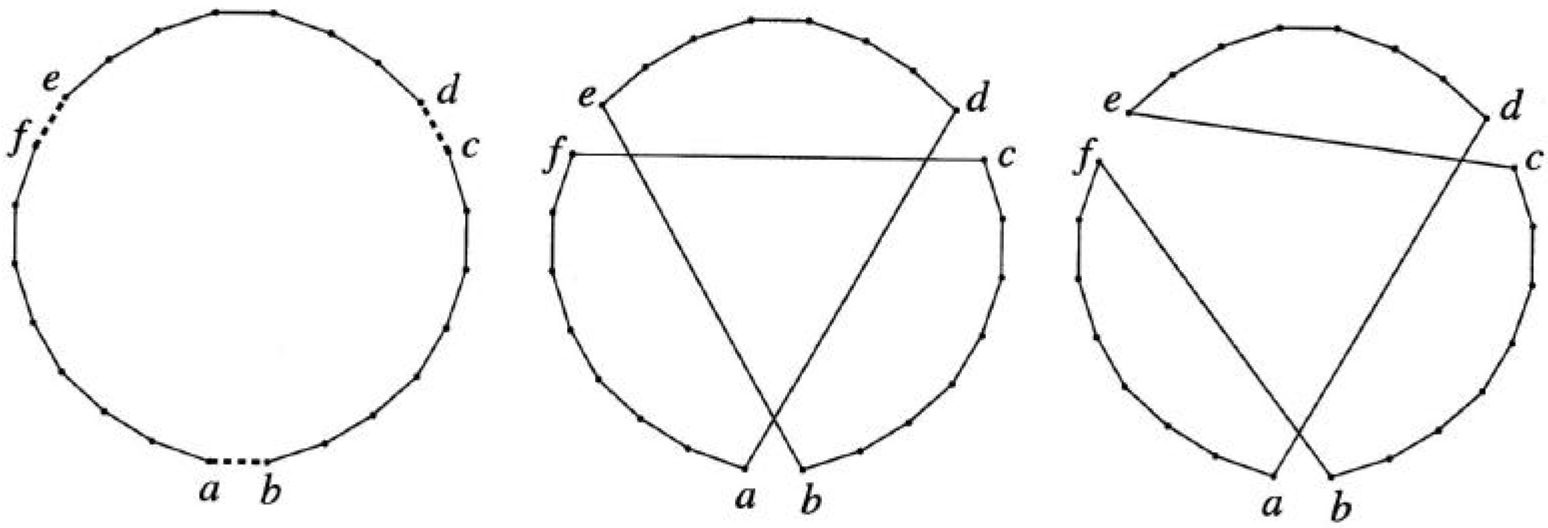


(Johnson 1997)

GAIN = (a,d)+(b,c)-(c,d)-(a,b)

Subpath (b,…,d) is reverted

3-OPT



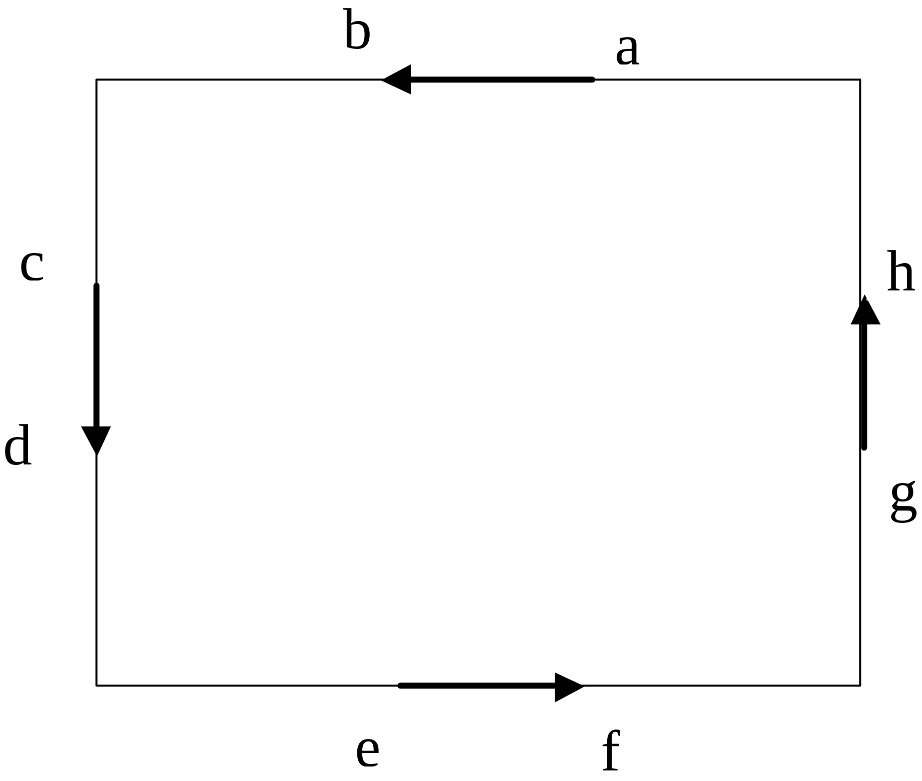
(Johnson 1997)

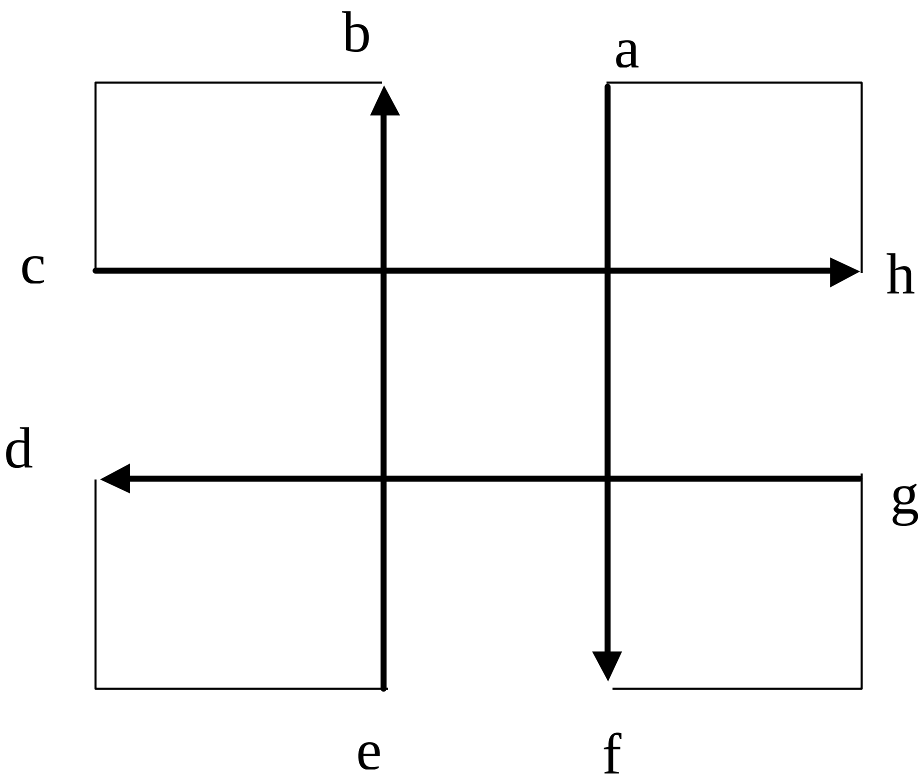
Two possible new tours

GAIN1 = (a,d)+(e,b)+(c,f)-(a,b)-(c,d)-(e,f) no path is reverted

GAIN2 = (a,d)+(e,c)+(b,f)-(a,b)-(c,d)-(e-f) path (c,…,b) is reverted

4-opt (double bridge, no path is reverted)





(Bentley 1992)

Table II. Robustness of the Heuristics

Heuristic Name	Distribution No. 1	2	3	4	5	6	7	8	9	10	Maximum Ratio
nn	0.68	0.60	1.01	1.06	0.64	0.70	1.00	1.01	1.03	0.73	1.06
denn	0.68	0.60	1.05	1.01	0.64	0.68	0.98	1.07	1.02	0.76	1.07
mf	0.64	0.48	1.01	1.05	0.62	0.65	1.01	0.71	1.03	0.64	1.05
na	0.82	0.56	1.07	1.56	0.61	1.03	1.36	1.11	1.43	1.23	1.56 +
fa	0.65	0.64	0.96	1.04	0.65	0.67	0.94	1.22	1.03	0.93	1.22
ra	0.56	0.61	1.01	1.01	0.57	0.60	0.98	1.35	1.06	0.86	1.35
ni	2.94	0.55	1.00	1.42	0.63	0.82	1.27	1.07	1.25	1.83	2.94 +
fi	0.51	0.60	1.02	1.11	0.64	0.73	1.09	1.10	1.07	0.68	1.11
ri	0.45	0.60	0.99	1.14	0.61	0.70	1.13	1.07	1.11	0.69	1.14
mst	1.02	0.56	1.01	1.90	0.65	1.41	1.44	0.91	1.71	1.27	1.90 +
ch	0.79	0.41	1.04	1.96	0.51	0.99	1.41	1.01	1.57	1.04	1.96 +
frp	1.11	0.85	1.00	1.01	0.97	0.97	1.00	0.98	0.99	1.02	1.11
mf-2	0.30	0.43	0.98	1.12	0.49	0.53	1.17	1.06	0.93	0.57	1.17
mf-2h	0.31	0.49	0.96	1.27	0.54	0.55	1.40	1.21	0.99	0.74	1.40
mf-3	0.22	37.43	1.03	3.40	36.45	36.89	11.93	0.97	1.60	28.34	37.43 +

NN=Nearest Neighbourhood, *DENN*=Double Ended NN, *MF*=Multiple Fragment, *NA*, *FA*, *RA*=Nearest, Farthest, Random Addition, *NI*, *FI*, *RI*=Nearest, Farthest, Random Insertion, *MST*=Min. Spanning Three, *CH*=Christofides, *FRP*=Fast Recursive Partition.

Constructive Procedure + Local Search

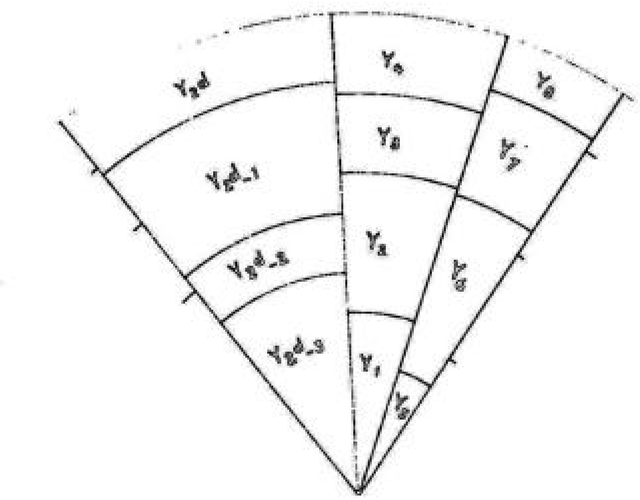
Table IV. Local Optimization Applied to Each Heuristic

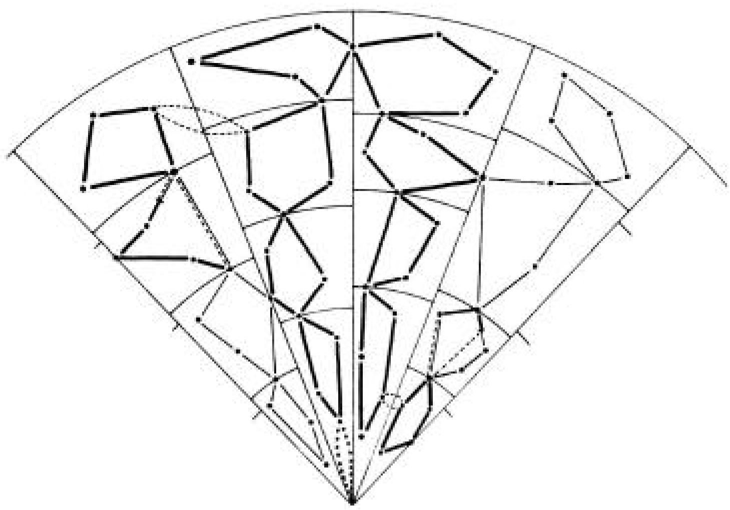
Heuristic Name	Percent Over Lower Bound				CPU Seconds			
	Start	2-Opt	2H-Opt	3-Opt	Start	2-Opt	2H-Opt	3-Opt
nn	24.2	8.7	6.8	4.5	4	27	28	54
denn	24.2	8.6	6.7	4.6	4	28	28	57
mf	15.7	5.8	4.7	3.5	14	30	33	55
na	26.9	16.9	11.2	6.9	26	45	47	83
fa	13.2	11.8	9.6	6.9	38	52	52	78
ra	15.2	12.0	9.8	6.8	16	31	31	56
ni	26.8	16.9	11.3	6.8	46	65	67	101
fi	13.0	11.9	9.7	6.9	76	89	89	112
ri	14.8	12.3	9.9	7.0	57	72	72	97
mst	44.5	12.8	9.3	5.6	16	44	45	80
ch	14.9	6.7	4.9	3.8	24	40	40	60
frp	55.2	14.9	10.5	5.8	2	35	34	73

(Bentley 1992)

Two phases approach

Fisher & Jaikumur (1981)
Taillard (1993)





Fisher & Jaikumur (1981)

1. Choose m seed points j_k in V to initialize each cluster (number of vehicles m is fixed a priori)

2. Compute the cost d_{ik} of allocating each customer i to each cluster k as

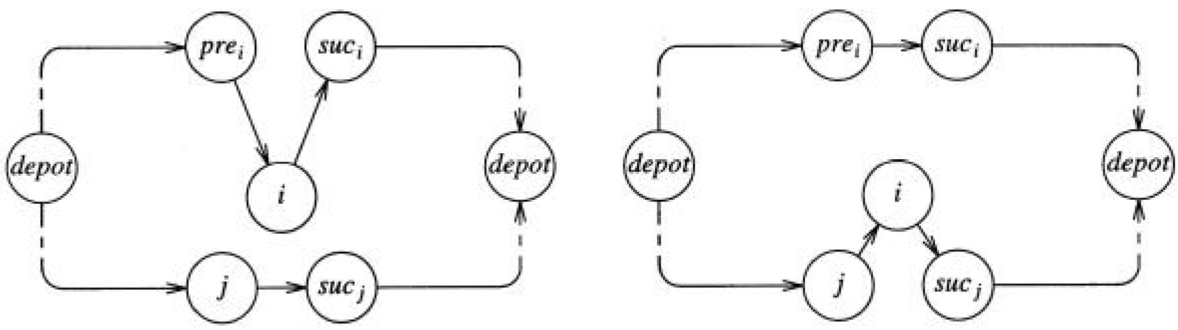
$d_{ijk} = \min \{c_{0i} + c_{ijk} + c_{jk0}, c_{0jk} + c_{jki} + c_{i0}\} - (c_{0jk} + c_{jk0})$

3. Solve an assignment problems with cost d_{ij}, customer weight q_i and vehicle capacity Q.

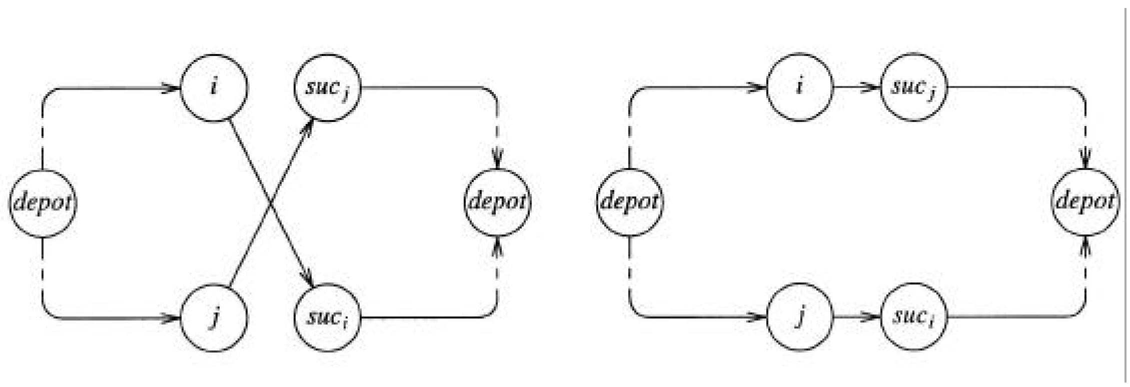
4. Solve a TSP in each cluster corresponding to the assignment solution

Tour Improvement Heuristics
(Kinderwater & Savelsberg 1997)

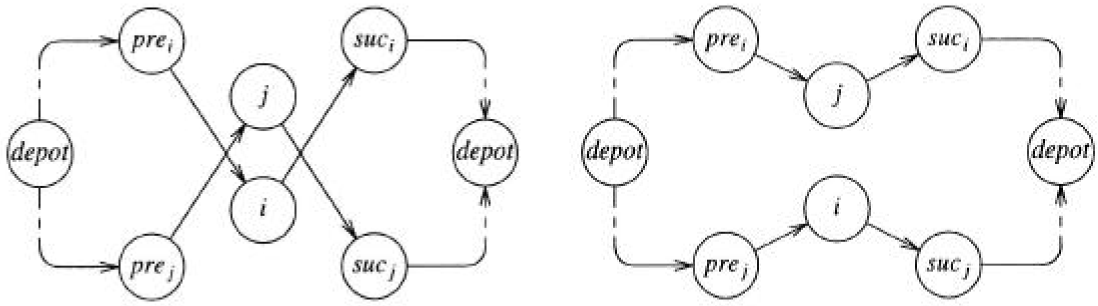
Tours are not considered in isolation. Paths and customers are exchanged between different tours



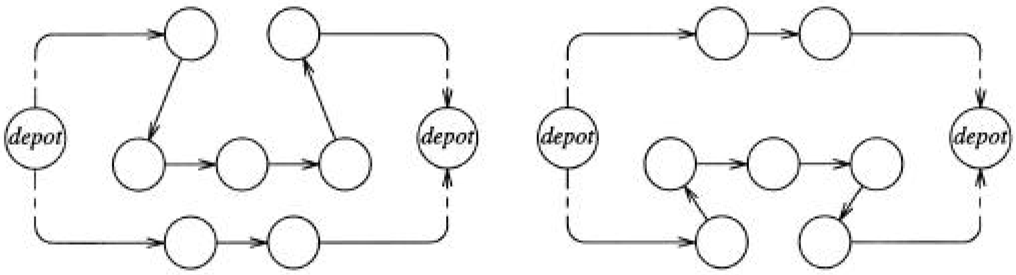
Customer relocation



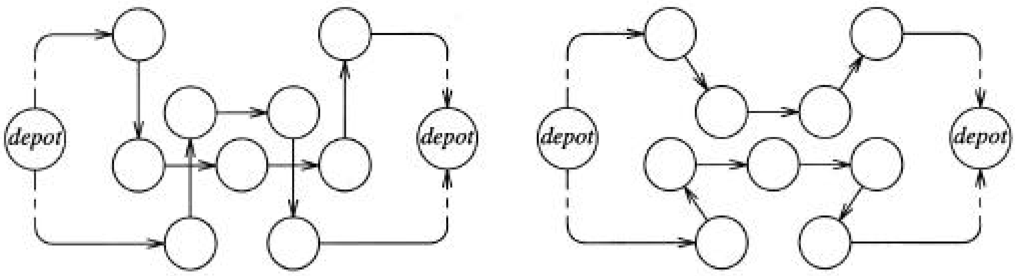
Crossover



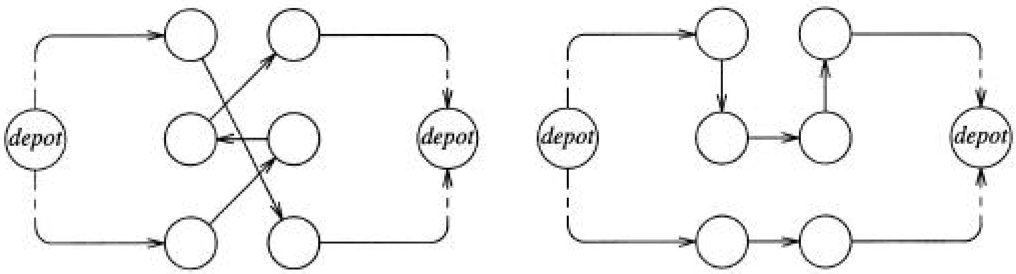
Customer Exchange



(a) Relocation of a path.



(b) Exchange of two paths.



(c) A crossover plus 2-exchange.

Tabu Search with set-partition based Heuristic

(Rochat & Taillard 1995, Kelly & Xu 1999)

1. Keep an adaptive memory as a pool of good solutions
2. Some element (single tour) of these solutions are combined together to form new solution (more weight is given to best solutions)
3. partial solutions are completed by an insertion procedure.
4. Tabu search is applied at the tour level

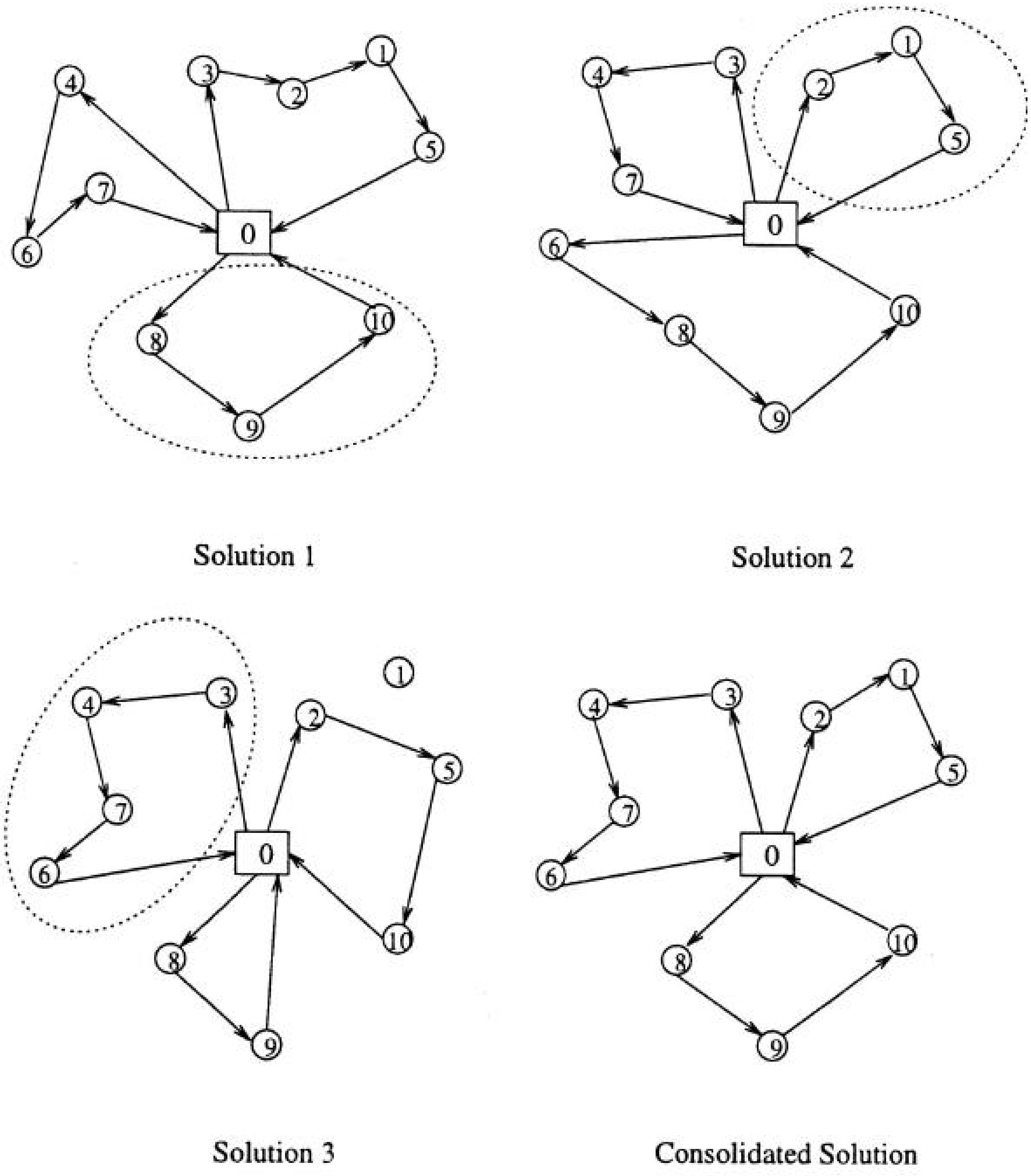


Figure 1. Two-phase procedure.

(Kelly & Xu1999)

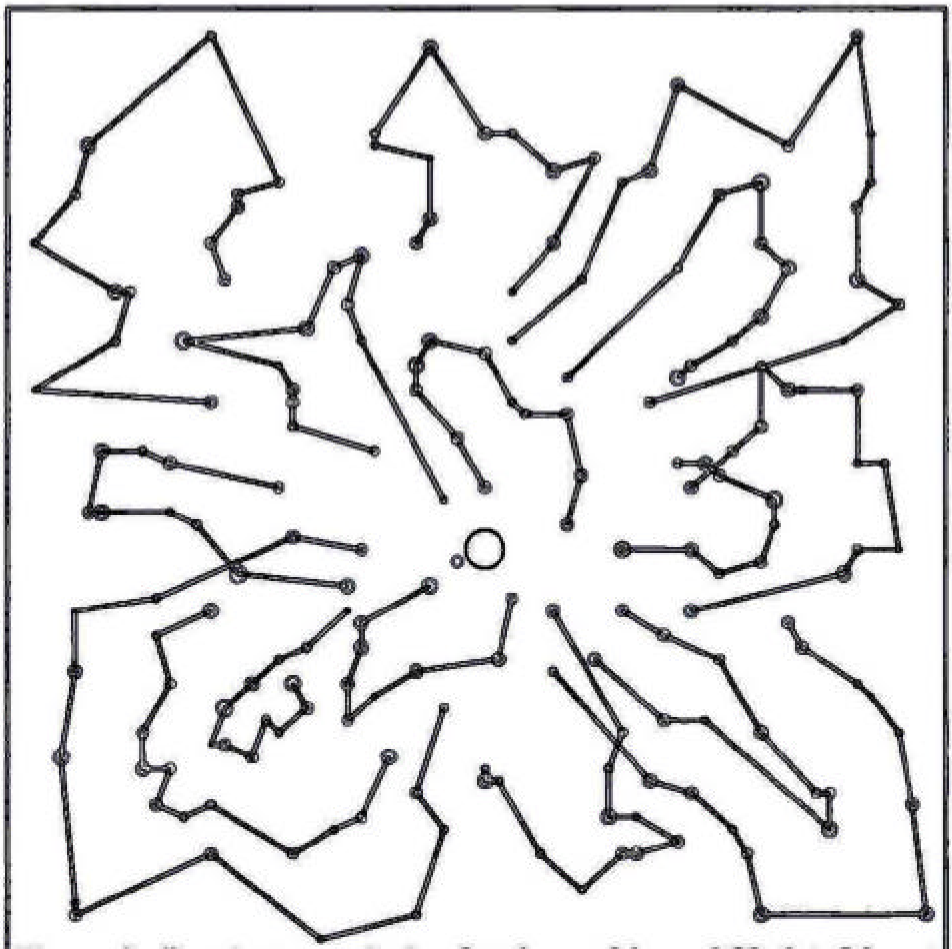


Figure 2 : Best known solution for the problem of Christofides et al. (1979) with 199 customers.

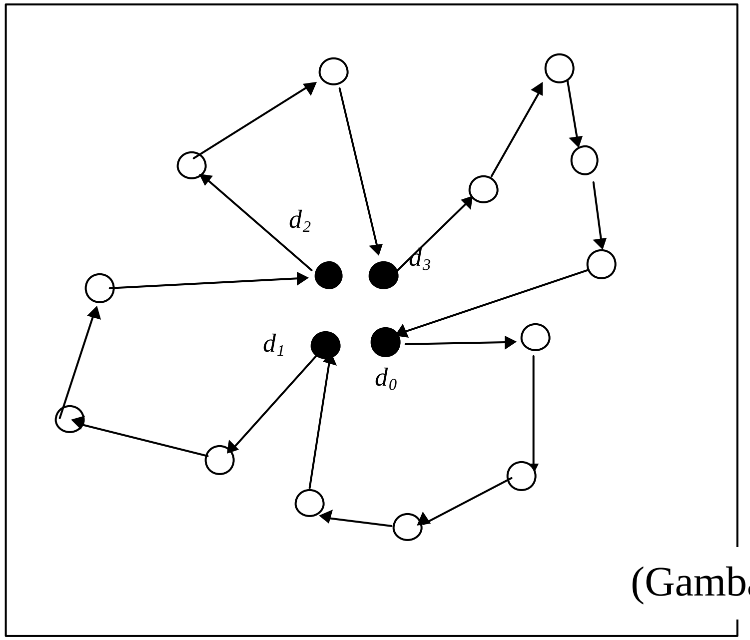
(Rochat & Taillard1995)

Ant Colony System (Gambardella et al. 1999)

VRP is transformed into a TSP by adding *m*-1 new depots

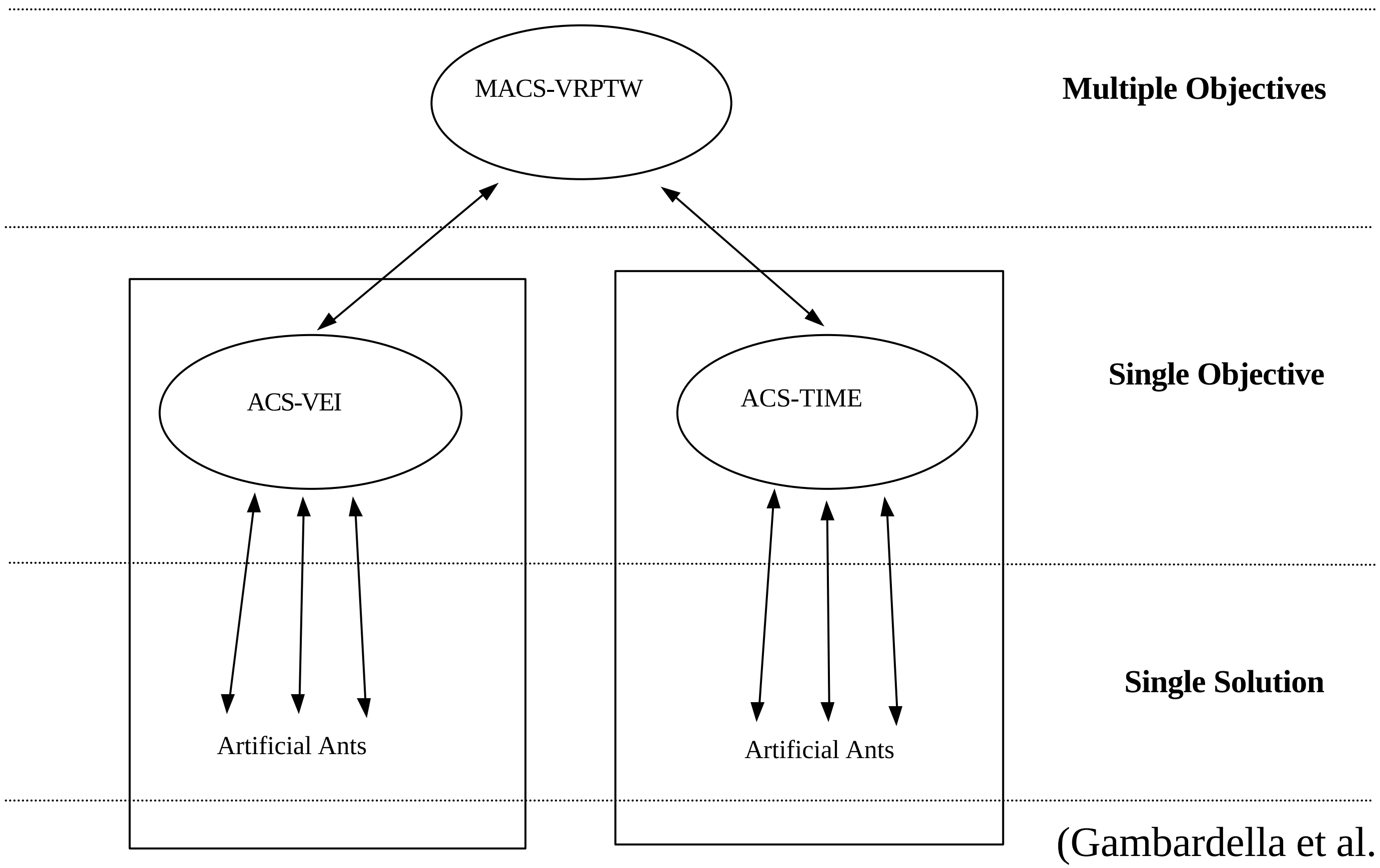
Ants are used to compute complete feasible tours

Local search that exchange paths and customers is executed



(Gambardella et al. 1999)

VRP-TW: in case of vehicle and distance minimization two ant colonies are working in parallel on the two objective functions (colonies exchange pheromone information)



(Gambardella et al. 1999)

VRP with TIME WONDOW: 56 benchmark problems (Solomon, 1987) composed of six different problem types (C1,C2,R1,R2,RC1,RC2).

Each data set contains between eight to twelve 100-node problems.

- Sets C have clustered customers whose time windows were generated based on a known solution.
- Sets R have customers generated uniformly randomly over a square.
- Sets RC have a combination of randomly placed and clustered customers.
- Sets of type 1 have narrow time windows and small vehicle capacity.
- Sets of type 2 have large time windows and large vehicle capacity. Therefore, the solutions of type 2 problems have very few routes and significantly more customers per route.

	R1			C1			RC1			R2			C2			RC2		
	VEI	DIST	TIME	VEI	DIST	TIME	VEI	DIST	TIME	VEI	DIST	TIME	VEI	DIST	TIME	VEI	DIST	TIME
	12.55	1214.80	100	10.00	828.40	100	12.46	1395.47	100	3.05	971.97	100	3.00	593.19	100	3.38	1191.87	100
MACS-	12.45	1212.95	300	10.00	828.38	300	12.13	1389.15	300	3.00	969.09	300	3.00	592.97	300	3.33	1168.34	300
VRPTW	12.38	1213.35	600	10.00	828.38	600	12.08	1380.38	600	3.00	965.37	600	3.00	592.89	600	3.33	1163.08	600
	12.38	1211.64	1200	10.00	828.38	1200	11.96	1385.65	1200	3.00	962.07	1200	3.00	592.04	1200	3.33	1153.63	1200
	12.38	1210.83	1800	10.00	828.38	1800	11.92	1388.13	1800	3.00	960.31	1800	3.00	591.85	1800	3.33	1149.28	1800
	12.83	1208.43	450	10.00	832.59	540	12.75	1381.33	430	3.18	999.63	1600	3.00	595.38	1200	3.62	1207.37	1300
RT	12.58	1202.00	1300	10.00	829.01	1600	12.50	1368.03	1300	3.09	969.29	4900	3.00	590.32	3600	3.62	1155.47	3900
	12.58	1197.42	2700	10.00	828.45	3200	12.33	1269.48	2600	3.09	954.36	9800	3.00	590.32	7200	3.62	1139.79	7800
	12.45	1198.37	900				12.05	1363.67	900									
SW	12.35	1201.47	1800				12.00	1363.68	1800									
	12.33	1201.79	3600				11.95	1364.17	3600									
KPS	12.67	1200.33	2900	10.00	830.75	2900	12.12	1388.15	2900	3	966.56	2900	3.00	592.29	2900	3.38	1133.42	2900
CW	12.50	1241.89	1382	10.00	834.05	649	12.38	1408.87	723	2.91	995.39	1332	3.00	591.78	292	3.38	1139.70	946
	12.64	1233.88	2296	10.00	830.41	2926	12.08	1404.59	1877	3.00	1046.56	3372	3.00	592.75	3275	3.38	1248.34	1933
TB	12.39	1230.48	6887	10.00	828.59	7315	12.00	1387.01	5632	3.00	1029.65	10116	3.00	591.14	8187	3.38	1220.28	5798
	12.33	1220.35	13774	10.00	828.45	14630	11.90	1381.31	11264	3.00	1013.35	20232	3.00	590.91	16375	3.38	1198.63	11596

(Gambardella et al. 1999)

Performance comparison among the best VRPTW algorithms for different computational time (in seconds). RT=Rochat and Taillard (1995), SW = Shaw (1998), KPS = Kilby, Prosser and Shaw (1999), CW = Cordone and Wolfler-Calvo (1998), TB= Taillard et al. (1997)

Average of the best solutions computed by different VRPTW algorithms. Best results are in boldface. RT=Rochat and Taillard (1995), TB= Taillard et al. (1997), CR=Chiang and Russel (1993), PB=Potvin and Bengio (1996), TH= Thangiah et al. (1994)

	R1		C1		RC1		R2		C2		RC2	
	VEI	DIST	VEI	DIST	VEI	DIST	VEI	DIST	VEI	DIST	VEI	DIST
MACS-VRPTW	**12.00**	1217.73	**10.00**	**828.38**	11.63	1382.42	**2.73**	967.75	3.00	**589.86**	**3.25**	1129.19
RT	12.25	1208.50	**10.00**	**828.38**	11.88	1377.39	2.91	961.72	3.00	**589.86**	3.38	1119.59
TB	12.17	1209.35	**10.00**	**828.38**	**11.50**	1389.22	2.82	980.27	3.00	**589.86**	3.38	1117.44
CR	12.42	1289.95	10.00	885.86	12.38	1455.82	2.91	1135.14	3.00	658.88	3.38	1361.14
PB	12.58	1296.80	10.00	838.01	12.13	1446.20	3.00	1117.70	3.00	589.93	3.38	1360.57
TH	12.33	1238.00	10.00	832.00	12.00	1284.00	3.00	1005.00	3.00	650.00	3.38	1229.00

(Gambardella et al. 1999)

	Old Best			New Bes	
Problem	**source**	**vehicles**	**length**	**vehicles**	**length**
r112.dat	RT	1	953.6	**9**	982.14
r201.dat	S	4	1254.0	4	**1253.23**
r202.dat	TB	3	1214.2	3	**1202.52**
r204.dat	S	2	867.3	2	**856.36**
r207.dat	RT	3	814.7	**2**	894.88
r208.dat	RT	2	738.	2	**726.82**
r209.dat	S	3	923.96	3	**921.659**
r210.dat	S	3	963.37	3	**958.241**
rc202.dat	S	4	1162.8	**3**	1377.089
rc203.dat	S	3	1068.07	3	**1062.301**
rc204.dat	S	3	803.9	3	**798.464**
rc207.dat	S	3	1075.25	3	**1068.855**
rc208.dat	RT	3	833.97	3	**833.401**
tai100a.dat	RT	11	2047.90	11	**2041.336**
tai100c.dat	RT	11	1406.86	11	**1406.202**
tai100d.dat	RT	11	1581.25	11	**1581.244**
tai150b.dat	RT	14	2727.77	14	**2656.474**

(Gambardella et al. 1999)

New best solution values computed by MACS-VRPTW.

RT=Rochat and Taillard (1995), S = Shaw (1998), TB= Taillard et al. (1997)

TAIxxxx.dat problems are CVRP problems

Conclusions

VRP is a difficult combinatorial optimization problem that is based on real applications

Different types of VRP problems are available in literature with many different features.

MetaNetwork will investigate the application of different metaheuristics to some of these problems (probably CVRP and VRPTW)

1. P. Badeau, M. Gendreau, F. Guertin, J.-Y. Potvin, É. D. Taillard, A Parallel Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows, *Transportation Research-C* 5, 1997, 109-122.

2. Bentley J.L., “Fast algorithms for geometric traveling salesman problem,” *ORSA Journal on Computing*, vol. 4, pp. 387–411, 1992.

3. W. C. Chiang, R. Russel, Hybrid Heuristics for the Vehicle Routing Problem with Time Windows, Working Paper, Department of Quantitative Methods, University of Tulsa, OK, USA, 1993.

4. Clark G, Wright, J.W., Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research*, 12, 568-581, 1964.

5. Fiala F., Vehicle Routing Problems, GMD-Mitteilungen, 46, Boon, 1978.

6. Fisher M. L., Optimal Solution of Vehicle Routing Problems Using Minimum K-trees, *Operations Research* 42, 1994, 626-642.

7. Fisher M. L., Jaikumur R., A generalized assignment heuristic for vehicle routing, *Network* 11, 109-124, 1981.

8. Flood M. M., The Traveling Salesman Problem, *Operations Research* 4, 1956, 61-75.

9. Gambardella L.M, Taillard E., Agazzi G., MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows , In D. Corne, M. Dorigo and F. Glover, editors, *New Ideas in Optimization*. McGraw-Hill, London, UK, pp. 63-76, 1999.

10. Johnson, D.S., L.A. McGeoch. 1997. The traveling salesman problem: a case study, E. H. Aarts, J. K. Lenstra, eds. *Local Search in Combinatorial Optimization*. John Wiley & Sons, Chichester, UK. 215–310.

11. P. Kilby, P. Prosser, P. Shaw, Guided Local Search for the Vehicle Routing Problems With Time Windows, in *Meta-heuristics: Advances and*

Trends in Local Search for Optimization, S.Voss, S. Martello, I.H. Osman and C.Roucairol (eds.), Kluwer Academic Publishers, Boston, 1999, 473-486.

12. Kindervater, G.A.P., M.W.P. Savelsbergh. 1997. Vehicle routing: handling edge exchanges, E. H. Aarts, J. K. Lenstra, eds. *Local Search in Combinatorial Optimization*. John Wiley & Sons, Chichester, UK. 311–336.

13. Laporte G. Gendreau M., Potvin J-Y., Semet F., Classical amd Modern Heuristics for the vehicle routing problem, *International Transaction in Operational Research*, vol. 7, pp. 285-300, 2000.

14. O. Martin, S.W. Otto, and E.W. Felten, “Large-step Markov chains for the TSP incorporating local search heuristics,” *Operations Research Letters*, vol. 11, pp. 219-224, 1992.

15. Papadimitriou C., Steiglitz K. *Combinatorial optimization : algorithms and complexity* by, Prentice-Hall, New Jersey, 1982

16. J.-Y. Potvin, S. Bengio, The Vehicle Routing Problem with Time Windows - Part II: Genetic Search, *INFORMS Journal of Computing* 8, 1996, 165-172.

17. C. Rego, C. Roucairol, A Parallel Tabu Search Algorithm Using Ejection Chains for the Vehicle Routing Problem, in *Meta-heuristics: Theory and applications*, I.H. Osman, J. Kelly (eds.), Kluwer Academic Publishers, Boston, 1996, 661-675.

18. Y. Rochat, É. D. Taillard, Probabilistic Diversification and Intensification in Local Search for Vehicle Routing, *Journal of Heuristics* 1, 1995, 147-167.

19. M. Solomon, Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints, *Operations Research* 35, 1987, 254-365.

20. P. Shaw, Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems, *Proceedings of the Fourth International*

Conference on Principles and Practice of Constraint Programming (CP '98), M. Maher and J.-F. Puget (eds.), Springer-Verlag, 1998, 417-431.

21. É. D. Taillard, Parallel Iterative Search Methods for Vehicle Routing Problems, *Networks* 23, 1993, 661-673.

22. É. D. Taillard, P. Badeau, M. Gendreau, F. Guertin, J.-Y. Potvin, A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows, *Transportation Science* 31, 1997, 170-186.

23. S. R. Thangiah, I. H. Osman, T. Sun, Hybrid Genetic Algorithm Simulated Annealing and Tabu Search Methods for Vehicle Routing Problem with Time Windows, Technical Report 27, Computer Science Department, Slippery Rock University, 1994.

24. P. Toth, D. Vigo, The Granular Tabu Search (and its Application to the Vehicle Routing Problem), Technical Report, Dipartimento di Elettronica, Informatica e Sistemistica, Università di Bologna, Italy, 1998.

25. J. Xu, J. Kelly, A Network Flow-Based Tabu Search Heuristic for the Vehicle Routing Problem, *Transportation Science* 30, 1996, 379-393.