Local Search methods for Vehicle Routing problems

Michel Gendreau

Département d'informatique et de recherche opérationnelle

and

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

Université de Montréal

University of Bologna

June 5, 2008

PRESENTATION OUTLINE

1. Introduction
2. Neighbourhoods and search spaces
3. Main classes of local search based search methods
4. Tabu Search
5. Recent trends in Tabu and Local Search
6. Local search operators in routing
7. Local search methods for the CVRP
8. Local search methods for the VRPTW
9. References

INTRODUCTION

- "Tough" combinatorial problems have been around for a long time and some have attracted a lot of interest (e.g.: Traveling Salesman Problem)

- Early 70's: complexity theory

$\rightarrow$ NP-hard problems

$\downarrow$

Little hope of solving efficiently many important problems

$\downarrow$

What can be done in practical contexts when solutions are needed?

$\downarrow$

USE HEURISTIC TECHNIQUES

- constructive heuristics (e.g. "greedy")
- iterative improvement methods

CLASSICAL LOCAL IMPROVEMENT HEURISTICS

Key idea:

- In most combinatorial problems, one would expect good solutions to share similar structures.
- Indeed, the best solutions should be obtainable by slightly modifying good ones, and so on…

THUS:

- Start with a (feasible) initial solution.
- Apply a sequence of ***local modifications*** to the current solution as long as these produce improvements in the value of the objective function (monotone evolution of the objective).

These methods are the basic (and earlier) trajectory based search methods.

They are usually called "***local search***" or "***neighbourhood search***" methods.

PROBLEMS AND LIMITATIONS

- These methods stop when they encounter a local optimum (w.r.t. to the allowed modifications).
- Solution quality (and CPU times) depends on the “richness” of the set of transformations considered at each iteration of the heuristic.
- Another key factor is the definition of the set of solutions explored by the algorithm.

THE CLASSICAL VEHICLE ROUTING PROBLEM

Problem data :

- Graph $G = (V, A)$
- Vertices : a depot + customers
- Arcs : possible movements (with travel times)
- A fleet of of m identical vehicles of capacity Q is based at the depot.
- With each customer vertex v_i are associated a demand q_i and a service time t_i.
- With each arc (v_i, v_j) of A are associated a cost c_{ij} and a travel time t_{ij}.

The CVRP consists in finding a set of routes such that:

1. Each route begins and ends at the depot;
2. Each customer is visited exactly once by exactly one route;
3. The total demand of the customers assigned to each route does not exceed Q;
4. The total duration of each route (including travel and service times) does not exceed a specified value L;
5. The total cost of the routes is minimized.

Feasible solution : a partition of the customers into m groups, each of total demand no larger than Q, that are sequenced to yield routes of duration no larger than L.

ANOTHER REFERENCE PROBLEM

THE CAPACITATED PLANT LOCATION PROBLEM

Problem data :

- I = { customers with demands d_i }
- J = { possible location of plants }
- f_j = fixed cost of "opening" the plant at j
- K_j = capacity of plant j
- c_{ij} = unit transportation cost from site j to customer i

Objective :

minimize the total cost
(fixed costs for open plants + transportation costs)

MATHEMATICAL FORMULATION OF THE CPLP

Variables

x_{ij} : quantity shipped from site j to customer i ($i \varepsilon I$, $j \varepsilon J$) (*flow variables*)

y_j : a 0-1 variable indicating the plant at j is open ($j \varepsilon J$) (*location variables*)

(CPLP) Minimize $z = \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$

subject to $\sum_{j \in J} x_{ij} = d_i, \ i \in I$

$$\sum_{i \in I} x_{ij} \le K_j y_j, \ j \in J$$

$$x_{ij} \ge 0, \ i \in I \ , \ j \in J$$

$$y_j \in \{0,1\} \ , \ j \in J$$

PROPERTIES OF THE CPLP (1)

For any vector $\tilde{\boldsymbol{y}}$ of location variables, optimal (w.r.t. to this plant configuration) flow values $\boldsymbol{x}(\tilde{\boldsymbol{y}})$ can be retrieved by solving the associated transportation problem:

$$\text{(TP)} \qquad \text{Minimize } z(\tilde{\boldsymbol{y}}) = \sum_{i\in I} \sum_{j\in J} c_{ij}\, x_{ij}$$

$$\text{subject to} \quad \sum_{j\in J} x_{ij} = d_i,\ i \in I$$

$$\sum_{i\in I} x_{ij} \leq K_j \tilde{y}_j,\ j\in J$$

$$x_{ij} \geq 0,\ i \in I\ ,\ j \in J$$

If $\tilde{\boldsymbol{y}} = \boldsymbol{y}^*$, the optimal solution to the original CPLP problem is given by $(\boldsymbol{y}^*, \boldsymbol{x}(\boldsymbol{y}^*))$.

PROPERTIES OF THE CPLP (2)

An optimal solution of the original CPLP problem can always be found at an extreme point of the polyhedron of feasible flow vectors defined by the constraints:

$$\sum_{j \in J} x_{ij} = d_i,\ i \in I$$

$$\sum_{i \in I} x_{ij} \leq K_j,\ j \in J$$

$$x_{ij} \geq 0,\ i \in I\ ,\ j \in J$$

Reason :

- The CPLP can be interpreted as a fixed-charge problem defined in the space of the flow variables.
- This fixed-charge problem has a concave objective function that always admits an extreme point minimum.

The optimal values for the location variables can easily be obtained from the optimal flow vector by setting y_j equal to 1 if $\sum_{i \in I} x_{ij} > 0,$ and to 0 otherwise.

SEARCH SPACES

AND

NEIGHBOURHOODS

SEARCH SPACES

- Simply the space of all possible solutions that can be considered (visited) during the search.

- Could be the set of all feasible solutions to the problem at hand, with each point in the search space corresponding to a solution satisfying all the specified constraints.

- While this definition of the search space might seem quite natural and straightforward, it is not so in many settings, as we shall see later in a few illustrative examples.

NEIGHBOURHOODS

- At each iteration of LS, the local transformations that can be applied to the current solution, denoted *S*, define a set of neighbouring solutions in the search space, denoted *N(S)* (the neighbourhood of *S*).

- *N(S)* = {solutions obtained by applying a single local modification to *S*}.

- In general, for any specific problem at hand, there are many more possible (and even, attractive) neighbourhood structures than search space definitions.

EXAMPLES OF SEARCH SPACES AND NEIGHBOURHOODS

Two illustrative problems:

- Vehicle routing problem (VRP)
- Capacitated plant location problem (CPLP)

CLASSICAL VEHICLE ROUTING PROBLEM

- $G = (V, A)$, a graph.
- One of the vertices represents the *depot*.
- The other vertices customers that need to be serviced.
- With each customer vertex v_i are associated a demand q_i and a service time t_i.
- With each arc (v_i, v_j) of A are associated a cost c_{ij} and a travel time t_{ij}.
- m identical vehicles of capacity Q are based at the depot.

The CVRP consists in finding a set of routes such that:

- Each route begins and ends at the depot;
- Each customer is visited exactly once by exactly one route;
- The total demand of the customers assigned to each route does not exceed Q;
- The total duration of each route (including travel and service times) does not exceed a specified value L;
- The total cost of the routes is minimized.

SEARCH SPACES AND NEIGHBOURHOODS FOR THE CVRP

Search space:

- Set of feasible routes.
- Allow routes with capacity violations.
- Allow routes with duration violations.

Neighbourhoods:

- Moving a single customer from its route.
- Insertion can be performed simply or in a complex fashion (e.g., GENI insertions).
- Swap customers.
- Simultaneous movement of customers to different routes and swapping of customers between routes (λ-interchange of Osman 1993).
- Coordinated movements of customers from one route to another (ejection chains).
- Swapping of sequences of several customers between routes (Cross-exchange of Taillard *et al.* 1997).

CAPACITATED PLANT LOCATION PROBLEM (CPLP)

- Set of customers I with demands d_i, $i \varepsilon I$.
- Set J of "potential sites" for plants.
- For each site $j \varepsilon J$, the fixed cost of "opening" the plant at j is f_j and its capacity is K_j.
- $c_{ij:}$ cost of transporting one unit of the product from site j to customer i.

The objective is to minimize the total cost, i.e., the sum of the fixed costs for open plants and the transportation costs.

CPLP: MATHEMATICAL FORMULATION

$$\text{(CPLP)} \quad \text{Minimize } z = \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_{j \in J} x_{ij} = d_i,\ i \in I$$

$$\sum_{i \in J} x_{ij} \leq K_j y_j,\ j \in J$$

$$x_{ij} \geq 0,\ i \in I,\ j \in J$$

$$y_j \in \{0,1\},\ j \in J$$

Formulation variables:

- x_{ij} ($i \,\varepsilon\, I, j \,\varepsilon\, J$): quantity shipped from site *j* to customer *i*
- y_j ($j \,\varepsilon\, J$): 0-1 variable indicating whether or not the plant at site *j* is open or closed.

Remark 1. For any vector $\tilde{\boldsymbol{y}}$ of location variables, optimal (w.r.t. to this plant configuration) values for the flow variables $\boldsymbol{x}(\tilde{\boldsymbol{y}})$ can be retrieved by solving the associated transportation problem:

$$\text{(TP)} \quad \text{Minimize } z(\tilde{\boldsymbol{y}}) = \sum_{i \in I} \sum_{j \in J} c_{ij}\, x_{ij}$$

$$\text{subject to } \sum_{j \in J} x_{ij} = d_i,\ i \in I$$

$$\sum_{i \in J} x_{ij} \le K_j\, \tilde{y}_j,\ j \in J$$

$$x_{ij} \ge 0,\ i \in I,\ j \in J$$

If $\tilde{\boldsymbol{y}} = \boldsymbol{y}^*$, the optimal location vector, the optimal solution to the original CPLP problem is simply given by $(\boldsymbol{y}^*, \boldsymbol{x}(\boldsymbol{y}^*))$.

Remark 2. An optimal solution of the original CPLP problem can always be found at an extreme point of the polyhedron of feasible flow vectors defined by the constraints:

$$\sum_{j \in J} x_{ij} = d_i,\ i \in I$$

$$\sum_{i \in J} x_{ij} \le K_j,\ j \in J$$

$$x_{ij} \ge 0,\ i \in I\ ,\ j \in J$$

This property follows from the fact that the CPLP can be interpreted as a fixed-charge problem defined in the space of the flow variables. This fixed-charge problem has a concave objective function that always admits an extreme point minimum. The optimal values for the location variables can easily be obtained from the optimal flow vector by setting y_j equal to 1 if $\sum_{i \in I} x_{ij} > 0,$ and to 0 otherwise.

SEARCH SPACES AND NEIGHBOURHOODS FOR THE CPLP

Search space:

1) Full feasible space defined by all variables.
2) Space defined by location variables.
3) Set of extreme points of the set of feasible flow vectors.

Neighbourhoods:

- Depend upon the search space chosen.
- For 2), one can use "Add/Drop" and/or "Swap" neighbourhoods.
- For 3), moves defined by the application of pivots to the linear programming formulation of the transportation problem, since each pivot operation moves the current solution to an adjacent extreme point.

A TEMPLATE FOR LOCAL SEARCH

To maximize $f(S)$ over some domain

Define: S, current solution,

f^*, value of the best-known solution,

S^*, this solution,

$N(S)$, the "neigbourhood" of S (solutions obtained from S by a single transformation).

Initialization

Choose (construct) an initial solution S_0

Set $S := S_0$, $f^* := f(S_0)$, $S^* := S_0$.

Search

While local optimum not reached *do*

- $S :\in \arg\max_{S' \in \overline{N}(S)} [f(S')]$;
- *if* $f(S) \rangle f^*$, *then* $f^* := f(S), S^* := S$.

MAIN CLASSES OF LOCAL SEARCH METHODS

Simple Local Search

- The simplest of all LS approaches
- Consists in constructing a single initial solution and improving it using a single neighbourhood structure until a local optimum is encountered.
- Two variants of simple LS:
  - "Best improvement"
  - "First improvement"

Multi-start Local Search

- A simple extension to the simple LS scheme
- Several (usually randomly generated) initial solutions
- Apply to each of them this simple scheme, thus obtaining several local optima from which the best is selected and returned as the heuristic solution.

SIMULATED ANNEALING

- Kirkpatrick, Gelatt and Vecchi (1983)
- Based on an analogy with the cooling of material in a heat bath.
- Metropolis' algorithm (1953)
- Solutions <—> Configurations of particles
- Objective function <—> Energy of system
- Can be interpreted as a controlled random walk in the space of solutions:
  - Improving moves are always accepted;
  - Deteriorating moves are accepted with a probability that depends on the amount of the deterioration and on the *temperature* (a parameter that decreases with time).
- Extensions/generalizations: deterministic annealing, threshold acceptance methods.
- Local search methods in which deterioration of the objective up to a *threshold* is accepted.
- As in SA, the threshold decreases as the algorithm progresses.

VARIABLE NEIGHBOURHOOD SEARCH

- Introduced, by Hansen and Mladenović in 1997.
- Use, instead of a single neighbourhood, several of these in pre-defined sequences.
- Over time VNS has yielded several variants of different complexity.
- The simplest one, called Variable Neighbourhood (VND), is clearly the multi-neighbourhood extension of LS.
- In VND, one first performs LS using the first neighbourhood structure until a local optimum is encountered; the search is then continued using the second neighbourhood structure until a local optimum (w.r.t. to that structure) is encountered, at which point, it switches to the third neighbourhood structure, and so on in a circular fashion.
- VND will eventually stop, but only in a point which is a local optimum for each of the considered neighbourhood structures.

THE TABU SEARCH APPROACH

- Glover (1977, 1986)

- Hansen (1986: *steepest ascent/mildest descent*)

- A metaheuristic that controls an **inner** heuristic designed for the specific problem that is to be solved.

- Artificial intelligence concepts: maintain a history of the search in a number of **memories**.

- **Basic principle**: allow non-improving moves to overcome local optimal (i.e. keep on transforming the current solution...).

- **PROBLEM**: How can **CYCLING** be avoided???

 SOLUTION: Keep a **HISTORY** of the searching process and prohibit «comebacks» to previous solutions (tabu moves).

TABUS

- A short-term memory of the search (in general, only a fixed amount of information is recorded).

- Several possibilities:

  - a list of the last solutions encountered (expensive, and not frequently used);

  - a list of the last modifications performed on current solutions; reverse modifications are then prohibited
    (the most common type of tabus);

  - a list of key characteristics of the solutions or of the transformations
    (sometimes more efficient)

EXAMPLES OF TABUS

Consider the situation where one is solving the TSP with 2-opt as inner heuristic.

The basic set of transformations at each step consists of moves obtained by removing two edges $[(i,j),(k,\ell)]$; and replacing them with edges $[(i,k),(j,\ell)]$.

Possible tabus

- Forbid tours themselves.
- Forbid **reverse transformations** $[(i,k),(j,\ell)] \to [(i,j),(k,\ell)]$ for a few iterations.
- Forbid any transformation involving either (i,k) or (j,ℓ) for some time.
- ...

MORE ON TABUS

- **Multiple tabu lists** can be used and have proved quite useful in many contexts.

- "Straightforward" tabus can be implemented as circular lists of fixed length.

- Fixed-length tabus cannot always prevent cycling: many authors have proposed schemes to vary tabu list length during execution (Skorin-Kapov, Taillard).

- Another solution: **random tabu tags**, the duration of a tabu status is a random variable generated when the tabu is created.

- Yet another solution: **randomly activated tabus**, at each iteration, a random number is generated indicating how far to look back in the tabu list (which is otherwise managed like a fixed-length list).

ASPIRATION CRITERIA

- Tabus are sometimes too “powerful”:

  - attractive moves are prohibited, even when there is no danger of cycling;

  - they can lead to overall stagnation of the searching process.

- Aspiration criteria are algorithmic devices that ***cancel tabus*** in some circumstances.

- The simplest aspiration criterion consists in allowing a move if it results in a solution with objective value better than that of the best-known solution.

- Much more complicated criteria have been proposed and implemented in some applications.

KEY RULE : If cycling cannot occur, you may disregard tabus

SIMPLE TABU SEARCH

To maximize $f(S)$ over some domain

Define: S, current solution,

f^*, value of the best-known solution,

S^*, this solution,

T, the tabu list,

$N(S)$, the "neigbourhood" of S (solutions obtained from S by a single transformation),

$\overline{N}(S)$, "admissible" subset of $N(S)$ (non-tabu or allowed by aspiration).

Initialization

Choose (construct) an initial solution S_0

Set $S := S_0$, $f^* := f(S_0)$, $S^* := S_0$, $T := \varnothing$

Search

While termination criterion not satisfied *do*

- $S :\in \arg\max_{S' \in \overline{N}(S)} [f(S')]$;
- *if* $f(S) \rangle f^*$, *then* $f^* := f(S)$, $S^* := S$;
- *record tabu for the current move in T (delete oldest tabu if necessary).*

TERMINATION CRITERIA

- In theory, the search could go on for ever (unless the optimal value of the problem is known beforehand).

- In practice, the search has to be stopped at some point:

  - after a fixed number of iterations (or a fixed amount of CPU time),

  - after some number of iterations without an improvement in the best objective value (probably the most commonly used criterion),

  - when the objective reaches a pre-specified threshold value.

- In complex tabu search schemes, the search will usually be stopped after completing a sequence of **phases**, the duration of each phase being determined by one of the above criteria.

PROBABILISTIC TABU SEARCH

In "regular" simple tabu search, one must evaluate the objective for every element in the neighbourhood $N(S)$ of the current solution.

Instead of considering the whole set $N(S)$, one may restrict its attention to a random sample $N'(S) \subset N(S)$.

Advantages :

- In most applications, a smaller computational effort, since one only evaluates the objective for $S' \in N'(S)$;
- The random choice of $N'(S)$ acts as an anti-cycling choice
  $\rightarrow$ shorter tabu lists can be used.

Disadvantage : the best solution may be missed.

SEARCH INTENSIFICATION

Idea : *To explore more thoroughly portions of the search space that seem "promising"*

- From times to times, the normal searching process is stopped and an intensification phase is executed.

- Often based on some kind of **intermediate-term memory**
  → ***recency memory*** records the number of iterations that "elements" have been present in the current solution.

- Often restarted from the best-known solution.

- Possible techniques:

  - "freezing" (fixing) "good" elements in the current solution;
  - changing (increasing) sample size in probabilistic TS;
  - switching to a different inner heuristic or modifying the parameters driving it.

SEARCH DIVERSIFICATION

- In many cases, the normal searching process tends to spend most of its time in a restricted portion of the search space. Good solutions may be obtained, but one may still be far from the optimum.

  Diversification : *a mechanism to “force” the search into previously unexplored areas.*

- Usually based on some form of **long-term memory** .
  → ***frequency memory*** records the number of times each “element” has appeared in the solution.

- Most common techniques:

  - **restart diversification** : force a few “unfrequent” elements in the solution and restart the search from the new current solution thus obtained;

  - **continuous diversification** : in the evaluation of moves, **bias** the objective by adding a small term related to element frequencies;

  - strategic oscillation : (see next transparency).

HANDLING CONSTRAINTS

- In many instances, accounting for all problem constraints in the definition of the search space severely restricts the search process and leads to mediocre solutions.

  $\rightarrow$ ***constraint relaxation is often effective!***

- "Wider" search space which is often easier to handle
  $\rightarrow$ simpler neighbourhoods can be used.

- Constraint violations are added to the objective as a weighted penalty term.

- But, how can one find "good" weights?

  $\rightarrow$ **self-adjusting penalties** can be used

  - weights are adjusted dynamically based on the recent history of the search

    + increase weights when only infeasible solutions are encountered,

    + decrease weights if the opposite occurs.

Strategic oscillation : changing weights to induce diversification.

SURROGATE AND AUXILIARY OBJECTIVES

- In some problems, the true objective function is extremely costly to evaluate (e.g., MIP, with the search space restricted to integer variables; stochastic programming;...).

  $\rightarrow$ The evaluation of moves becomes prohibitive (even if sampling is used).

- Solution: evaluate neighbours using a surrogate objective function

  - correlated to the true objective,
  - less demanding computationally,
  - the value of the true objective is computed only for the chosen move or for a subset of promising candidates.

- In some problems, most neighbours have the same objective value. How can one choose the next move among them?

  By using an auxiliary objective function measuring a desirable attribute of solutions.

RECENT TRENDS IN TABU SEARCH (AND OTHER LOCAL SEARCH APPROACHES)

PARALLEL VARIANTS

Parallel processing opens up great opportunities for new developments in tabu search.

- **Low-level parallelization**

  Using parallel processing to speed up computationally demanding steps of “standard” tabu search.

- **High-level parallelization**

  Run several search threads in parallel to obtain more information and come up with better solutions

  (parallel search threads can also be used on sequential architectures).

These techniques have already been used with very good results.

Taxonomy paper by Crainic, Toulouse and Gendreau (1997).

Book edited by E. Alba (2005).

HYBRIDS

Using local or tabu search in combination with other optimization techniques.

- In branch-and-bound, to compute bounds.
- In conjunction with genetic algorithms or ant colony optimization.
- Alternately with other LS or TS methods.
- In conjunction with Constraint Logic Programming techniques.

Currently, the most successful methods.

Two general schemes:

- “unified” architectures (a single algorithm combining components of several methods),
- “parallel hybrids” (running concurrently “pure” implementations of two or more algorithms).

USING INFORMATION IN A DIFFERENT WAY

- **Reactive Tabu Search**
  - Battiti and Tecchiolli (1992, 1994)

- **Path relinking, Scatter search**
  - Glover (1994, 1995)
  - Glover and Laguna (1997)

- **Candidate list and elite solutions**
  - see Glover and Laguna (1997)

- **Hashing and Chunking**
  - Woodruff and Zemel (1993)
  - Carlton and Barnes (1995)
  - Woodruff (1996)

- **Vocabulary building**
  - Glover (1992)
  - Glover and Laguna (1993)
  - Rochat and Taillard (1995)
  - Kelly and Xu (1995)
  - Lopez, Carter and Gendreau (1998)

NEW APPLICATION AREAS

- Integer and mixed-integer programming
- Continuous optimization problems
  - with extreme point solutions
    - + concave programming
    - + fixed-charge problems
  - with “general” solution structure
- Continuous, multi-criteria optimization
- Stochastic programming problems

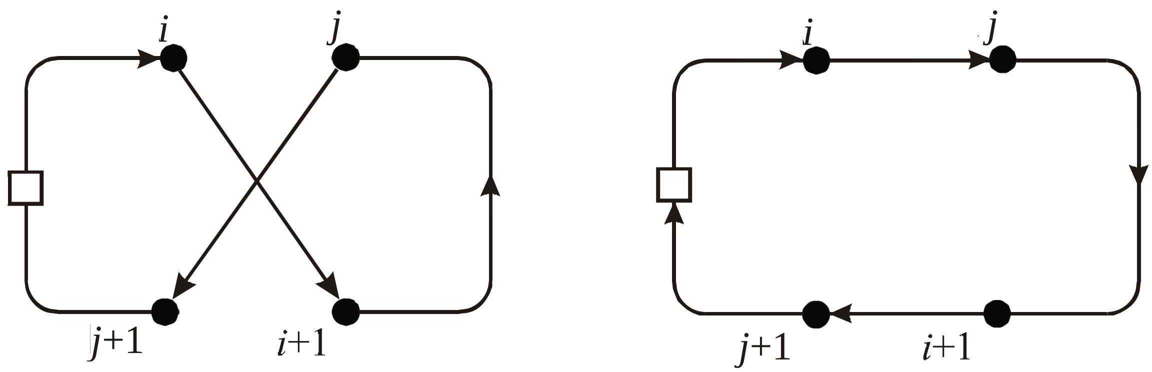
  especially those with a large number of possible realizations (intractable using standard approaches)
- Real-time decision problems
  - LS methods almost possess the “Anytime” property;
  - Solutions can often be adjusted in real time to new information.

IN-DEPTH PERFORMANCE ANALYSIS

- New area launched about 5 years ago by Jean-Paul Watson and his co-authors.

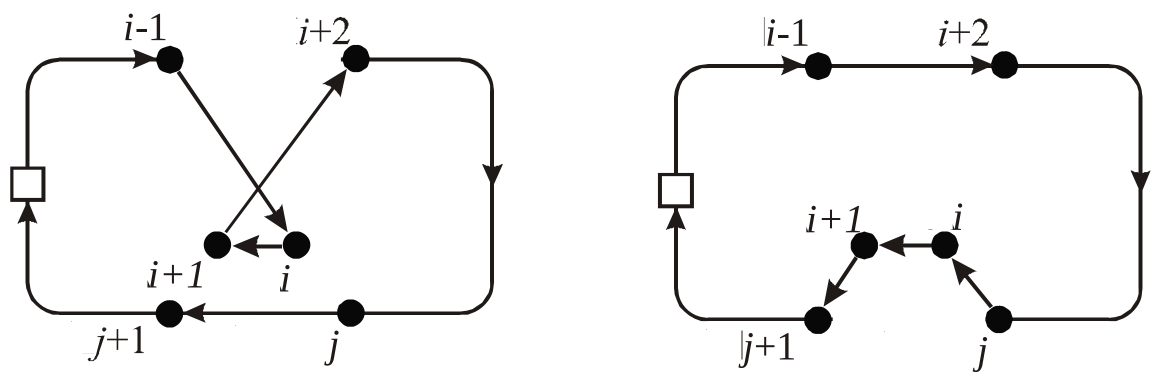
- The focus is not on developing new methods, but in modelling and understanding the behaviour of existing methods.

LOCAL SEARCH OPERATORS IN ROUTING



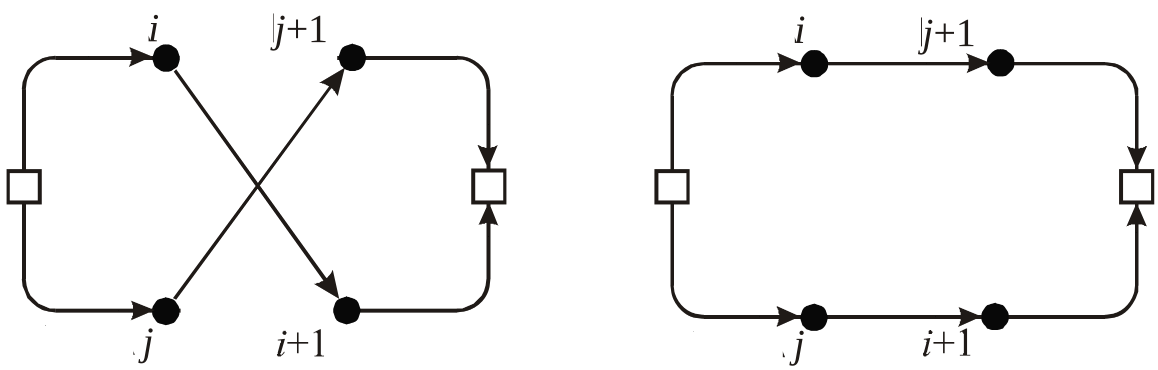
2-opt exchange operator

Edges (i, i+1) and (j, j+1) are replaced by edges (i, j) and (i+1, j+1), thus reversing the direction of customers between i+1 and j.



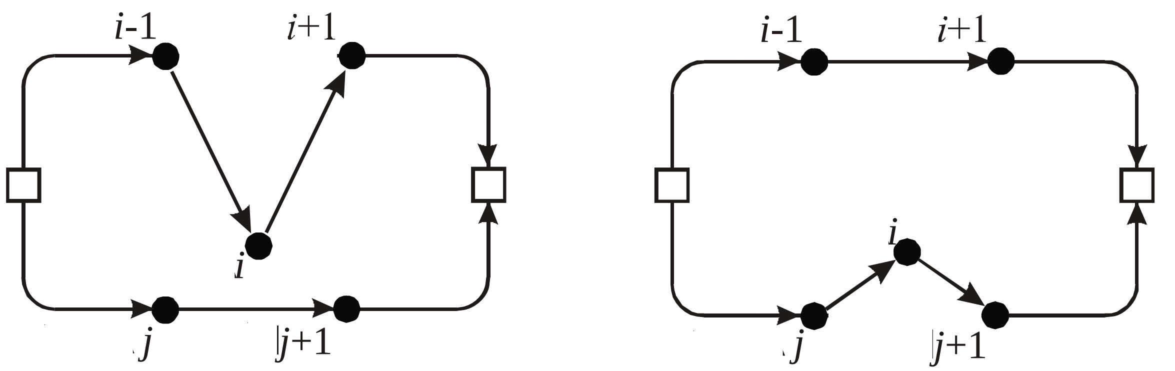
Or-opt operator

Customers *i* and *i*+1 are relocated to be served between two customers *j* and *j*+1 instead of customers *i*-1 and *i*+2. This is performed by replacing 3 edges (*i*-1, *i*), (*i*+*1*, *i*+2) and (*j*, j+1) by the edges (*i*-1, *i*+2), (*j*, *i*) and (*i*+*1*, *j*+1), preserving the orientation of the route.



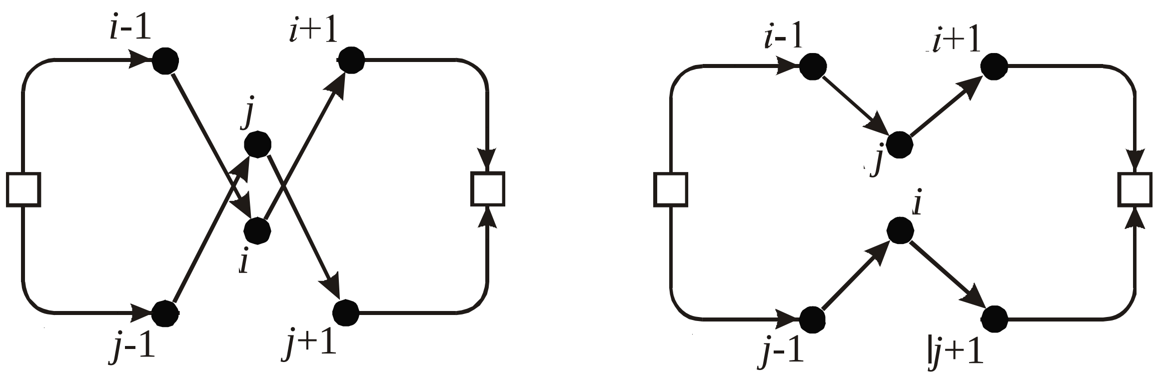
2-opt* operator

The customers served after customer *i* on the upper route are reinserted to be served after customer *j* on the lower route and customers after *j* on the lower route are moved to be served on the upper route after customer *i*. This is performed by replacing edges (*i*, *i*+1) and (*j*, *j*+1) with edges (*i*, *j*+1) and (*j*, *i*+1).



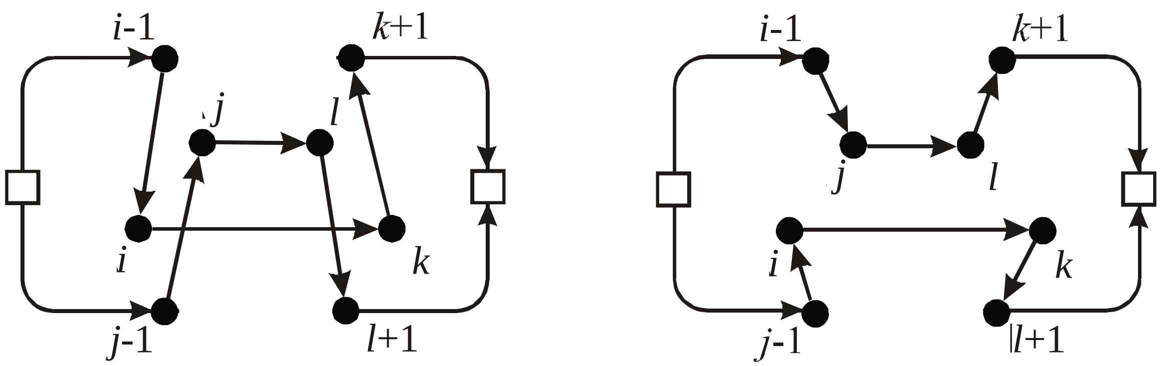
Relocate operator

Edges (i-1, i), (i, i+1) and (j, j+1) are replaced by (i-1, i+1), (j, i) and (i, j+1), i.e., customer i from the origin route is placed into the destination route.



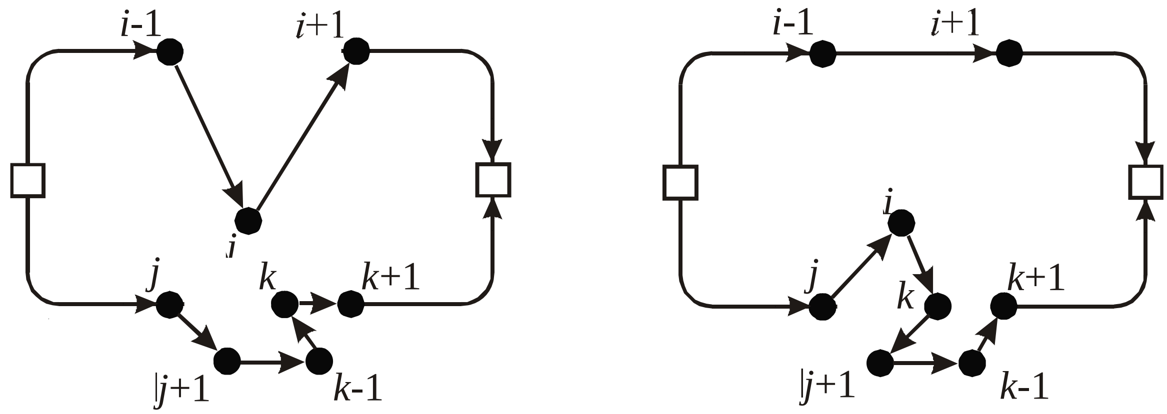
Exchange operator

Edges (*i*-1, *i*), (*i*, *i*+1), (*j*-1, *j*) and (*j*, *j*+1) are replaced by (*i*-1, *j*), (*j*, *i*+1), (*j*-1, *i*) and (*i*, *j*+1), i.e., two customers from different routes are simultaneously placed into the other routes.



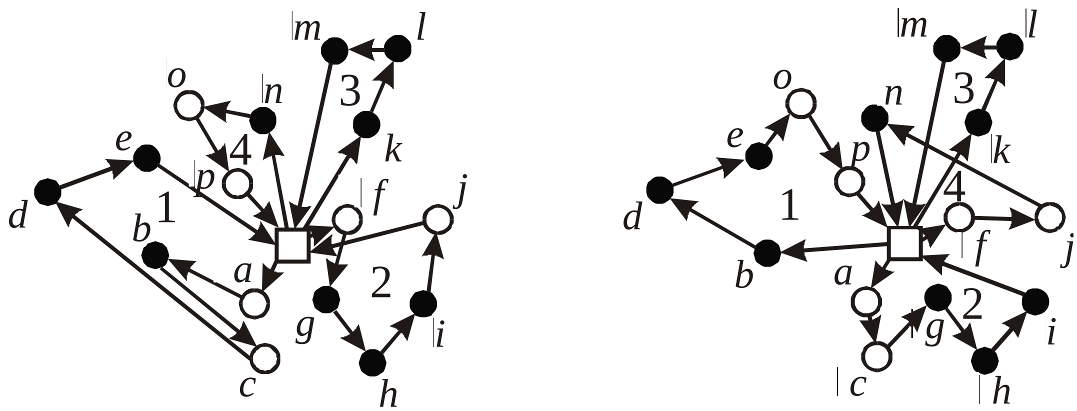
CROSS-exchange

Segments (i, k) on the upper route and (j, l) on the lower route are simultaneously reinserted into the lower and upper routes, respectively. This is performed by replacing edges (i-1, i), (k, k+1), (j-1, j) and (l, l+1) by edges (i-1, j), (l, k+1), (j-1, i) and (k, l+1). Note that the orientation of both routes is preserved.



GENI-exchange operator

Customer *i* on the upper route is inserted into the lower route between the customers *j* and *k* closest to it by adding the edges (*j*, *i*) and (*i*, *k*). Since *j* and *k* are not consecutive, one has to reorder the lower route. Here the feasible tour is obtained by deleting edges (*j*, *j*+1) and (*k*-1, *k*) and by relocating the path {*j*+1,…, *k*-1}.



Cyclic transfer operator

The basic idea is to transfer simultaneously the customers denoted by white circles in cyclical manner between the routes. More precisely here customers *a* and *c* in route 1, *f* and *j* in route 2 and *o* and *p* in route 4 are simultaneously transferred to routes 2, 4, and 1 respectively, and route 3 remains untouched.

LOCAL SEARCH METHODS FOR THE VRP

EARLY METHODS

SIMULATED ANNEALING

- Robusté et al. (1990):
  - Complex neighbourhood (swap + Or-opt + …)
  - Only tested on four instances

- Alfa et al. (1991):
  - Route-first, cluster second heuristic for the initial solution
  - 3-opt neighbourhood
  - Not competitive

- Osman (1993):
  - λ-interchange neighbourhood (includes swaps and relocate for subsets of size $\leq \lambda$)
  - Special cooling schedule
  - Generally produces good, but not exceptional results

- Van Breedam (1995):
  - Tested several variants of SA
  - Could not match results produced with Tabu Search

EARLY METHODS (2)

TABU SEARCH

- Willard (1989):
  - Solution represented as a giant tour by replication of the depot
  - Neighbourhood based on 2- and 3-opt moves
  - Does not seem competitive

- Pureza and França (1991):
  - (Relocate + swap) neighbourhood
  - Preserve feasibility
  - Did not produce exceptionally good results

MEDIEVAL METHODS

TABU SEARCH

- Osman (1993):
  - λ-interchange neighbourhood with $\lambda = 2$
  - Variants with “best accept” and “first improvement” rules
  - Generally produces excellent, but not the best results

- Taillard (1993):
  - λ-interchange neighbourhood
  - Feasible solutions
  - Decomposition into smaller subproblems that are modified during the execution of the algorithm
  - Suitable for parallel implementations
  - Continuous diversification
  - Excellent computational results, but unknown CPU times

MEDIEVAL METHODS (2)

TABU SEARCH

- TABUROUTE (Gendreau, Hertz, Laporte 1994)
  - GENI neighbourhoods
  - Moves in infeasible space with self-adjusting penalties
  - Continuous diversification
  - Random tabu tags
  - Excellent computational results

- Rochat and Taillard (1995):
  - Introduces the concept of ***adaptive memory*** (pool of elite solutions used to reconstruct solutions for intensification/diversification purposes)
  - Outstanding computational results on both the VRP and the VRPTW

- Rego and Roucairol (1996)
  - Based on ejection chains (cyclic transfer neighbourhood)
  - Parallel implementation
  - Generally produces excellent, but not the best results

MEDIEVAL METHODS (3)

DETERMINISTIC ANNEALING

- Golden et al. (1998):
  - Applied record-to-record travel to 20 large instances
  - Produces better results than Xu and Kelly's tabu search heuristic for 11 instances out of 20
  - Much faster than Xu and Kelly's heuristic

RECENT METHODS

TABU SEARCH

- Granular Tabu Search (Toth and Vigo 2003)
  - Removes from the graph long edges unlikely to belong to the optimal solution
  - Typically keep between 10 to 20% of the original edges
  - The *sparsification parameter* can be adjusted dynamically to yield intensification or diversification
  - Edge-exchange neighbourhood
  - Excellent results (see tables later)

- Unified Tabu Search (Cordeau et al. 1997, 2001, 2004)
  - Similar in many ways to TABUROUTE
  - A single initial solution is considered
  - Additional diversification is used by moving the depot arbitrarily at some points
  - Can be applied to many variants of the VRP
  - Excellent computational results

RECENT METHODS

DETERMINISTIC ANNEALING

- Li et al. (2004):
  - Combines record-to-record principles with a variable-length neighbour list whose principle is similar to Granular Tabu Search
  - Neighbourhood based on intra-route and inter-route 2-opt moves
  - Excellent results

VERY LARGE NEIGHBORHOOD SEARCH

- Ergun et al. (2003):
  - Descent mechanism
  - The method considers at each iteration a composite neighbourhood involving changes to several routes as in ejection chains or the cyclic transfer neighbourhood
  - Changes to individual routes are based on 2-opt, swap and relocate moves.
  - The set f moves to be performed at each iteration is obtained by solving a shortest path problem.
  - Excellent results

Table 1.1. Computational results for the Christofides et al. (1979) instances

			GTS Toth and Vigo (2003)			Li, Golden and Wasil (2004)		USTA Cordeau et al. (2001)			VLNS Ergun et al. (2003)			Prins (2004)		
Instance	n	Type[1]	Value	%	Minutes[2]	Value[3]	%	Value[4]	%	Minutes[5]	Value[6]	%	Minutes[7]	Value	%	Minutes[8]
1	50	C	**524.61**	0.00	0.81	**524.61**	0.00	**524.61**	0.00	2.32	**524.61**	0.00	23.13	**524.61**	0.00	0.01
2	75	C	838.60	0.40	2.21	836.18	0.11	835.28	0.00	14.78	835.43	0.02	33.93	**835.26**	0.00	0.77
3	100	C	828.56	0.29	2.39	827.39	0.15	**826.14**	0.00	11.67	827.46	0.16	21.30	**826.14**	0.00	0.46
4	150	C	1033.21	0.47	4.51	1045.36	1.65	1032.68	0.41	26.66	1036.24	0.76	24.45	1031.63	0.31	5.50
5	199	C	1318.25	2.09	7.50	1303.47	0.94	1315.76	1.90	57.68	1307.33	1.24	57.25	1300.23	0.69	19.10
6	50	C, D	**555.43**	0.00	0.86			**555.43**	0.00	3.03	**555.43**	0.00	3.50	**555.43**	0.00	0.01
7	75	C, D	920.72	1.21	2.75			**909.68**	0.00	7.41	910.04	0.04	36.53	912.30	0.29	1.42
8	100	C, D	869.48	0.41	2.90			865.95	0.00	10.93	**865.94**	0.00	12.43	**865.94**	0.00	0.37
9	150	C, D	1173.12	0.91	5.67			1167.85	0.46	51.66	1164.88	0.20	42.47	1164.25	0.15	7.25
10	199	C, D	1435.74	2.86	9.11			1416.84	1.50	106.28	1404.36	0.61	28.32	1420.20	1.74	26.83
11	120	C	1042.87	0.07	3.18	**1042.11**	0.00	1073.47	3.01	11.67	**1042.11**	0.00	69.13	**1042.11**	0.00	0.30
12	100	C	**819.56**	0.00	1.10	**819.56**	0.00	**819.56**	0.00	9.02	**819.56**	0.00	5.98	**819.56**	0.00	0.05
13	120	C, D	1545.51	0.28	9.34			1549.25	0.53	21.00	1544.99	0.25	39.73	1542.97	0.12	10.44
14	100	C, D	**866.37**	0.00	1.41			**866.37**	0.00	10.53	**866.37**	0.00	6.55	**866.37**	0.00	0.09
Average				0.64	3.84		0.41		0.56	24.62		0.23	28.91		0.24	5.19

1. C: Capacity restrictions; D: Route length restrictions.
2. Pentium (200 MHz).
3. Best variant ($\alpha = 0.4$) .
4. Results of recent computational experiments (see Section 3.3); the average % deviation in Cordeau et al. (2001) is 0.69.
5. Pentium IV (2GHz).
6. Best of five runs.
7. Time for reaching the best value for the first time (Pentium III, 733 MHz).
8. GHz PC (75 MFlops).

Table 1.1. (continued). Computational results for the Christofides et al. (1979) instances

			Bone Route (Tarantitis and Kiranoudis (2002))			AGES best Mester and Bräysy (2004)			AGES fast Mester and Bräysy (2004)			Berger and Barkaoui (2004)			
Instance	n	Type[1]	Value	%	Minutes[9]	Value[10]	%	Minutes[11]	Value[10]	%	Minutes[11]	Value	%	Minutes[12]	Best
1	50	C	**524.61**	0.00	0.11	**524.61**	0.00	0.01	**524.61**	0.00	0.01	**524.61**	0.00	2.00	**524.61**
2	75	C	**835.26**	0.00	4.56	**835.26**	0.00	0.26	**835.26**	0.00	0.26	**835.26**	0.00	14.33	**835.26**
3	100	C	**826.14**	0.00	7.66	**826.14**	0.00	0.05	**826.14**	0.00	0.05	827.39	0.15	27.90	**826.14**
4	150	C	1030.88	0.24	9.13	**1028.42**	0.00	0.47	**1028.42**	0.00	0.47	1036.16	0.75	48.98	**1028.42**
5	199	C	1314.11	1.77	16.97	**1291.29**	0.00	101.93	1294.25	0.23	0.50	1324.06	2.54	55.41	**1291.29**
6	50	C, D	**555.43**	0.00	0.10	**555.43**	0.00	0.02	**555.43**	0.00	0.02	**555.43**	0.00	2.33	**555.43**
7	75	C, D	**909.68**	0.00	0.92	**909.68**	0.00	0.43	**909.68**	0.00	0.43	**909.68**	0.00	10.50	**909.68**
8	100	C, D	**865.94**	0.00	4.28	**865.94**	0.00	0.44	**865.94**	0.00	0.44	868.32	0.27	5.05	**865.94**
9	150	C, D	1163.19	0.06	5.83	**1162.55**	0.00	1.22	1164.54	0.17	0.50	1169.15	0.57	17.88	**1162.55**
10	199	C, D	1408.82	0.93	14.32	1401.12	0.41	2.45	1404.67	0.42	0.45	1418.79	1.64	43.86	**1395.85**
11	120	C	**1042.11**	0.00	0.21	**1042.11**	0.00	0.05	**1042.11**	0.00	0.05	1043.11	0.10	22.43	**1042.11**
12	100	C	**819.56**	0.00	0.10	**819.56**	0.00	0.01	**819.56**	0.00	0.01	**819.56**	0.00	7.21	**819.56**
13	120	C, D	1544.01	0.19	8.75	**1541.14**	0.00	0.63	1543.26	0.14	0.47	1553.12	0.78	34.91	**1541.14**
14	100	C, D	**866.37**	0.00	0.10	**866.37**	0.00	0.08	**866.37**	0.00	0.08	**866.37**	0.00	4.73	**866.37**
Average				0.23	5.22		0.03	7.72		0.07	0.27		0.49	21.25	

9. Pentium II (400 MHz).
10. For C instances, see Mester and Bräysy (2004). Otherwise, see Mester (2004).
11. Pentium IV (2 GHz).
12. Pentium (400 MHz).

Table 1.2. Computational results for the Golden et al. (1998) instances

			GTS Toth and Vigo (2003)			Li, Golden and Wasil (2004)		USTA Cordeau et al. (2001)			VLNS Ergun et al. (2003)			Prins (2004)		
Instance		Type[1]	Value	%	Minutes[2]	Value[3]	%	Value[4]	%	Minutes[5]	Value[6]	%	Minutes[7]	Value	%	Minutes[8]
1	240	C	5736.15	1.93	4.98	5666.42	0.69	5681.97	0.97	10.29	5741.79	2.03	134.95	5646.63	0.34	32.42
2	320	C	8553.03	1.24	8.28	8469.32	0.25	8657.36	2.48	35.39	8917.41	5.56	150.83	**8447.92**	0.00	77.92
3	400	C	11402.75	3.32	12.94	11145.80	0.99	11037.40	0.01	55.39	12106.64	9.70	15.67	**11036.22**	0.00	120.83
4	480	C	14910.62	9.44	15.13	13758.08	0.98	13740.60	0.85	83.19	15316.69	12.42	106.50	**13624.52**	0.00	187.60
5	200	C	6697.53	3.66	2.38	6478.09	0.26	6756.44	4.57	5.13	6570.28	1.69	15.50	**6460.98**	0.00	1.04
6	280	C	8963.32	6.54	4.65	8539.61	1.51	8537.17	1.48	18.64	8836.25	5.03	81.98	**8412.80**	0.00	9.97
7	360	C	10547.44	3.45	11.66	10289.72	0.92	10267.40	0.70	25.60	11116.68	9.03	85.00	10195.59	0.00	39.05
8	440	C	12036.24	3.20	11.08	11920.52	2.20	11869.50	1.77	71.44	12634.17	8.32	33.95	11828.78	1.42	88.30
9	255	C,D	593.35	1.71	11.67	588.25	0.83	587.39	0.69	37.26	587.89	0.77	49.20	591.54	1.40	14.32
10	323	C,D	751.66	1.30	15.83	749.49	1.01	752.76	1.45	51.11	749.85	1.05	125.05	751.41	1.26	36.58
11	399	C,D	936.04	1.92	33.12	925.91	0.81	929.07	1.16	41.54	932.74	1.56	171.05	933.04	1.59	78.50
12	483	C,D	1147.14	3.61	42.90	1128.03	1.88	1119.52	1.11	157.01	1134.63	2.48	388.62	1133.79	2.40	30.87
13	252	C,D	868.80	1.13	11.43	865.20	0.71	875.88	1.95	34.83	870.90	1.37	235.13	875.16	1.87	15.30
14	320	C,D	1096.18	1.38	14.51	1097.78	1.52	1102.03	1.92	21.56	1097.11	1.46	31.17	1086.24	0.46	34.07
15	396	C,D	1369.44	1.80	18.45	1361.41	1.20	1363.76	1.38	57.64	1367.15	1.63	65.30	1367.37	1.65	110.48
16	480	C,D	1652.32	1.83	23.07	1635.58	0.79	1647.06	1.50	129.50	1643.00	1.25	31.58	1650.94	1.74	130.97
17	240	C,D	711.07	0.46	14.29	711.74	0.56	710.93	0.44	18.03	716.46	1.22	223.62	710.42	0.37	5.86
18	300	C,D	1016.83	1.81	21.45	1010.32	1.16	1014.62	1.59	67.11	1023.32	2.46	299.23	1014.80	1.61	39.33
19	360	C,D	1400.96	2.49	30.06	1382.59	1.15	1383.79	1.24	66.21	1404.84	2.78	393.03	1376.49	0.70	74.25
20	420	C,D	1915.83	5.20	43.05	1850.92	1.63	1854.24	1.82	135.29	1883.33	3.41	121.62	1846.55	1.39	210.42
Average				2.87	17.55		1.05		1.45	56.11		3.76	137.95		0.91	66.90

1. C: Capacity restrictions; D: Route length restrictions.
2. Pentium (200 MHz).
3. Best variant ($\alpha = 0.01$)
4. Results of recent computational experiments (see Section 3.3).
5. Pentium IV (2GHz).
6. Best of two runs.
7. Time for reaching the best value for the first time (Pentium III, 733 MHz).
8. GHz PC (75 MFlops).

Table 1.2. (continued). Computational results for the Golden et al. (1998) instances

		Bone Route (Tarantitis and Kiranoudis (2002))			AGES best Mester and Bräysy (2004)			AGES fast Mester and Bräysy (2004)			D-Ants Reimann et al. (2004)			
Instance	Type	Value	%	Minutes[9]	Value[10]	%	Minutes[11]	Value[10]	%	Minutes[11]	Value[12]	%	Minutes[13]	Best
1 240	C	5676.97	0.88	27.86	**5627.54**	0.00	8.73	5644.00	0.30	0.70	5644.02	0.29	62.52	**5627.54**
2 320	C	8512.64	0.77	55.62	**8447.92**	0.00	46.66	8468.00	0.24	0.20	8449.12	0.01	57.67	**8447.92**
3 400	C	11199.72	1.48	59.21	**11036.22**	0.00	40.55	11146.00	0.99	0.70	**11036.22**	0.00	21.92	**11036.22**
4 480	C	13637.53	0.10	47.63	**13624.52**	0.00	470.00	13704.52	0.59	2.50	13699.11	0.55	119.12	**13624.52**
5 200	C	**6460.98**	0.00	11.34	**6460.98**	0.00	0.17	6466.00	0.08	0.50	**6460.98**	0.00	0.87	**6460.98**
6 280	C	8429.28	0.20	12.54	**8412.88**	0.00	75.22	8539.61	1.51	0.10	8412.90	0.00	5.72	**8412.80**
7 360	C	10216.50	0.21	42.50	**10195.56**	0.00	2.55	10240.42	0.44	0.85	10195.59	0.00	14.03	**10195.56**
8 440	C	11936.16	2.34	79.69	**11663.55**	0.00	34.30	11918.75	2.19	0.27	11828.78	1.42	35.30	**11663.55**
9 255	C,D				**583.39**	0.00	8.33	588.25	0.83	0.80	586.87	0.60	21.52	**583.39**
10 323	C,D				**742.03**	0.00	6.00	752.92	1.39	0.43	750.77	1.25	17.48	**742.03**
11 399	C,D				**918.45**	0.00	110.00	925.94	0.82	1.10	927.27	0.96	96.88	**918.45**
12 483	C,D				**1107.19**	0.00	600.00	1128.67	1.94	1.50	1140.87	3.04	61.38	**1107.19**
13 252	C,D				**859.11**	0.00	10.25	865.20	0.71	0.18	865.07	0.69	87.20	**859.11**
14 320	C,D				**1081.31**	0.00	1.22	1097.68	1.51	0.28	1093.77	1.15	25.85	**1081.31**
15 396	C,D				**1345.23**	0.00	7.17	1354.76	0.71	0.26	1358.21	0.96	23.80	**1345.23**
16 480	C,D				**1622.69**	0.00	20.00	1634.99	0.76	1.15	1635.16	0.77	39.90	**1622.69**
17 240	C,D				**707.79**	0.00	0.75	710.22	0.34	0.16	708.76	0.14	68.50	**707.79**
18 300	C,D				**998.73**	0.00	2.50	1009.53	1.08	0.18	998.83	0.01	42.73	**998.73**
19 360	C,D				**1366.86**	0.00	6.00	1381.88	1.10	0.25	1367.20	0.02	112.80	**1366.86**
20 420	C,D				**1821.15**	0.00	8.40	1840.57	1.03	0.55	1822.94	0.10	71.42	**1821.15**
Average			0.74	42.05		0.00	72.94		0.93	0.63		0.60	49.33	

9. Pentium II (400 MHz).
10. For C instances, see Mester and Bräysy (2004). Otherwise, see Mester (2004).
11. Pentium IV (2GHz).
12. Best value obtained in several experiments.
13. Pentium (900 MHz).

LOCAL SEARCH METHODS FOR THE VRPTW

TABU SEARCH FOR VRPTW

- Initial solution: typically created with some cheapest insertion heuristic.
- Improvement using local search with one or more neighborhood structures and the best-accept strategy. Most of the neighborhoods used are well known.
- To reduce the complexity of the search, some authors propose special strategies for limiting the neighborhood.
- To cross the barriers of the search space, created by time window constraints, some authors allow infeasibilities during the search. The violations of constraints are penalized in the cost function and the parameter values regarding each type of violation are adjusted dynamically.
- Since the number of routes is often considered as the primary objective, some authors use different explicit strategies for reducing the number of routes.
- Most of the proposed tabu searches use specialized diversification and intensification strategies to guide the search (e.g., “adaptive memory”, Rochat and Taillard, 1995).
- Several authors report using various post-optimization techniques.

THE MAIN FEATURES OF TABU SEARCH HEURISTICS FOR VRPTW

Authors	Year	Initial solution	Neighborhood Operators	Route min.	Notes
Garcia et al.	1994	Solomon's I1 heuristic	2-opt*, Or-opt	Yes	Neighborhood restricted to arcs close in distance
Rochat et al.	1995	Modification of Solomon's I1, 2-opt	2-opt, relocate	No	Adaptive memory
Carlton	1995	Insertion heuristic	relocate	No	Reactive tabu search
Potvin et al.	1996	Solomon's I1 heuristic	2-opt*, Or-opt	Yes	Neighborhood restricted to arcs close in distance
Taillard et al.	1997	Solomon's I1 heuristic	CROSS	No	Soft time windows, adaptive memory
Badeau et al.	1997	Solomon's I1 heuristic	CROSS	No	Soft time windows, adaptive memory
Chiang et al	1997	Modification of Russell (1995)	λ-interchange	No	Reactive tabu search
De Backer et	1997	Savings heuristic	exchange, relocate,	No	Constraint programming

al.			2-opt*, 2-opt, Or-opt		used to check feasibility of moves
Brandão	1999	Insertion heuristic	relocate, exchange, GENI	No	Neighborhoods restricted to arcs close in distance
Schulze et al.	1999	Solomon's I1, parallel I1 and savings heuristic	Ejection chains, Or-opt	Yes	Generated routes stored in a pool
Tan et al.	2000	Insertion heuristic of Thangiah (1994)	λ-interchange, 2-opt*	No	———
Lau et al.	2000	Insertion heuristic	exchange, relocate	No	Constraint based diversification
Cordeau et al.	2001	Modification of Sweep heuristic	relocate, GENI	No	———
Lau et al.	2002	Relocation from a holding list	exchange, relocate	Yes	Holding list for unrouted nodes, limit for number of routes

PERFORMANCE OF TABU SEARCH HEURISTICS FOR THE VRPTW

Average results with respect to Solomon's benchmarks. The notations CNV and CTD in the rightmost column indicate the cumulative number of vehicles and cumulative total distance over all 56 test problems.

Authors	R1	R2	C1	C2	RC1	RC2	CNV/CTD
Garcia et al.	**12.92**	**3.09**	**10.00**	**3.00**	**12.88**	**3.75**	**436**
(1994)	1317.7	1222.6	877.1	602.3	1473.5	1527.0	65977
Rochat et al.	**12.25**	**2.91**	**10.00**	**3.00**	**11.88**	**3.38**	**415**
(1995)	12085	961.72	828.4	589.9	1377.4	1119.6	57231
Potvin et al.	**12.50**	**3.09**	**10.00**	**3.00**	**12.63**	**3.38**	**426**
(1996)	1294.5	1154.4	850.2	594.6	1456.3	1404.8	63530
Taillard et al.	**12.17**	**2.82**	**10.00**	**3.00**	**11.50**	**3.38**	**410**
(1997)	1209.3	980.27	828.4	589.9	1389.2	1117.4	57523
Chiang et al.	**12.17**	**2.73**	**10.00**	**3.00**	**11.88**	**3.25**	**411**
(1997)	1204.2	986.32	828.4	591.4	1397.4	1229.5	58502
De Backer et al.	**14.17**	**5.27**	**10.00**	**3.25**	**14.25**	**6.25**	**508**
(1997)	1214.9	930.18	829.8	604.8	1385.1	1100.0	56998
Brandão (1999)	**12.58**	**3.18**	**10.00**	**3.00**	**12.13**	**3.50**	**425**
	1205	995	829	591	1371	1250	58562
Schulze et al.	**12.25**	**2.82**	**10.00**	**3.00**	**11.75**	**3.38**	**414**
(1999)	1239.1	1066.7	828.9	589.9	1409.3	1286.0	60346
Tan et al. (2000)	**13.83**	**3.82**	**10.00**	**3.25**	**13.63**	**4.25**	**467**
	1266.4	1080.2	870.9	634.8	1458.2	1293.4	62008
Lau et al. (2000)	**14.00**	**3.55**	**10.00**	**3.00**	**13.63**	**4.25**	**464**
	1211.5	960.43	832.1	612.2	1385.0	1232.6	58432
Cordeau et al.	**12.08**	**2.73**	**10.00**	**3.00**	**11.50**	**3.25**	**407**
(2001)	1210.1	969.57	828.4	589.9	1389.8	1134.5	57556
Lau et al. (2002)	**12.17**	**3.00**	**10.00**	**3.00**	**12.25**	**3.38**	**418**
	1211.5	1001.1	832.1	589.9	1418.8	1170.9	58477

RECENT WORK ON THE VRPTW

- Gehring and Homberger have proposed larger benchmark instances for the VRPTW (200-1000 customers)
- Several authors have presented methods for tackling these.
- Survey by Gendreau and Tarantilis almost completed.

REFERENCES

Introductory references on Local Search

Aarts, E. and J.K. Lenstra (eds.) (2003), *Local Search in Combinatorial Optimization*, Wiley, Chichester.

Gendreau, M. (2003), “An Introduction to Tabu Search”, in *Handbook of Metaheuristics*, F.W. Glover and G.A. Kochenberger (eds.), Kluwer, Boston, MA, 37-54.

Gendreau, M., and J.-Y. Potvin (2005), “Tabu Search”, in *Search Methodologies : Introductory Tutorials in Optimization and Decision Support Techniques*, E. Burke and G. Kendall (eds.), Springer, New York, NY (USA), 165-186.

Gendreau, M., and J.-Y. Potvin (2005), “Metaheuristics in Combinatorial Optimization”, *Annals of Operations Research* **140**, 189-213.

Glover, F., É. Taillard and D. de Werra (1993), “A User's Guide to Tabu Search”, *Annals of O.R.* **41**, 3-28.

Glover and G.A. Kochenberger (eds.) (2003), *Handbook of Metaheuristics*, Kluwer, Boston, MA

Glover, F. and M. Laguna (1997), *Tabu Search*, Kluwer.

Applications

de Werra, D., F. Glover, M. Laguna, É. Taillard (eds.) (1993), *Annals of Operations Research* **41**, "Tabu Search".

Doerner, K.F., M. Gendreau, P. Greistorfer, W.J. Gutjahr, R.F. Hartl, and M. Reimann (eds.) (2007), *Metaheuristics - Progress in Complex Systems Optimization*, Springer Science+Business Media, New York, NY (USA), 408 pages.

Laporte, G. and I. Osman (eds.) (1996), *Annals of Operations Research* **63**, "Metaheuristics in Combinatorial Optimization".

Osman, I.H. and J.P. Kelly (eds.) (1996), *Meta-Heuristics: Theory and Applications*, Kluwer Academic Publishers, Norwell, MA.

Ribeiro, C.C. and P. Hansen (eds.) (2002), *Essays and Surveys in Metaheuristics*, Kluwer Academic Publishers, Norwell, MA.

Voss, S., S. Martello, I.H. Osman and C. Roucairol (eds.) (1999), *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Kluwer Academic Publishers, Norwell, MA.

Other references

Battiti, R. and G. Tecchiolli (1994), "The Reactive Tabu Search", *ORSA Journal on Computing* **6**, 126-140.

Bräysy, O. and M. Gendreau (2002), "Tabu Search Heuristics for the Vehicle Routing Problem with Time Windows", *TOP* **10**, 211-237.

Crainic, T.G. and M. Gendreau (1999), "Towards an Evolutionary Method – Cooperative Multi-Thread Parallel Tabu Search Heuristic Hybrid", in *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, S. Voss, S. Martello, I.H. Osman and C. Roucairol (eds.), Kluwer Academic Publishers, pp. 331-344.

Crainic, T.G. and M. Gendreau (2002), "Cooperative Parallel Tabu Search for Capacitated Network Design", *Journal of Heuristics* **8**, 601-627.

Crainic, T.G., M. Gendreau and J.M. Farvolden (2000) "Simplex-based Tabu Search for the Multicommodity Capacitated Fixed Charge Network Design Problem", *INFORMS Journal on Computing* **12**, 223-236.

Crainic, T.G., M. Toulouse and M. Gendreau (1997), "Toward a Taxonomy of Parallel Tabu Search Heuristics", *INFORMS Journal on Computing* **9**, 61-72.

Cung, V.-D., S.L. Martins, C.C. Ribeiro and C. Roucairol (2002), "Strategies for the Parallel Implementation of Metaheuristics", in *Essays and Surveys in Metaheuristics*, C.C. Ribeiro and P. Hansen (eds.), Kluwer Academic Publishers, pp. 263-308.

Dueck, G. (1993), "New optimization heuristics: The great deluge algorithm and record-to-record travel", *Journal of Computational Physics* **90**, 161–175.

Dueck, G. and T. Scheurer (1990), "Threshold accepting: A general purpose optimization algorithm", *Journal of Computational Physics* **104**, 86–92.

Gendreau, M. (2002), "Recent Advances in Tabu Search", in *Essays and Surveys in Metaheuristics*, C.C. Ribeiro and P. Hansen (eds.), Kluwer Academic Publishers, pp. 369-377.

Gendreau, M., F. Guertin, J.-Y. Potvin and É.D. Taillard (1999), "Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching", *Transportation Science* **33**, 381-390.

Gendreau, M., A. Hertz and G. Laporte (1994), "A Tabu Search Heuristic for the Vehicle Routing Problem", *Management Science* **40**, 1276-1290.

Gendreau, M., G. Laporte and J.-Y. Potvin (2002), "Metaheuristics for the Capacitated VRP", in *The Vehicle Routing* Problem, P. Toth and D. Vigo (eds.), SIAM Monographs on Discrete Mathematics and Applications, pp. 129-154.

Gendreau, M., P. Soriano and L. Salvail (1993), "Solving the Maximum Clique Problem Using a Tabu Search Approach", *Annals of Operations Research* **41**, 385-403.

Glover, F. (1977), "Heuristics for Integer Programming Using Surrogate Constraints", *Decision Sciences* **8**, 156-166.

Glover, F. (1986), "Future Paths for Integer Programming and Links to Artificial Intelligence", *Computers and Operations Research* **13**, 533-549.

Glover, F. (1989), "Tabu Search – Part I", *ORSA Journal on Computing* **1**, 190-206.

Glover, F. (1990), "Tabu Search – Part II", *ORSA Journal on Computing* **2**, 4-32.

Glover, F. (1992), "Ejection chains, Reference Structures and Alternating Path Methods for Traveling Salesman Problems", University of Colorado. Shortened version published in *Discrete Applied Mathematics* **65**, 223-253, 1996.

Grünert, T. (2002), "Lagrangean Tabu Search", in *Essays and Surveys in Metaheuristics*, C.C. Ribeiro and P. Hansen (eds.), Kluwer Academic Publishers, pp. 379-397.

Kirkpatrick, S., C.D. Gelatt Jr. and M.P. Vecchi (1983), "Optimization by Simulated Annealing", *Science* **220**, 671-680.

Lokketangen, A. and F. Glover (1996), "Probabilistic Move Selection in Tabu Search for 0/1 Mixed Integer Programming Problems", in *Meta-Heuristics: Theory and Applications*, I.H. Osman and J.P. Kelly (eds.), Kluwer Academic Publishers, pp. 467-488.

Osman, I.H. (1993), "Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem", *Annals of Operations Research* **41**, 421-451.

Pesant, G. and M. Gendreau (1999), "A Constraint Programming Framework for Local Search Methods", *Journal of Heuristics* **5**, 255-280.

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

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

Skorin-Kapov, J. (1990), "Tabu Search Applied to the Quadratic Assignment Problem", *ORSA Journal on Computing* **2**, 33-45.

Soriano, P. and M. Gendreau (1996), "Diversification Strategies in Tabu Search Algorithms for the Maximum Clique Problems", *Annals of Operations Research* **63**, 189-207.

Taillard, É. (1990), "Some efficient heuristic methods for the flow shop sequencing problem", *European Journal of Operational Research* **47**, 65-74.

Taillard, É. (1991), "Robust taboo search for the quadratic assignment problem", *Parallel Computing* **17**, 443-455.

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

Toth, P., and Vigo, D. (2003): "The Granular Tabu Search (and its application to the Vehicle Routing Problem)", *Informs Journal on Computing* 15, 333–346.

1 Vehicle Routing Problem

The VRP [19] can be formally defined as follows. Let $G = (V, A)$ be a graph with A the arc set and $V = \{1, ..., n\}$ the vertex set, where vertex 1 is the depot and the other vertices are cities or customers to be served. With every arc (i, j), $i \neq j$, is associated a non-negative distance matrix $D = (d_{ij})$, where d_{ij} can be interpreted either as a true distance, a travel time or a travel cost. Note that the undirected version of the VRP is obtained when D is symmetric. A fleet of vehicles, based at the depot, is available for serving the vertices. Usually, the number of vehicles is variable, and a fixed cost f is incurred each time a new vehicle is used. It can also happen that the number of vehicles is fixed or upper bounded. A non-negative weight or demand q_i is associated with each vertex $i > 1$ and the sum of demands on any vehicle route should not exceed the vehicle capacity. The capacity and fixed cost can be the same for all vehicles (homogeneous fleet) or not (heterogeneous fleet). In some variants, the total travel distance or total travel time of each vehicle is also constrained. The problem is to find a set of least-cost vehicle routes such that:

- each vertex in $V - \{1\}$ is served exactly once by exactly one vehicle;
- each vehicle route starts and ends at the depot;
- all side constraints are satisfied (capacity, maximum travel distance or maximum travel time).

Note that this section also covers methods developed to solve Open VRP (OVRP), in which each route is a Hamiltonian path instead of Hamiltonian cycle; this difference comes from the fact that vehicles do not return to the starting depot or, if they do so, they must follow the same path backwards. Problems with multiple objectives are also considered.

The reader is referred to [9] for a general survey about metaheuristics for the classical VRP with capacity constraints. References on specific metaheuristics are found in the following subsections.

1.1 Simulated annealing

I.H. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41:421–451, 1993.

I. Zeng, H.L. Ong and K.M. Ong. An assignment-based local search method for solving vehicle routing problems. *Asia-Pacific Journal of Operational Research*, 22:85–104, 2005.

S. Chen, B. Golden and E. Wasil. The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. *Networks*, 49:318–329, 2007.

1.2 Tabu search

I.H. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41:421–451, 1993.

É.D. Taillard. Parallel iterative search methods for vehicle routing problems. *Networks*, 23:661–673, 1993.

M. Gendreau, A. Hertz and G. Laporte. A tabu search heuristic for the vehicle routing problem. *Management Science*, 40:1276–1290, 1994.

Y. Rochat and É. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1:147–167, 1995.

C. Rego. A subpath ejection method for the vehicle routing problem. *Management Science*, 44:1447-1459, 1998.

G. Barbarosoglu and D. Ozgur. A tabu search algorithm for the vehicle routing problem. *Computers & Operations Research*, 26:255–270, 1999.

J.-F. Cordeau, G. Laporte and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52:928–936, 2001.

C.D. Tarantilis and C.T. Kiranoudis. Boneroute: an adaptive memory-based method for effective fleet management. *Annals of Operations Research*, 115:227–241, 2002.

P. Toth and D. Vigo. The granular tabu search and its application to the vehicle routing problem. *INFORMS Journal on Computing*, 15:333–348, 2003.

J. Brandão. A tabu search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, 157:552–564, 2004.

Z. Fu, R.W. Eglese and L. Li. A new tabu search heuristic for the open vehicle routing problem. *Journal of the Operational Research Society*, 56:267–274, 2005.

C.D. Tarantilis. Solving the vehicle routing problem with adaptive memory programming methodology. *Computers & Operations Research*, 32:2309–2327, 2005.

C. Archetti, A. Hertz and M.G. Speranza. A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science*, 40:64 73, 2006.

M. Gendreau, M. Iori, G. Laporte and S. Martello. A tabu search algorithm for a routing and container loading problem. *Transportation Science*, 40:342–350, 2006.

N.A. Wassan. A reactive tabu search for the vehicle routing problem. *Journal of the Operational Research Society*, 57:111–116, 2006.

U. Derigs and R. Kaiser. Applying the attribute based hill climber heuristic to the vehicle routing problem. *European Journal of Operational Research*, 177:719–732, 2007.

D. Pisinger and S. Røpke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34:2403–2435, 2007.

1.3 Variable neighborhood search

J. Kytöjoki, T. Nuortio, O. Bräysy and M. Gendreau. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Computers & Operations Research*, 34:2743–2757, 2007.

2 VRP with Time Windows

In the VRP with Time Windows (VRPTW) [2], a time interval $[a_i, b_i]$ is associated with vertex $i \in V$. In the hard time window variant, the vertex must be served within that interval (although the vehicle can wait, if it arrives before the lower bound a_i). In the soft time window variant, the vertex can be served outside of its time interval, but a penalty is incurred in the objective. A general survey about metaheuristics for the VRPTW is found in [1].

2.1 Simulated annealing

W.-C. Chiang and R.A. Russell. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, 63:3–27, 1996.

K.C. Tan, L.H. Lee, Q.L. Zhu and K. Ou. Heuristic methods for vehicle routing problem with time windows. *Artificial Intelligence in Engineering*, 15:281–295, 2001.

Z. Czech and P. Czarnas. Parallel simulated annealing for the vehicle routing problem with time windows. In *Proceedings of 10th Euromicro Workshop on Parallel Distributed and Network-Based Processing*, Canary Islands, Spain, 376–383, 2002.

2.2 Tabu search

B.-L. Garcia, J.-Y. Potvin and J.-M. Rousseau. A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. *Computers & Operations Research*, 21:1025–1033, 1994.

Y. Rochat and É. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1:147–167, 1995.

J.-Y. Potvin, T. Kervahut, B.L. Garcia and J.-M. Rousseau. The vehicle routing problem with time windows - Part I: Tabu search. *INFORMS Journal on Computing*, 8:157–164, 1996.

P. Badeau, M. Gendreau, F. Guertin, J.-Y. Potvin and É. Taillard. A parallel tabu search heuristic for the vehicle routing problem with time windows. *Transportation Research – Part C*, 5, 109–122, 1997.

W.-C. Chiang and R.A. Russell. A reactive tabu search metaheuristic for the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 9:417–430, 1997.

B. De Backer and V. Furnon. Meta-heuristics in constraint programming experiments with tabu search on the vehicle routing problem. In *Proceedings of the Second International Conference on Metaheuristics (MIC'97)*, S. Antipolis, France, 1–14, 1997.

É. Taillard, P. Badeau, M. Gendreau, F. Guertin and J.-Y. Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31:170–186, 1997.

J. Brandão. Metaheuristic for the vehicle routing problem with time windows. In *Meta-heuristics - Advances and Trends in Local Search Paradigms for Optimization*, S. Voss, S. Martello, I.H. Osman and C. Roucairol, eds., Kluwer Academic Publishers, Boston, 19–36, 1999.

J. Schulze and T. Fahle. A parallel algorithm for the vehicle routing problem with time window constraints. *Annals of Operations Research*, 86:585–607, 1999.

J.-F. Cordeau, G. Laporte and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52:928–936, 2001.

K.C. Tan, L.H. Lee, Q.L. Zhu and K. Ou. Heuristic methods for vehicle routing problem with time windows. *Artificial Intelligence in Engineering*, 15:281–295, 2001.

H.C. Lau, M. Sim and K.M. Teo. Vehicle routing problem with time windows and a limited number of vehicles. *European Journal of Operational Research*, 148:559–569, 2003.

J.-F. Cordeau, G. Laporte and A. Mercier. Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. *Journal of Operational Research Society*, 55:542–546, 2004.

S.C. Ho and D. Haugland. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers & Operations Research*, 31:1947–1964, 2004.

D. Pisinger and S. Røpke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34:2403–2435, 2007.

2.3 Variable neighborhood search

L.-M. Rousseau, M. Gendreau and G. Pesant. Using constraint-based operators to solve the vehicle routing problem with time windows. *Journal of Heuristics*, 8:43–58, 2002.

O. Bräysy. A reactive variable neighbourhood search for the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 15: 347–368, 2003.

O. Bräysy, G. Hasle and W. Dullaert. A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research*, 159:586–605, 2004.

3 VRP with Backhauls

In the VRP with Backhauls (VRPB) [20], the demand at each vertex i corresponds either to a delivery or a pick-up (backhaul) which is then brought back to the depot. While goods are picked up or delivered, the quantity on board should never exceed the capacity of the vehicle. This problem is a special case of the VRPPD (see Section 4).

3.1 Tabu search

C. Duhamel, J.-Y. Potvin and J.-M, Rousseau. A tabu search heuristic for the vehicle routing problem with backhauls and time windows. *Transportation Science*, 31:49–59, 1997.

I.H. Osman and N. Wassan. A reactive tabu search metaheuristic for the vehicle routing problem with backhauls. *Journal of Scheduling*, 5:263–285, 2002.

J. Brandão. A new tabu search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research*, 173:540–555, 2006.

4 VRP with Pick-ups and Deliveries

In the VRP with Pick-ups and Deliveries (VRPPD) [3], a transportation request i is associated with two vertices o_i and d_i, and the demand q_i should be picked up at o_i and delivered at d_i. For a solution to be feasible, both o_i and d_i should be in the same route. Furthermore, o_i should appear before d_i in the route. In this problem, capacity constraints can be present or not, depending on the application, and a time window is typically associated with each vertex. For example, in transportation-on-demand applications where people with special needs are transported (a problem referred to as the Dial-A-Ride Problem), there are both capacity and time window constraints. Furthermore, there is a constraint on the maximum ride time of each passenger.

4.1 Simulated annealing

S.M. Hart. The modeling and solution of a class of dial-a-ride problems using simulated annealing. *Control and Cybernetics*, 25:131–157, 1996.

H. Li and A. Lim. A metaheuristic for the pickup and delivery problem with time windows. *International Journal on Artificial Intelligence Tools*, 12:173–186, 2003.

4.2 Tabu search

W.P. Nanry and J.W. Barnes. Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research – Part B*, 34:107–121, 2000.

P. Caricato, G. Ghiani, A. Grieco and E. Guerriero. Parallel tabu search for a pickup and delivery problem with track contention. *Parallel Computing*, 29:631–639, 2003.

J.-F. Cordeau and G. Laporte. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research - Part B*, 37:579–594, 2003.

A. Attanasio, J.-F. Cordeau, G. Ghiani and G. Laporte. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, 30:377–387, 2004.

M. Gendreau, F. Guertin, J.-Y. Potvin and R. Séguin. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. *Transportation Research C* 14:157–174, 2006.

F.A. Montané, R.D. Galvão. A tabu search algorithm for the vehicle routing problem with simultaneous pickup and delivery service. *Computers & Opera-*

tions Research, 33:595–619, 2006.

E. Melachrinoudis, A.B. Ilhan and H. Min. A dial-a-ride problem for client transportation in a health-care organization. *Computers & Operations Research*, 34:742–759, 2007.

4.3 Others

S. Røpke and D. Pisinger. A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171:750-775, 2006.

S. Røpke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40:455-472, 2006.

5 VRP with Multiple Use of Vehicles

In standard vehicle routing problems, it is implicitly assumed that each vehicle serves a single route. In some cases, however, it might be possible or even necessary to assign the vehicle to several routes. This situation happens, for example, when the capacity of the vehicle is relatively small. In this case, frequent returns to the depot are required to load or unload the vehicle.

5.1 Tabu search

É. Taillard, G. Laporte and M. Gendreau. Vehicle routeing with multiple use of vehicles. *Journal of the Operational Reseach Society*, 47:1065–1070, 1996.

J. Brandão and A. Mercer. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European Journal of Operational Research*, 100:180–191, 1997.

J. Brandão and A. Mercer. The multi-trip vehicle routing problem. *Journal of the Operational Research Society*, 49:799–805, 1998.

A. Olivera and O. Viera. Adaptive memory programming for the vehicle routing problem with multiple trips. *Computers & Operations Research*, 34:28–47, 2007.

6 Fleet Size and Mix VRP

When the number of vehicles is free and the fleet is heterogeneous, one is faced with the Fleet Size and Mix VRP (FSMVRP) [8], which exhibits special features that need to be addressed through specific algorithmic procedures. In particular, the benefits of replacing one type of vehicle by another for serving a particular route must be taken into account. We also include in this section methods devised for solving the VRP with trailers (VRPT), where one has to determine the optimal deployment of a vehicle fleet of truck-trailer combinations.

6.1 Simulated annealing

R. Tavakkoli-Moghaddam, N. Safaei and Y. Gholipour. A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length. *Applied Mathematics and Computation*, 176:445–454, 2006.

F. Li, B. Golden and E. Wasil. A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 34:2734-2742, 2007.

6.2 Tabu search

I.H. Osman and S. Salhi. Local search strategies for the vehicle fleet mix problem. In *Modern Heuristic Search Methods*, V.J. Rayward-Smith, I.H. Osman, C.R. Reeves and G.D. Smith, eds., John Wiley & Sons, Chichester, 131–153, 1996.

M. Gendreau, G. Laporte, C. Musaraganyi and É. Taillard. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 26:1153–1173, 1999.

R. Mechti, S. Poujade, C. Roucairol and B. Lemarié. Global and local moves in tabu search: a real-life mail collecting application. In *Meta-heuristics Advances and Trends in Local Search Paradigms for Optimization*, S. Voss, S. Martello, I.H. Osman and C. Roucairol, eds., Kluwer, Boston, 155–174, 1999.

I.-M. Chao. A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, 29:33–51, 2002.

N.A. Wassan and I.H. Osman. Tabu search variants for the mix fleet vehicle routing problem. *Journal of the Operational Research Society*, 53:768–782, 2002.

S. Scheurer. A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research*, 33:894–909, 2006.

6.3 Others

C.D. Tarantilis, C.T. Kiranoudis and V.S. Vassiliadis. A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *Journal of the Operational Research Society*, 54:65–71, 2003.

C.D. Tarantilis, C.T. Kiranoudis and V.S. Vassiliadis. A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *European Journal of Operational Research*, 152:148–158, 2004.

7 VRP with Multiple Depots and Periodic VRP

In the VRP with Multiple Depots (MDVRP), there is not a single depot, but rather a number of depots with different locations and an associated fleet of vehicles. Depending on the variant considered, each vehicle may be required to terminate its route at its starting depot.

The Periodic VRP (PVRP) is an extension of the VRP in which customers must be visited one or more times during a planning horizon of several periods with routes performed by vehicles in each period. By substituting days for depots, one can show the equivalence of some variants of the MDVRP and the PVRP.

7.1 Simulated annealing

A. Lim and W. Zhu. A fast and effective insertion algorithm for multi-depot vehicle routing problem with fixed distribution of vehicles and a new simulated annealing approach. In *IEA/AIE 2006, Lecture Notes in Artificial Intelligence 4031*, M. Ali and R. Dapoigny, eds., Springer, 282–291, 2006. Springer-Verlag.

7.2 Tabu search

J. Renaud, G. Laporte and F.F. Boctor. A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research*, 23:229–235, 1996.

J.-F. Cordeau, M. Gendreau and G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30:105–119, 1997.

E. Hadjiconstantinou and R. Baldacci. A multi-depot period vehicle routing problem arising in the utilities sector. *Journal of the Operational Research Society*, 49:1239–1248, 1998.

J.-F. Cordeau, G. Laporte and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52:928–936, 2001.

E. Angelelli and M.G. Speranza. The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research*, 139:233–247, 2002.

D. Pisinger and S. Røpke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34:2403–2435, 2007.

8 Dynamic VRP

In dynamic vehicle routing problems [10, 17], some data about the problem are not known beforehand. That is, new information are revealed on-line, as the routes are executed by the vehicles. In most cases, a quick or real-time response time is also required. The new information often correspond to the occurrence of a new vertex (customer) that must be included into the current routes. It can also be some new information about the travel time of a vehicle, the current customer status (e.g., cancellation of a transportation request), etc. This section includes (repeats) papers on the dynamic variant of the VRPPD.

8.1 Tabu search

C. Rego and C. Roucairol. Using tabu search for solving a dynamic multi-terminal truck dispatching problem. *European Journal of Operational Research*, 83:411–429, 1995.

M. Gendreau, F. Guertin,, J.-Y. Potvin and E. Taillard. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science*, 33:381–390, 1999.

S. Ichoua, M. Gendreau and J.-Y. Potvin. Diversion issues in real-time vehicle dispatching. *Transportation Science*, 34:426–438, 2000.

S. Ichoua, M. Gendreau and J.-Y. Potvin. Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research*, 144:379–396, 2003.

A. Attanasio, J.-F. Cordeau, G. Ghiani and G. Laporte. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, 30:377–387, 2004.

M. Gendreau, F. Guertin,, J.-Y. Potvin and R. Séguin. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. *Transportation Research – Part C*, 14:1157-174, 2006.

S. Ichoua, M. Gendreau and J.-Y. Potvin. Exploiting knowledge about future demands for real-time vehicle dispatching. *Transportation Science*, 40:211-215, 2006.

References

[1] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation Science*, 39:119–139, 2005.

[2] J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M.M. Solomon and F. Soumis. VRP with Time Windows, Chapter 7 in *The Vehicle Routing Problem*, P. Toth and D. Vigo, eds., SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA, 157–194, 2002.

[3] G. Desaulniers, J. Desrosiers, A. Erdmann, M.M. Solomon and F. Soumis. VRP with Pickup and Delivery, Chapter 9 in *The Vehicle Routing Problem*, P. Toth and D. Vigo, eds., SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA, 225–242, 2002.

[4] M. Dorigo and T. Stützle. *Ant Colony Optimization*. The MIT Press, Cambridge, MA, 2004.

[5] J. Dréo, A. Pétrowski, P. Siarry, and É. Taillard. *Métaheuristiques pour l'Optimisation Difficile*. Eyrolles, France, 2003.

[6] G. Dueck. New optimization heuristics: the great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104:86–92, 1993.

[7] G. Dueck and T. Scheuer. Threshold Accepting: a general purpose optimization algorithm. *Journal of Computational Physics*, 90:161–175, 1990.

[8] T. Etezadi and J.E. Beasley. Vehicle fleet composition. *Journal of the Operational Research Society*, 34:87–91, 1983.

[9] M. Gendreau, G. Laporte, and J.-Y. Potvin. Metaheuristics for the Capacitated VRP, Chapter 6 in *The Vehicle Routing Problem*, P. Toth and D. Vigo, eds., SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA, 129–154, 2002.

[10] M. Gendreau and J.-Y. Potvin. Dynamic Vehicle Routing and Dispatching, Chapter 5 in *Fleet Management and Logistics*, T.G. Crainic and G. Laporte, eds., Kluwer, Boston,115–126, 1998.

[11] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13:533–549, 1986.

[12] F. Glover and M. Laguna. *Tabu Search*. Kluwer, Boston, MA, 1997.

[13] F. Glover and G.A. Kochenberger. *Handbook of Metaheuristics*. Kluwer, Boston, MA, 2003.

[14] D.E. Goldberg. *Genetic Algorithms in Search, Optimization & Learning*. Addison-Wesley, Reading, MA, 1989.

[15] P. Hansen and N. Mladenović. Variable Neighborhood Search. Chapter 6 in *Handbook of Metaheuristics*, F. Glover and G.A. Kochenberger, eds., Kluwer, 145–184, 2003.

[16] P.J.M. van Laarhoven and E.H.L. Aarts. *Simulated Annealing: Theory and Applications*. Springer, Dordrecht, 1987.

[17] H.N. Psaraftis. Dynamic Vehicle Routing Problems. Chapter 11 in *Vehicle Routing: Methods and Studies*, B.L. Golden and A.A. Assad, eds., North-Holland, Amsterdam, 223–248, 1988.

[18] M.G.C. Resende and C.C. Ribeiro. Greedy Randomized Adaptive Search Procedures. Chapter 8 in *Handbook of Metaheuristics*, F. Glover and G.A. Kochenberger, eds., Kluwer, 219–249, 2003.

[19] P. Toth and D. Vigo. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA, 2002.

[20] P. Toth and D. Vigo. VRP with Backhauls, Chapter 8 in *The Vehicle Routing Problem*, P. Toth and D. Vigo, eds., SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA, 195–224, 2002.