Heuristics for Vehicle Routing Problems: Structural Problem decompositions and Unified Search

Thibaut Vidal
Departamento de Informatica
Pontificia Universidade Catolica do Rio de Janeiro

VeRoLog PhD School
June 1\textsuperscript{st}-3\textsuperscript{rd}, Cagliari, Italy
## Contents

1. **Multi-attribute Vehicle Routing Problems**

2. **Heuristics and Metaheuristics**
   - A quick guided tour of CVRP metaheuristics
   - Successful strategies – Analysis

3. **Unified Hybrid Genetic Search**
   - General description
   - Tricks of the trade

4. **Structural Problem Decompositions**
   - Arc Routing Problems
   - Team-Orienteering Problems
   - CVRP – Sequence or Set optimization?

5. **Conclusions and Perspectives**
1 Multi-attribute Vehicle Routing Problems

2 Heuristics and Metaheuristics
   • A quick guided tour of CVRP metaheuristics
   • Successful strategies – Analysis

3 Unified Hybrid Genetic Search
   • General description
   • Tricks of the trade

4 Structural Problem Decompositions
   • Arc Routing Problems
   • Team-Orienteering Problems
   • CVRP – Sequence or Set optimization?

5 Conclusions and Perspectives
Multi-attribute vehicle routing problems (MAVRPs)

- Capacitated vehicle routing problems (VRP)
  - **INPUT**: $n$ customers, with locations and demand quantity. All-pair distances. Homogeneous fleet of $m$ vehicles with capacity $Q$ located at a central depot.
  - **OUTPUT**: Least-cost delivery routes (at most one route per vehicle) to service all customers.
  
- NP-Hard problem
- recent breakthrough in exact methods enable to solve problems of moderate size with up to 300-400 customers (Uchoa et al., 2013).
- A Scopus search “Vehicle Routing” for 2007-2011 returns 1258 publications, including 566 journal papers.
- Massive research on heuristics
- Diverse and larger instances available (Uchoa et al., 2017)
• Capacitated vehicle routing problems (VRP)

   - **Combinatorial explosion:** For a problem with **n=100 customers** and **a single vehicle**, the number of feasible solutions is:

   \[ n! = 93326215443944152681699 \]

   \[ 2388562667004907159682643816 \]

   \[ 2146859296389521759999322991 \]

   \[ 5608941463976156518286253697 \]

   \[ 9208272237582511852109168640 \]

   \[ 0000000000000000000000000 \approx 10^{158} \]
Even with a grid of computers which:

- Contains as many CPUs as the estimated number of atoms in the universe: \( n_{\text{CPU}} = 10^{80} \)
- Does one operation per Planck time: \( t_P = 5.39 \times 10^{-44} \) seconds

We would need \( T = \frac{10^{158} \times 5.39 \times 10^{-44}}{1080} = 5.39 \times 10^{34} \) seconds to enumerate all solutions.

Compare this to the estimated age of Universe: \( 4.33 \times 10^{17} \) seconds
Multi-attribute vehicle routing problems (MAVRPs)

- **VRP “attributes”:** Supplementary decisions, constraints and objectives combined with the classic VRP (Vidal et al., 2013b)
  - **Realistic objectives:** Profitability, equity, service Levels, persistence, compactness, robustness, externalities
  - **Integrated planning:** Multiple periods, depots, echelons, fleet mix, LRP, IRP, synchronization...
  - **Fine-grained modeling:** Time windows (soft or multiple), loading constraints (2D,3D), driver skills, time-dependent travel times, charging stations, engine modes, drones etc...
Multi-attribute vehicle routing problems (MAVRPs)

- **VRP “attributes”:** Supplementary decisions, constraints and objectives combined with the classic VRP (Vidal et al., 2013b)
Multi-attribute vehicle routing problems (MAVRPs)

- **VRP “attributes”:** Supplementary decisions, constraints and objectives which complement the classic VRP formulation (Vidal et al., 2013b)
Multi-attribute vehicle routing problems (MAVRPs)

- Three main resolution tasks and related problem attributes
  - ASSIGNMENT (assignment of customers and routes to time-periods or depots)
    - multi-period, multi-depot, heter. fleet, location routing...
  - SEQUENCING (choice of the sequence of visits)
    - P&D, Backhauls, 2-echelon...
  - ROUTE EVALUATION (route feasibility/cost & other decisions)
    - Time windows, time-dep travel time, loading constraints, HOS regulations, lunch breaks, load-dependent costs...
Multi-attribute vehicle routing problems (MAVRPs)

- Challenges: **VARIETY** and **COMBINATION** of attributes
- *Over 200 attributes* have been proposed to this date...
  ...which often appear together ⇒ $2^{200}$ problems... $2^{200}$ different methods, and $2^{200}$ papers ?!!!
- “Double” combinatorial explosion: Combinatorial optimization problem and combinatorial *family of problems*

⇒ Progress towards unified solution concepts and methods
⇒ Solvers that can address a wide range of problems without need for extensive adaptation or user expertise.
⇒ Necessary tools for the timely application of current optimization methods to industrial settings.

[SHOW EXAMPLE 2. PROBLEM VARIETY]
Contents

1 Multi-attribute Vehicle Routing Problems

2 Heuristics and Metaheuristics
   - A quick guided tour of CVRP metaheuristics
   - Successful strategies – Analysis

3 Unified Hybrid Genetic Search
   - General description
   - Tricks of the trade

4 Structural Problem Decompositions
   - Arc Routing Problems
   - Team-Orienteering Problems
   - CVRP – Sequence or Set optimization?

5 Conclusions and Perspectives
Constructive methods

- **Constructive methods**: mostly between 1960s and 1980s.
  - Making step-by-step definitive decisions, which cannot be revoked afterwards

- Savings method (Clarke and Wright 1964)
  - Merge routes step by step based on a savings measure $s_{ij} = c_{i0} + c_{0j} - c_{ij}$
  - Some refinements by Gaskell (1967) and Yellow (1970): $s_{ij} = c_{i0} + c_{0j} - \lambda c_{ij}$
  - Mole and Jameson (1976) and Solomon (1987) later generalize the concepts and consider insertions inside the routes.
Constructive methods

• **Constructive methods**: mostly between 1960s and 1980s.
  ▶ Making step-by-step definitive decisions, which cannot be revoked afterwards

• Sweep algorithm (Gillett and Miller, 1974)
  ▶ Sweep the deliveries in circular order to create routes.
  ▶ A new route is initiated each time the capacity is exceeded.

• Petal methods: generate several alternative routes, called petals, and select a subset of these by solving a set-covering linear program.
Constructive methods

- **Constructive methods:** mostly between 1960s and 1980s.
  - Making step-by-step definitive decisions, which cannot be revoked afterwards

- Route first cluster second (Newton and Thomas, 1974; Bodin and Berman, 1979; Beasley, 1983)
  - construct a giant circuit (TSP tour) that visits all customers.
  - Segmenting this tour into several routes. Optimal segmentation is assimilated to a shortest path problem in a auxiliary directed acyclic graph
  - Possible to solve the segmentation problem (Split) in $O(n)$ (Vidal, 2016)
Classical Local Searches

- **Local-improvement procedures:**
  - From an *incumbent solution* \( s \) define a *neighborhood* \( N(s) \) of solutions obtained by applying some changes.
  - The set of solutions, linked by neighborhood relationships = search space.
  - LS-improvement method progress from one solution to another in this search space as long as the cost improves.
• For optimizing a single route (TSP tour);
  
  ▶ in the terminology of Lin (1965), \( \lambda \)-opt neighborhood = subset of moves obtained by deleting and reinserting \( \lambda \) arcs.
  
  ▶ 2-opt and 3-opt are commonly used,
  
  ▶ Or-opt which comes to relocate sequences of bounded size, and is a subset of 3-opt.
Classical Local Searches

- For optimizing multiple routes together,
  - Insert neighborhood (relocate a delivery)
  - Swap neighborhoods (swap two deliveries from different routes)
  - CROSS-exchange (exchange two sequences of visits)
  - I-CROSS (exchange and reverse two sequences)
  - 2-opt* exchange two route tails (special case of CROSS)
Classical Local Searches

- These neighborhoods contain a polynomial number of moves.
  - For all moves except CROSS and I-CROSS, the number of neighbors is $O(n^2)$
  - CROSS and I-CROSS are often limited to sequences of bounded size with less than $k$ customers, in that case the number of neighbors becomes $O(k^2 n^2)$

- Other non-enumerative large-scale neighborhoods:
  - Heuristic of Lin and Kernighan (1973) – efficient implementation from Helsgaun (2000);
  - Ruin-and-recreate (Shaw, 1998; Schrimpf et al., 2000);
  - Ejection chains (Glover, 1992, 1996)
### Classical Local Searches

- Efficient move evaluations and pruning procedures are critical to address large-scale problem instances

  - Neighborhood restrictions, granular search (Johnson and McGeoch, 1997; Toth and Vigo, 2003): restrain the subset of moves to spatially related customers

  - Sequential search (Christofides and Eilon, 1972; Irnich and Villeneuve, 2003): any profitable move can be broken down into a list of arc exchanges \((a_1, \ldots, a_\lambda)\) with gains \((g_1, \ldots, g_\lambda)\) such that for any \(k \in \{1, \ldots, \lambda\}\), \(g_1 + \cdots + g_k \geq 0\).

  - This condition allows to prune many non-promising moves.
Classical Metaheuristics

- Local-improvement methods ⇒ local optima.
- Metaheuristics to escape and guide the search
- Main classes of methods:
  - Neighborhood-centered search – iterative improvement of one single solution – Tabu search, Simulated annealing, ILS, VNS...
  - Population-based search – improving a population of solutions – Hybrid GA, evolutionary algorithms, ACO, path relinking...
  - Hybrid approaches – often combining many successful strategies
- Hybrids are very common ⇒ the limits between metaheuristics become blurred – Necessity to use a simple and optimization-oriented terminology to identify their common structures and success elements (Sörensen, 2015).
• Tabu search – choice of **best move** at each step (possibly non-improving).

• Neighborhood: single **RELOCATE**

• Short-term tabu memories to avoid cycling:
  ▶ Moving Client $i$ from route $R_1$ to $R_2$ ⇒ Not allowed to insert $i$ back into route $R_1$ for $X$ iterations.

• Longer term **diversification strategies:**
  ▶ Penalizing recurrent solution attributes in the objective function
  ▶ Penalized infeasible solutions (excess load or duration)
• From 1995, but already contained most of the ideas used nowadays:

• **Diversification**
  ▶ Tabu search based on *Swap* and *Relocate* moves
  ▶ *Probabilistic* selection of moves driven by measures of attractiveness

• **and Intensification:**
  ▶ Detection of good fragments of solutions that consistently appear in elite solutions and creation of new solutions from these fragments to obtain new stating points
  ▶ Decomposition phases based on spatial proximity
  ▶ Exact solution of the TSPs at regular intervals
  ▶ Set covering optimization as a post-optimization
• **Large neighborhoods** based on the *ruin-and-recreate* principle (Shaw, 1998; Schrimpf et al., 2000).

• Variety of operators to **partially destroy** the solutions
  - Based on randomness, cost metrics, relatedness, history...
  - Adaptive probabilities for operator selection

• Variety of operators to **reconstruct** the solutions

• Deteriorating solutions are accepted with some probability, as in a simulated annealing
• **Iterated local search:** at each iteration local search until a local optimum is encountered, *shaking* and local search again...

• A large diversity of neighborhoods is used
  - Relocate and Swap of one to three customers in different routes, 2-OPT, 2-OPT*, empty-route, swap depot...
  - Multiple shaking operators: multi-swap, multi-shift, double-bridge...
Set covering model to create new solutions out of a set of high-quality routes. Adaptation of the pool size.

Set covering model to create new solutions out of a set of high-quality routes.

\[
\begin{align*}
\text{minimize} & \quad \sum_{k \in S} d_k x_k \\
\text{subject to} & \quad \sum_{k \in S} a_{ik} x_k = 1 \quad \text{for } i = 1, \ldots, n \\
& \quad x_k = 0 \text{ or } 1 \quad \text{for } k \in S.
\end{align*}
\]

Solver for integer linear programming (Cplex)

Best solution created from these routes

Elite routes found during the search
• **First Genetic Algorithm (GA)** to achieve competitive results on some VRP variants.

• **Genetic algorithms** mimic natural evolution
  - Population of solutions
  - Selection
  - Crossover
  - Mutation (replaced here by a local search)
The algorithm of Prins (2004) includes a few important “tricks”:

- Giant-tour solution representation
  - Polynomial Split algorithm to obtain a complete solution
- Simple Crossover
- Local search on the offspring
- Population management (spacing constraint)
• Classic hybrid genetic algorithm with:
  ▶ Giant-tour solution representation in the crossover: the same as Prins (2004)
  ▶ Efficient local search
  ▶ Management of penalized infeasible solutions
  ▶ Promotion of diversity in the population: Biased fitness

• Easily generalizable ⇒ Applied to over 50 VRP variants
**Biased fitness:** combining ranks in terms of solution cost $C(I)$ and contribution to the population diversity $D(I)$, measured as a distance to other individuals:

$$BF(I) = C(I) + \left(1 - \frac{\text{nbElite}}{\text{popSize} - 1}\right) D(I)$$

- Used for parents selection
  - Balancing quality with innovation to promote a more thorough exploration of the search space.
- Used during selection of survivors
  - Removing individuals with worst $BF(I)$ still guarantees elitism in terms of solution quality.
• Based on a Guided local search
  ▶ Detect and temporarily penalize bad edges
  ▶ Characterization of bad edges result from a prior study of defining features of good and bad solutions

• Three efficient types of neighborhoods
  ▶ CROSS-exchanges
  ▶ Ejection chains
  ▶ Heuristic of Lin and Kernighan (1973)

• Leading to an efficient and fast method for the CVRP

[SHOW EXAMPLE 3. ALGO. ARNOLD & SORENSEN]
SISRs – Christiaens and Vanden Berghe (2018)

• Based on the ruin-and-recreate principle
• One ruin operator (adjacent string removals)
  ▶ Aims to introduce capacity and spatial slack
• One recreate method (greedy insertion with blinks)
  ▶ Skipping best insertions in a controlled manner

• Excellent results on the new large-scale CVRP instances of Uchoa et al. (2017)
• State-of-the-art results for multiple problem variants: CVRP, VRPTW, PDPTW...
• A very large variety of metaheuristics have been developed.
• Many existing concepts and methods... and many open questions:
  • Why using a strategy or a metaheuristic of a given type
  • For which problems some strategies are most successful
• Method: tradeoff between solution quality, speed, ability to generalize, robustness and simplicity
1 Multi-attribute Vehicle Routing Problems

2 Heuristics and Metaheuristics
   • A quick guided tour of CVRP metaheuristics
   • Successful strategies – Analysis

3 Unified Hybrid Genetic Search
   • General description
   • Tricks of the trade

4 Structural Problem Decompositions
   • Arc Routing Problems
   • Team-Orienteering Problems
   • CVRP – Sequence or Set optimization?

5 Conclusions and Perspectives
Successful strategies – Analysis

• Vidal et al. (2013b): analysis of solution concepts for multiple VRP variants

• Protocole:
  ▶ Selection of 14 multi-attribute VRPs – Criterion : classic benchmark instances available + large number of heuristics).
  ▶ Identification of the top 3 to 5 best metaheuristics for each problem
  ▶ Analysis of the resulting 64 methods, to pinpoint successful methodological elements.
Successful strategies – Analysis

- 19 aspects of the methods have been examined:

<table>
<thead>
<tr>
<th>Search space</th>
<th>1) presence of infeasible solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2) use of indirect representations of solutions</td>
</tr>
<tr>
<td>Neighbourhoods</td>
<td>3) presence of multiple neighbourhoods</td>
</tr>
<tr>
<td></td>
<td>4) use of polynomially enumerable neighbourhoods</td>
</tr>
<tr>
<td></td>
<td>5) use of pruning procedures</td>
</tr>
<tr>
<td></td>
<td>6) use of large neighbourhoods</td>
</tr>
<tr>
<td></td>
<td>7) use of solution recombinations</td>
</tr>
<tr>
<td>Trajectory</td>
<td>8) presence of random components</td>
</tr>
<tr>
<td></td>
<td>9) continuous aspect of trajectories</td>
</tr>
<tr>
<td></td>
<td>10) discontinuous aspect</td>
</tr>
<tr>
<td></td>
<td>11) mixed aspect</td>
</tr>
<tr>
<td>Control and memories</td>
<td>12) use of populations</td>
</tr>
<tr>
<td></td>
<td>13) diversity management</td>
</tr>
<tr>
<td></td>
<td>14) parameter adaptation</td>
</tr>
<tr>
<td></td>
<td>15) advanced guidance mechanisms</td>
</tr>
<tr>
<td>Hybrid strategies</td>
<td>16) use of hybridization</td>
</tr>
<tr>
<td></td>
<td>17) matheuristics with integer programming</td>
</tr>
<tr>
<td>Parallelism</td>
<td>18) use of parallelism or cooperation concepts</td>
</tr>
<tr>
<td>Problem decompositions</td>
<td>19) use of problem decompositions</td>
</tr>
</tbody>
</table>
• 19 aspects of the methods have been examined:

<table>
<thead>
<tr>
<th>Methodology</th>
<th>SP.</th>
<th>NEIGHBOUR.</th>
<th>TRAJEC.</th>
<th>CONTROL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>BACKHAULS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brandão (2006)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Ropke and Pisinger (2006a)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Gajpal and Abad (2009)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Zachariadis and K. (2012)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PICK-UP AND DELIVERIES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ropke and Pisinger (2006b)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cordeau and Laporte (2003)</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Parragh et al. (2010)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Nagata and Kobayashi (2011)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>MULTIPLE TRIPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taillard et al. (1996)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Olivera and Viera (2007)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Alonso et al. (2008)</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>TIME WINDOWS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hashimoto et Y. (2008)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Repoussis et al. (2009)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P.-Gagnon et al. (2009)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Nagata et al. (2010)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

© Vidal Thibaut 2012
• **Search Space: Using infeasible solutions (31/64 methods).**

• Usual to relax route constraints and penalize violations: capacity, duration, TW...

• Enables to transition more easily in the search space between feasible solutions.

• Strategic oscillation (Glover, 1986): good solutions are known to be close to the borders of feasibility – Oscillating close to these borders by adapting the penalty coefficients.
Successful strategies – Analysis

- The space of feasible solutions is often...

- On problems with tight constraints, infeasible solutions are necessary to transition from one solution to another
Successful strategies – Analysis

- Experimental analysis of TW relaxations in Vidal et al. (2015a).
  - Solomon VRPTW instances – simple LS-improvement procedure.

<table>
<thead>
<tr>
<th>Inst.</th>
<th>Soll1</th>
<th>NoUnf</th>
<th>Late</th>
<th>Twarp</th>
<th>Flex</th>
<th>Soll1 Initial Solution</th>
<th>NoUnf</th>
<th>Late</th>
<th>Twarp</th>
<th>Flex</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>1431.97</td>
<td>1225.25</td>
<td>1219.11</td>
<td>1220.13</td>
<td><strong>1219.41</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>1326.64</td>
<td>963.53</td>
<td>957.11</td>
<td>947.77</td>
<td>942.87</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>936.48</td>
<td>844.00</td>
<td>835.17</td>
<td>835.67</td>
<td><strong>834.26</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td>696.57</td>
<td>605.62</td>
<td>603.62</td>
<td>603.08</td>
<td>600.59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RC1</td>
<td>1578.28</td>
<td>1401.49</td>
<td>1399.11</td>
<td>1389.83</td>
<td>1396.54</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RC2</td>
<td>1653.61</td>
<td>1139.12</td>
<td>1077.93</td>
<td>1093.02</td>
<td>1079.40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTD</td>
<td>71633</td>
<td>58067</td>
<td>57319</td>
<td>57275</td>
<td><strong>57125</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T(sec)</td>
<td>0.03</td>
<td>3.41</td>
<td>20.06</td>
<td>6.59</td>
<td>7.90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Similar conclusions regarding distance and load relaxations on CVRP, PVRP and MDVRP (Vidal et al 2012).
Successful strategies – Analysis

- **Search Space: Indirect solution representations (12/64)**

- Focus the heuristic search on a subset of the decision variables
- Use a decoder to determine the rest of the decisions – Exact algorithms may be used for decoding
- Examples:
  - Giant tours without trip delimiters in Prins (2004)
  - Storing a subset of the decision sets: visit-day choices for the PVRP, sequences of visits without visit-time information...
• Neighborhoods
  ▶ All successful VRP metaheuristics use either local or large neighborhood search.
  ▶ LS neighborhoods usually contain $O(n^2)$ moves
  ▶ Ruin-and-recreate is also frequently used
  ▶ Covering at least the main families of moves (RELOCATE, SWAP, 2-OPT) is determining to achieve high-quality solutions.
  ▶ Trade-off between neighborhood size and search speed
  ▶ Optimizing all attributes of the problems (sequencing, assignment to depots, vehicles, days) is a key to success.
• Neighborhood search = bottleneck of most recent metaheuristics (but for a good reason)

• Speed-up techniques (used in 26/64 methods)
  ▶ Neighborhood restrictions: granular or sequential search
  ▶ Memories: matrices for move evaluations, hashtables for route evaluations.
  ▶ Preprocessing on subsequences to speed move evaluations in presence of complicating attributes (discussed later in this talk).
Successful strategies – Analysis

• Search Trajectory – Randomization (56/64)

► Prerequisite for some asymptotic convergence properties (e.g., SA, GA).

► A simple way of avoiding cycling and bringing more diversity.

► “an intelligent use of randomization, which is not blindly uniform but embedded in probabilities that account for history and measures of attractiveness, offers a useful type of diversification that can substitute for more complex uses of memory” (Rochat and Taillard, 1995)

► If needed, fix the seed to obtain a deterministic algorithm.

► Random order is not worse than any fixed customer-indices order obtained from the instance (often arbitrary).
• **Populations (28/64)**

• Acquisition, management, and exploitation of problem-knowledge  
  ⇒ Core function of a metaheuristics.

• Glover (1989) discern two types of memories
  ▶ Short term memories (e.g. tabu lists) – used to escape local optima
  ▶ Medium and long-term memories – guide the overall exploration

• Other forms of memories: populations to store full solutions,  
  routes or fragments of solutions, statistics on decision variables,  
  pheromones, supported patterns...
Successful strategies – Analysis

• Population management (14/28)

• A need for diverse and high-quality solutions
  ▶ Critical to avoid premature convergence. Needed to counterbalance the aggressive-improvement abilities of local search in hybrid GA.

▶ Diversity management strategies (Prins, 2004; Sörensen and Sevaux, 2006)
▶ Promotion of diversity in the objective (Vidal et al., 2012)
▶ Based on distance measures, in objective or solution space.
Successful strategies – Analysis

- **Memories and control – Population management (14/28)**
- Some experiments on this topic in Vidal et al. (2012), solution quality of HGSADC on standard PVRP, MDVRP, and MDPVRP instances:
  - HGA: No diversity management method
  - HGA-DR: Spacing rule in objective space (Prins, 2004)
  - HGA-PM: Spacing rule in solution space (Sörensen and Sevaux, 2006)
  - HGSADC: Promotion of diversity in the objective (Vidal et al., 2012)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>HGA</th>
<th>HGA-DR</th>
<th>HGA-PM</th>
<th>HGSADC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PVRP</td>
<td>6.86 min</td>
<td>7.01 min</td>
<td>7.66 min</td>
<td>8.17 min</td>
</tr>
<tr>
<td></td>
<td>+0.64%</td>
<td>+0.49%</td>
<td>+0.39%</td>
<td>+0.13%</td>
</tr>
<tr>
<td>MDVRP</td>
<td>7.93 min</td>
<td>7.58 min</td>
<td>9.03 min</td>
<td>8.56 min</td>
</tr>
<tr>
<td></td>
<td>+1.04%</td>
<td>+0.87%</td>
<td>+0.25%</td>
<td>-0.04%</td>
</tr>
<tr>
<td>MDPVRP</td>
<td>25.32 min</td>
<td>26.68 min</td>
<td>28.33 min</td>
<td>40.15 min</td>
</tr>
<tr>
<td></td>
<td>+4.80%</td>
<td>+4.07%</td>
<td>+3.60%</td>
<td>+0.44%</td>
</tr>
</tbody>
</table>
Successful strategies – Analysis

• Some experiments on this topic in Vidal et al. (2012), solution quality on standard PVRP, MDVRP, and MDPVRP instances:
  ▶ HGA: No diversity management method
  ▶ HGA-DR: Spacing rule in objective space (Prins, 2004)
  ▶ HGA-PM: Spacing rule in solution space (Sörensen and Sevaux, 2006)
  ▶ HGSADC: Promotion of diversity in the objective (Vidal et al., 2012)
Successful strategies – Analysis

• Search Guidance

• A very simple form of guidance: parameters adaptation (30/64)
  ▶ Driving infeasibility penalties, mutation and crossover rates, frequency of use of some operators or strategies.

• More advanced forms of guidance: collect and analyze information on the past search to guide the method
  ▶ Collect historical statistics on solution features, arcs, sets of arcs, routes, or problem specific attributes.
  ▶ Mine supported patterns (Ribeiro et al., 2006; Santana, 2018)
Successful strategies – Analysis

• Exploitation of guidance information:
  ▶ Guidance actions to
    • Either **intensify** the search around promising solution features
    • Or **diversify** the search around promising unexplored areas.
  ▶ Applying penalties or incentives on solution features
  ▶ Restarts from elite solutions or fragments of solutions
  ▶ Target solutions in path relinking
  ▶ Neighborhood choice driven by pheromone matrices in ACO

• Continuous through the search, or punctually through a purposeful move
Successful strategies – Analysis

• **Hybridizations (39/64)**
  - Combining features of several methods
  - Most frequent in the heuristics surveyed: GA+LS, ACO+LS or ACO+LNS, Tabu + recombinations, ILS + VNS...

• Generally speaking, metaheuristics are inherently hybrids, described sometimes as “heuristics that guide other heuristics”

• **Matheuristics (9/64)**, blending metaheuristics with math. programming components:
  - Handling problem-attributes (e.g. loading constraints or split deliveries)
  - Exploring large neighborhoods
  - Recombining solution elements...
• Decompositions

  ▶ Help to focus on subsets of decision variables: useful as an intensification procedure or to deal with large-scale problems
  ▶ Some examples:
• Decompositions

▶ Help to focus on subsets of decision variables: useful as an intensification procedure or to deal with large-scale problems

▶ Some examples:
Some conclusions

- Success is not related to a single feature but rather to a combination of concepts
- Several search strategies are combined to achieve a good balance between search intensification and diversification
- Well-designed LS or LNS-based improvement methods are essential to refine the solutions
  - Computational complexity. Do not confound search and enumeration (Bentley and Friedman, 1978; Bentley, 1992; De Berg et al., 2018)
  - Preprocessing, memories, neighborhood restrictions...
Some conclusions

• Many components can contribute to increase performance
  ⇒ One can always improve a method by “adding more”...
  ⇒ Success comes from a good tradeoff between performance and simplicity.
  ⇒ To gain methodological insights, need to trim off all unnecessary component and avoid complex methodologies with only marginal contributions to performance.
  ⇒ Computational experiments to assess the impact of each separate component
Some References

References:


Contents

1. Multi-attribute Vehicle Routing Problems

2. Heuristics and Metaheuristics
   - A quick guided tour of CVRP metaheuristics
   - Successful strategies – Analysis

3. Unified Hybrid Genetic Search
   - General description
   - Tricks of the trade

4. Structural Problem Decompositions
   - Arc Routing Problems
   - Team-Orienteering Problems
   - CVRP – Sequence or Set optimization?

5. Conclusions and Perspectives
Unified Hybrid Genetic Search

- Unified Hybrid Genetic Search (UHGS): aiming to address the challenges related to the variety of problem combinations (Vidal et al., 2014)

- Hybrid genetic search with Advanced Diversity Control (HGA):
  - Hybrid genetic Algorithm
  - Well-designed selection and crossover operators
  - High-performance local-improvement procedure ("education")
  - Management of penalized infeasible solutions in two subpopulations
  - Diversity & Cost objective for individuals evaluations
• The method relies on assignment, sequencing & route evaluation operators, which are selected and combined by the method relatively to the problem structure (using polymorphism and inheritance), to perform the assignment, sequencing and route evaluations.
Unified Hybrid Genetic Search

**Components:**
- **ROUTE EVALUATION COMPONENT**
- **ASSIGNMENT COMPONENT**
- **SEQUENCING COMPONENT**
- **ROUTE EVALUATION COMPONENT**

**Population with Diversity Management**
- Penalties adaptation
- Survivors’ selection
- Diversification and decomposition phases

**Selection**
- Binary Tournament Based on COST & DIVERSITY

**AIX Crossover**
- Split
  - Placement of depot occurrences for each resource

**Merge**
- Removal of trip delimiters

**Education and Repair**
- Based on the unified local search

**Initialize Population**

[if terminated]

[if not terminated]

Return Best Solution

**Problem Metaheuristics UHGS Decompositions Conclusions References**
• One **important structural property of local searches** helps to progress towards **unified** and **efficient** metaheuristics:

  ▶ Vidal et al. (2015b): Any LS move involving a bounded number of node relocations or arc exchanges can be assimilated to a recombination of a bounded number of consecutive visit sequences from the incumbent solution.
Unified Hybrid Genetic Search

- Data preprocessing: compute auxiliary data on subsequences to speed up the search
- Evaluate moves by induction on the concatenation operator ($\oplus$)

**Easy to adapt:**
- Define all moves based on the concatenation operators
- To deal with multiple problems: adapt the preprocessing and concatenation operators
• Example 1) Distance and capacity constraints

**Auxiliary data structures:**
Partial loads $Q(\sigma)$ and distance $D(\sigma)$

**Initialization**
For a sequence $\sigma_0$ with a single visit $v_i$, $Q(\sigma_0) = q_i$ and $D(\sigma_0) = 0$

**Induction Step:**
\begin{align*}
Q(\sigma_1 \oplus \sigma_2) &= Q(\sigma_1) + Q(\sigma_2) \\
D(\sigma_1 \oplus \sigma_2) &= D(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} + D(\sigma_2)
\end{align*}
Unified Hybrid Genetic Search

- Example 2) Objectives based on cumulated arrival time objectives

**Auxiliary data structures in use:**

Travel time $D(\sigma)$, Cumulated arrival time $C(\sigma)$, Delay Cost $W(\sigma)$ associated to one unit of delay in starting time

**Initialization**

For a sequence $\sigma_0$ with a single visit $v_i$, $D(\sigma_0) = 0$ and $C(\sigma_0) = 0$, and $W(\sigma_0) = 1$ if $v_i$ is a customer, and $W(\sigma_0) = 0$ if $v_i$ is a depot visit.

**Induction Step:**

\[
D(\sigma_1 \oplus \sigma_2) = D(\sigma_1) + d_{\sigma_1(\|\sigma_1\|)}\sigma_2(1) + D(\sigma_2)
\]

\[
C(\sigma_1 \oplus \sigma_2) = C(\sigma_1) + W(\sigma_2)(D(\sigma_1) + d_{\sigma_1(\|\sigma_1\|)}\sigma_2(1)) + C(\sigma_2)
\]

\[
W(\sigma_1 \oplus \sigma_2) = W(\sigma_1) + W(\sigma_2)
\]
Unified Hybrid Genetic Search

- Example 3) Time windows (only feasibility check, see Vidal et al. 2013a for similar equations with penalized infeasibility)

**Auxiliary data structures in use:**

Travel time and service time $T(\sigma)$, earliest feasible completion time $E(\sigma)$, latest feasible starting date $L(\sigma)$, statement of feasibility $F(\sigma)$.

**Initialization:**

For a sequence $\sigma_0$ with a single visit $v_i$, $T(\sigma_0) = s_i$, $E(\sigma_0) = e_i + s_i$, $L(\sigma_0) = l_i$ and $F(\sigma_0) = true$.

**Induction Step:**

\[
T(\sigma_1 \oplus \sigma_2) = T(\sigma_1) + d_{\sigma_1(1)} + T(\sigma_2)
\]

\[
E(\sigma_1 \oplus \sigma_2) = \max\{E(\sigma_1) + d_{\sigma_1(1)} + T(\sigma_2), E(\sigma_2)\}
\]

\[
L(\sigma_1 \oplus \sigma_2) = \min\{L(\sigma_1), L(\sigma_2) - d_{\sigma_1(1)} - T(\sigma_1)\}
\]

\[
F(\sigma_1 \oplus \sigma_2) \equiv F(\sigma_1) \land F(\sigma_2) \land (E(\sigma_1) + d_{\sigma_1(1)} \leq L(\sigma_2))
\]
• Example 4) Clustered VRP (CluVRP)

• **Cluster constraint:** All visits of each cluster need to be **consecutive and in the same route**
Example 4) Clustered VRP (CluVRP)

We can work on solutions as sequences of clusters

⇒ From the heuristic point of view, a solution looks like this:
Unified Hybrid Genetic Search

- Example 4) Clustered VRP (CluVRP)
- We can work on solutions as sequences of clusters
  ⇒ For route evaluation operator, it’s a shortest path subproblem:

![Diagram of route representation in UHGS]

- Like this, a route evaluation would be in $O(n)$, assuming that the number of customers in a cluster is bounded by a constant
  ▶ Difficult to evaluate many solutions, need to do better.
• Example 4) Clustered VRP (CluVRP)
• Idea: Store shortest paths on partial sequences.
  ⇒ To evaluate a move, solve a shortest path sub-problem with only $O(1)$ arcs:
Example 4) Clustered VRP (CluVRP)

**Auxiliary data**

Shortest path $S(\sigma)[i,j]$ inside sequence $\sigma$ starting at the location $i$ of the starting cluster and finishing at location $j$ of the ending cluster.

**Initialization**

For $\sigma_0$ with a single visit $v_i$, $S(\sigma_0)[i,j] = \begin{cases} +\infty & \text{if } i = j \\ \hat{c}_{ij} & \text{if } i \neq j \end{cases}$

**Induction step**

By induction on the concatenation operator:

$$S(\sigma_1 \oplus \sigma_2)[i,j] = \min_{1 \leq x \leq \lambda_{\sigma_1(|\sigma_1|)}, 1 \leq y \leq \lambda_{\sigma_2(1)}} S(\sigma_1)[i,x] + c_{xy} + S(\sigma_2)[y,j]$$

$\forall i \in \{1, \ldots, \lambda_{\sigma_1(1)}\}, \forall j \in \{1, \ldots, \lambda_{\sigma_2(|\sigma_2|)}\}$
• Solution representation and Split:

<table>
<thead>
<tr>
<th>1 2 3 4 5 6</th>
<th>1 2 7 8</th>
<th>1 3 4 6 7 8</th>
</tr>
</thead>
</table>

- Assignment of customer services to some ASSIGN attributes (AARs) + separate optimization of routes for each AAR.
- Solution representation is designed accordingly.
- Furthermore, representation without trip delimiters for each AAR.

Giant Tour Representation

Routes of a Solution
• Crossover operator:

![Diagram of crossover operator with steps and parent representations](image-url)
Biased fitness: combining ranks in terms of solution cost $C(I)$ and contribution to the population diversity $D(I)$, measured as a distance to other individuals:

$$BF(I) = C(I) + \left(1 - \frac{nbElite}{popSize - 1}\right) D(I)$$

- Used for parents selection
  - Balancing quality with innovation to promote a more thorough exploration of the search space.
- Used during selection of survivors
  - Removing individuals with worst $BF(I)$ still guarantees elitism
• **Computational Experiments:** UHGS has been tested on more than 2000 benchmark instances, and 50 different problems from the vehicle routing literature

• The method has been compared to over 240 previous algorithms

  ▶ State-of-the-art results in the literature on all considered problems: VRP with capacity constraints, duration, backhauls, asymmetry, cumulative costs, simultaneous and mix pickup and deliveries, fleet mix, load dependency, multiple periods, depots, generalized deliveries, open routes, time windows, time-dependent travel time and costs, soft and multiple TW, truck driver scheduling regulations, many other problems and their combinations...

  ▶ First method which addresses efficiently many problems and their combinations, equals or outperforms all available methods from the literature.
## Unified Hybrid Genetic Search

### Problem Metaheuristics UHGS Decompositions Conclusions References

<table>
<thead>
<tr>
<th>Variant</th>
<th>Bench.</th>
<th>$n$</th>
<th>Obj.</th>
<th>State-of-the-art methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVRP</td>
<td>CMT79</td>
<td>[50,199]</td>
<td>C</td>
<td>GG11: +0.03% 8×2.38 8×Xe 2.3G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MB07: +0.02% 2.80 P-IV 2.8G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS*: +0.02% 11.90 Opt 2.4G</td>
</tr>
<tr>
<td>CVRP</td>
<td>GWKC98</td>
<td>[200,483]</td>
<td>C</td>
<td>GG11: +0.29% 8×5 8×Xe 2.3G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NB09: +0.16% 21.51 Opt 2.4G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS*: +0.02% 71.41 Opt 2.4G</td>
</tr>
<tr>
<td>VRPB</td>
<td>GJ89</td>
<td>[25,200]</td>
<td>C</td>
<td>ZK12: +0.38% 1.09 T5500 1.67G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GA09: +0.09% 1.13 Xe 2.4G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS: +0.01% 0.99 Opt 2.4G</td>
</tr>
<tr>
<td>CCVRP</td>
<td>CMT79</td>
<td>[50,199]</td>
<td>C</td>
<td>NPW10: +0.74% 5.20 Core2 2G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RL12: +0.37% 2.69 Core2 2G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS: +0.01% 1.42 Opt 2.2G</td>
</tr>
<tr>
<td>CCVRP</td>
<td>GWKC98</td>
<td>[200,483]</td>
<td>C</td>
<td>NPW10: +2.03% 94.13 Core2 2G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RL12: +0.34% 21.11 Core2 2G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS: -0.14% 17.16 Opt 2.2G</td>
</tr>
<tr>
<td>VRPSDP</td>
<td>SN99</td>
<td>[50,199]</td>
<td>C</td>
<td>SDBOF10: +0.16% 256×0.37 256×Xe 2.67G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ZTK10: +0.11% T5500 1.66G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS: +0.01% 2.79 Opt 2.4G</td>
</tr>
<tr>
<td>VRPSDP</td>
<td>MG06</td>
<td>[100,400]</td>
<td>C</td>
<td>SDBOF10: +0.30% 256×3.11 256×Xe 2.67G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS: +0.20% 12.00 Opt 2.4G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S12: +0.08% 7.23 I7 2.93G</td>
</tr>
</tbody>
</table>
## Unified Hybrid Genetic Search

<table>
<thead>
<tr>
<th>Variant</th>
<th>Bench.</th>
<th>(n)</th>
<th>Obj.</th>
<th>(\text{State-of-the-art methods})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Author</td>
</tr>
<tr>
<td>VFMP-F</td>
<td>G84</td>
<td>[20,100]</td>
<td>C</td>
<td>ISW09:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SPUO12:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS:</td>
</tr>
<tr>
<td>VFMP-V</td>
<td>G84</td>
<td>[20,100]</td>
<td>C</td>
<td>ISW09:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SPUO12:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS:</td>
</tr>
<tr>
<td>VFMP-FV</td>
<td>G84</td>
<td>[20,100]</td>
<td>C</td>
<td>P09:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SPUO12:</td>
</tr>
<tr>
<td>LDVRP</td>
<td>CMT79</td>
<td>[50,199]</td>
<td>C</td>
<td>XZKX12:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS:</td>
</tr>
<tr>
<td>LDVRP</td>
<td>GKW98</td>
<td>[200,483]</td>
<td>C</td>
<td>XZKX12:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS:</td>
</tr>
<tr>
<td>PVRP</td>
<td>CGL97</td>
<td>[50,417]</td>
<td>C</td>
<td>HDH09:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS*:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CM12:</td>
</tr>
<tr>
<td>MDVRP</td>
<td>CGL97</td>
<td>[50,288]</td>
<td>C</td>
<td>CM12:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S12:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS*:</td>
</tr>
<tr>
<td>GVRP</td>
<td>B11</td>
<td>[16,262]</td>
<td>C</td>
<td>BER11:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MCR12:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS:</td>
</tr>
<tr>
<td>Variant</td>
<td>Bench.</td>
<td>$n$</td>
<td>Obj.</td>
<td>State-of-the-art methods</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>--------</td>
<td>------</td>
<td>---------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>OVRP</strong></td>
<td>CMT79 &amp; F94 [50,199]</td>
<td>F/C</td>
<td>RTBI10: 0%/+0.32%</td>
<td>S12: —/+0.16%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ZK10: 0%/+0.39%</td>
<td>S12: 0%/+0.13%</td>
</tr>
<tr>
<td><strong>OVRP</strong></td>
<td>GWKC98 [200,480]</td>
<td>F/C</td>
<td>RTI09: 0%/+0.11%</td>
<td>UHGS*: 0%/+0.04%</td>
</tr>
<tr>
<td><strong>VRPTW</strong></td>
<td>SD88 100</td>
<td>F/C</td>
<td>RTI09b: —</td>
<td>NBD10: +0.20%/+0.42%</td>
</tr>
<tr>
<td><strong>VRPTW</strong></td>
<td>HG99 [200,1000]</td>
<td>F/C</td>
<td>RTI09a: +0.89%/+0.42%</td>
<td>KTDHS12: 0%/+0.79%</td>
</tr>
<tr>
<td><strong>OVRPTW</strong></td>
<td>SD88 100</td>
<td>F/C</td>
<td>KTDHS12: +2.25%</td>
<td>UHGS: -3.31%</td>
</tr>
<tr>
<td><strong>TDVRPTW</strong></td>
<td>SD88 100</td>
<td>F/C</td>
<td>BDHMG08: —</td>
<td>BDHMG08: —</td>
</tr>
<tr>
<td><strong>VFMPTW</strong></td>
<td>LS99 100</td>
<td>D</td>
<td>BDHMG08: —</td>
<td>BDHMG08: —</td>
</tr>
<tr>
<td><strong>VFMPTW</strong></td>
<td>LS99 100</td>
<td>C</td>
<td>BDHMG08: —</td>
<td>BDHMG08: —</td>
</tr>
</tbody>
</table>
## Unified Hybrid Genetic Search

<table>
<thead>
<tr>
<th>Variant</th>
<th>Bench.</th>
<th>$n$</th>
<th>Obj.</th>
<th>State-of-the-art methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Author</td>
</tr>
<tr>
<td>CVRP</td>
<td>CMT79</td>
<td>[50,199]</td>
<td>C</td>
<td>GG11:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MB07:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS*:</td>
</tr>
<tr>
<td>CVRP</td>
<td>GWKC98</td>
<td>[200,483]</td>
<td>C</td>
<td>GG11:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NB09:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS*:</td>
</tr>
<tr>
<td>VRPB</td>
<td>GJ89</td>
<td>[25,200]</td>
<td>C</td>
<td>ZK12:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GA09:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS:</td>
</tr>
<tr>
<td>CCVRP</td>
<td>CMT79</td>
<td>[50,199]</td>
<td>C</td>
<td>NPW10:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RL12:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS:</td>
</tr>
<tr>
<td>CCVRP</td>
<td>GWKC98</td>
<td>[200,483]</td>
<td>C</td>
<td>NPW10:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RL12:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS:</td>
</tr>
<tr>
<td>VRPSDP</td>
<td>SN99</td>
<td>[50,199]</td>
<td>C</td>
<td>SDBOF10:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ZTK10:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS:</td>
</tr>
<tr>
<td>VRPSDP</td>
<td>MG06</td>
<td>[100,400]</td>
<td>C</td>
<td>SDBOF10:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UHGS:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S12:</td>
</tr>
</tbody>
</table>
A few lessons learned through the years:

- **Neighborhood restrictions** or granular search – some “better ways” (Vidal et al., 2013a; Schneider et al., 2017)

- When applicable: consider the `RELOCATE`, `SWAP`, `2-OPT`, `2-OPT*` moves as a single neighborhood – don’t evaluate in different phases (Vidal et al., 2014)

- Eliminate useless move re-evaluations: remember *when* a route was last modified and *when* a move was last tested (Vidal et al., 2014; Homsi et al., 2018)

- Hash memories can help (Goel and Vidal, 2014; Toffolo et al., 2018)

- **Move lower bounds** – multi-phase evaluations for harder problems (Vidal, 2017)
References:


Contents

1 Multi-attribute Vehicle Routing Problems

2 Heuristics and Metaheuristics
   • A quick guided tour of CVRP metaheuristics
   • Successful strategies – Analysis

3 Unified Hybrid Genetic Search
   • General description
   • Tricks of the trade

4 Structural Problem Decompositions
   • Arc Routing Problems
   • Team-Orienteering Problems
   • CVRP – Sequence or Set optimization?

5 Conclusions and Perspectives
• Studying some other problems from the perspective of structural problem decomposition:

Efficient exact methods, such as bi-directional dynamic programming or integer programming on restricted formulations → used to derive other decisions

Difficult combinatorial optimization problem with several families of decisions

Heuristic search, e.g., local search on a decision set
Contents

1 Multi-attribute Vehicle Routing Problems

2 Heuristics and Metaheuristics
   - A quick guided tour of CVRP metaheuristics
   - Successful strategies – Analysis

3 Unified Hybrid Genetic Search
   - General description
   - Tricks of the trade

4 Structural Problem Decompositions
   - Arc Routing Problems
   - Team-Orienteering Problems
   - CVRP – Sequence or Set optimization?

5 Conclusions and Perspectives
Challenges

• Arc routing for home delivery, snow plowing, refuse collection, postal services, among others.

• Lead to additional challenges:

  ⇒ *Deciding* on travel directions for services on edges

  ⇒ Shortest path between services are *conditioned* by service orientations (may also need to include some additional aspects such as turn penalties or delays at intersections).
Challenges

- Arc routing for home delivery, snow plowing, refuse collection, postal services, among others.

- Lead to additional challenges:
  - Deciding on travel directions for services on edges
  - Shortest path between services are *conditioned* by service orientations (may also need to include some additional aspects such as turn penalties or delays at intersections).
A question of neighborhood

- Most recent CARP heuristics rely on several enumerative neighborhood classes to optimize assignment, sequencing and service orientation decisions
  - See, e.g. Brandão and Eglese (2008); Usberti et al. (2013); Dell’Amico et al. (2016)...
  - Shortest paths between node extremities have been pre-processed
  - Three decision classes are heuristically addressed

⇒ This is, however, not the only option.
A question of neighborhood

- Most recent CARP heuristics rely on several enumerative neighborhood classes to optimize assignment, sequencing and service orientation decisions
  - See, e.g. Brandão and Eglese (2008); Usberti et al. (2013); Dell’Amico et al. (2016)...
  - Shortest paths between node extremities have been pre-processed
  - Three decision classes are heuristically addressed

⇒ This is, however, not the only option.
A question of neighborhood

- Most recent CARP heuristics rely on several enumerative neighborhood classes to optimize assignment, sequencing and service orientation decisions
  - See, e.g. Brandão and Eglese (2008); Usberti et al. (2013); Dell’Amico et al. (2016)...
  - Shortest paths between node extremities have been pre-processed
  - Three decision classes are heuristically addressed

⇒ This is, however, not the only option.
A question of neighborhood

- In Beullens et al. (2003) and Muyldermans et al. (2005), $O(n)$ dynamic-programming based optimization of service orientations:
  - Combined in Irnich (2008) with the neighborhood of Balas and Simonetti (2001), leading to promising results on mail delivery applications.
A question of neighborhood

• In Beullens et al. (2003) and Muyldermans et al. (2005), $O(n)$ dynamic-programming based optimization of service orientations:

• Combined in Irnich (2008) with the neighborhood of Balas and Simonetti (2001), leading to promising results on mail delivery applications.
A question of neighborhood

- Transferring several decision classes into exact dynamic-programming based components.
- This is a structural problem decomposition:

Efficient exact methods, such as bidirectional dynamic programming or integer programming on restricted formulations → used to derive other decisions

Heuristic search, e.g., local search on a decision set
SOLUTION AS PERMUTATIONS OF SERVICES

DECODING in O(1)

OPTIMAL EVALUATION OF SERVICE ORIENTATIONS AND INTERMEDIATE PATHS

Difficult combinatorial optimization problem with several families of decisions
Solution representation and decoding

• How to decode/evaluate a solution = deriving optimal orientations for the services?

⇒ Simple dynamic programming subproblem (Beullens et al., 2003; Wøhlk, 2003, 2004):

Solution Representation:

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\text{Depot} & \sigma(1) & \sigma(2) & \sigma(3) & \sigma(4) & \sigma(5) & \text{Depot} \\
\hline
\end{array}
\]

Shortest Path Problem:

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\sigma(1)^1 & C_{\sigma(1),\sigma(2)}^1 & S_{\sigma(2)} & \sigma(3)^2 & \sigma(4)^2 & \sigma(5)^2 & \text{Depot} \\
\hline
\sigma(1)^2 & C_{\sigma(1),\sigma(2)}^2 & S_{\sigma(2)} & \sigma(3)^2 & \sigma(4)^2 & \sigma(5)^2 & \text{Depot} \\
\hline
\end{array}
\]

• Each service represented by two nodes, one for each orientation. Travel costs \(c_{ij}^{kl}\) between \((i, j)\) are conditioned by the orientations \((k, l)\) for departure and arrival.
• Modern neighborhood-centered heuristics evaluate millions/billions of neighbor solutions during one run.

• Back to our key property of classical routing neighborhoods:
Seeking low complexity for solution evaluations

**Auxiliary data structures = partial shortest paths**

Partial shortest path $C(\sigma)[k, l]$ between the first and last service in the sequence $\sigma$, for any (entry, exit) direction pair $(k, l)$

### Initialization

For $\sigma_0$ with a single visit $v_i$, $S(\sigma_0)[k, l] = \begin{cases} 0 & \text{if } k = l \\ +\infty & \text{if } k \neq l \end{cases}$

### Induction Step:

By induction on the concatenation operator:

$$C(\sigma_1 \oplus \sigma_2)[k, l] = \min_{x, y} \left\{ C(\sigma_1)[k, x] + c_{\sigma_1(\sigma_1(1))\sigma_2(1)}^{xy} + C(\sigma_2)[y, l] \right\}$$
• Pre-processing partial shortest paths in the incumbent solution – in $O(n^2)$ before the neighborhood exploration – dramatically simplifies the shortest paths:

Shortest path problem:

Shortest path problem on a reduced graph, using pre-processed labels:

• Only a constant number of edges
Lower bounds on moves

• Each move evaluation was still taking a bit more operations (constant of $4\times$) than in the classic CVRP.

• Even this can be avoided...
  ⇒ by developing lower bounds on the cost of neighbors...
Lower bounds on moves

- Let \( \bar{Z}(\sigma) \) be a lower bound on the cost of a route \( \sigma \)

- A move that modifies two routes: \( \{\sigma_1, \sigma_2\} \Rightarrow \{\sigma_1', \sigma_2'\} \) has a chance to be improving if and only if:

\[
\Delta_\Pi = \bar{Z}(\sigma_1') + \bar{Z}(\sigma_2') - Z(\sigma_1) - Z(\sigma_2) < 0.
\]
Lower bounds on moves

- Let \( C_{\text{MIN}}(\sigma) = \min_{k,l} \{ C(\sigma)[k,l] \} \) the shortest path for the sequence \( \sigma \) between any pair of origin/end orientations.

- Let \( c_{ij}^{\text{MIN}} = \min_{k,l} \{ c_{ij}^{kl} \} \) be the minimum cost of a shortest path between services \( i \) and \( j \), for any orientation.

- Lower bound on the cost of a route \( \sigma = \sigma_1 \oplus \cdots \oplus \sigma_X \) composed of a concatenation of \( X \) sequences:

\[
\overline{Z}(\sigma_1 \oplus \cdots \oplus \sigma_X) = \sum_{j=1}^{X} C_{\text{MIN}}(\sigma_j) + \sum_{j=1}^{X-1} c_{\sigma_j,\sigma_{j+1}}^{\text{MIN}}.
\]

- The bound helps to filter a lot of moves (\( \geq 90\% \))
  
  In practice: possible to evaluate a move with implicit service orientations for the CARP, using roughly the same number of elementary operations as the same move for a CVRP!
Lower bounds on moves

- Let $C^\text{MIN}(\sigma) = \min_{k,l} \{ C(\sigma)[k,l] \}$ the shortest path for the sequence $\sigma$ between any pair of origin/end orientations.

- Let $c^\text{MIN}_{ij} = \min_{k,l} \{ c_{ij}^{kl} \}$ be the minimum cost of a shortest path between services $i$ and $j$, for any orientation.

- Lower bound on the cost of a route $\sigma = \sigma_1 \oplus \cdots \oplus \sigma_X$ composed of a concatenation of $X$ sequences:

$$
\bar{Z}(\sigma_1 \oplus \cdots \oplus \sigma_X) = \sum_{j=1}^{X} C^\text{MIN}(\sigma_j) + \sum_{j=1}^{X-1} c^\text{MIN}_{\sigma_j,\sigma_{j+1}}.
$$

- The bound helps to filter a lot of moves ($\geq 90\%$)

  - In practice: possible to evaluate a move with implicit service orientations for the CARP, using roughly the same number of elementary operations as the same move for a CVRP!
### Experimental setting

- Initial experiments on CARP and MCGRP
- Literature on CARP and MCGRP built around several sets of well-known benchmark instances:

|    | Reference | $|N_R|$ | $|E_R|$ | $|A_R|$ | $n$ | Specificities |
|----|-----------|-------|-------|-------|-----|----------------|
| **CARP:** |           |       |       |       |     |                |
| VAL | (34) Benavent et al. (1992) | 0     | [39,97] | 0     | [39,97] | Random graphs; Only required edges |
| BMCV | (100) Beullens et al. (2003) | 0     | [28,121] | 0     | [28,121] | Intercity road network in Flanders |
| EGL | (24) Li and Eglese (1996) | 0     | [51,190] | 0     | [51,190] | Winter-gritting application in Lancashire |

| **MCGRP:** | | |
| MGGDB | (138) Bosco et al. (2012) | [3,16] | [1,9] | [4,31] | [8,48] | From CARP instances GBD |
| MGVAL | (210) Bosco et al. (2012) | [7,46] | [6,33] | [12,79] | [36,129] | From CARP instances VAL |
| CBMix | (23) Prins and B. (2005) | [0,93] | [0,94] | [0,149] | [20,212] | Randomly generated planar networks |
| BHW | (20) Bach et al. (2013) | [4,50] | [0,51] | [7,380] | [20,410] | From CARP instances GDB, VAL, & EGL |
| DI-NEARP | (24) Bach et al. (2013) | [120,347] | [120,486] | 0 | [240,833] | Newspaper and media product distribution |
For each benchmark set, we collected the best three solution methods in the literature (some are heavily tailored for specific benchmark sets).

<table>
<thead>
<tr>
<th>Problem Metaheuristics</th>
<th>UHGS Decompositions</th>
<th>Conclusions</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLMV14 Bosco et al. (2014)</td>
<td>LPR01 Lacomme et al. (2001)</td>
<td>PDHM08 Polacek et al. (2008)</td>
<td></td>
</tr>
<tr>
<td>BMCV03 Beullens et al. (2003)</td>
<td>MLY14 Mei et al. (2014)</td>
<td>TMY09 Tang et al. (2009)</td>
<td></td>
</tr>
<tr>
<td>DHDI14 Dell’Amico et al. (2016)</td>
<td>MPS13 Martinelli et al. (2013)</td>
<td>UFF13 Usberti et al. (2013)</td>
<td></td>
</tr>
</tbody>
</table>

Comparison with the proposed metaheuristics, which are searching the space of service permutations (our methods are not fine-tuned for any of these instance sets).
## Comparison with previous literature

<table>
<thead>
<tr>
<th>Variant</th>
<th>Bench.</th>
<th>n</th>
<th>Author</th>
<th>Runs</th>
<th>Avg.</th>
<th>Best</th>
<th>T</th>
<th>T*</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDB</td>
<td>[11,55]</td>
<td></td>
<td>TMY09</td>
<td>30</td>
<td>0.009%</td>
<td>0.000%</td>
<td>0.11</td>
<td>—</td>
<td>Xe 2.0G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BMCV03</td>
<td>1</td>
<td>0.000%</td>
<td>—</td>
<td>0.03</td>
<td>P-II 500M</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MTY09</td>
<td>1</td>
<td>0.000%</td>
<td>—</td>
<td>0.01</td>
<td>Xe 2.0G</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILS</td>
<td>10</td>
<td>0.002%</td>
<td>0.000%</td>
<td>0.16</td>
<td>0.03</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UHGS</td>
<td>10</td>
<td>0.000%</td>
<td>0.000%</td>
<td>0.22</td>
<td>0.01</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td>VAL</td>
<td>[39,97]</td>
<td></td>
<td>MTY09</td>
<td>1</td>
<td>0.142%</td>
<td>—</td>
<td>0.11</td>
<td>Xe 2.0G</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LPR01</td>
<td>1</td>
<td>0.126%</td>
<td>—</td>
<td>2.00</td>
<td>P-III 500M</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BMCV03</td>
<td>1</td>
<td>0.060%</td>
<td>—</td>
<td>1.36</td>
<td>P-II 500M</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILS</td>
<td>10</td>
<td>0.054%</td>
<td>0.024%</td>
<td>0.68</td>
<td>0.16</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UHGS</td>
<td>10</td>
<td>0.048%</td>
<td>0.021%</td>
<td>0.82</td>
<td>0.08</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td>CARP</td>
<td>[28,121]</td>
<td></td>
<td>BE08</td>
<td>1</td>
<td>0.156%</td>
<td>—</td>
<td>1.08</td>
<td>P-M 1.4G</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MTY09</td>
<td>1</td>
<td>0.073%</td>
<td>—</td>
<td>0.35</td>
<td>Xe 2.0G</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BMCV03</td>
<td>1</td>
<td>0.036%</td>
<td>—</td>
<td>2.57</td>
<td>P-II 450M</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILS</td>
<td>10</td>
<td>0.027%</td>
<td>0.000%</td>
<td>0.82</td>
<td>0.22</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UHGS</td>
<td>10</td>
<td>0.007%</td>
<td>0.000%</td>
<td>0.87</td>
<td>0.11</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td>BMCV</td>
<td>[51,190]</td>
<td></td>
<td>PDHM08</td>
<td>10</td>
<td>0.624%</td>
<td>—</td>
<td>30.0</td>
<td>8.39</td>
<td>P-IV 3.6G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UFF13</td>
<td>15</td>
<td>0.560%</td>
<td>0.206%</td>
<td>13.3</td>
<td>—</td>
<td>I4 3.0G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MTY09</td>
<td>1</td>
<td>0.553%</td>
<td>—</td>
<td>2.10</td>
<td>Xe 2.0G</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILS</td>
<td>10</td>
<td>0.236%</td>
<td>0.106%</td>
<td>2.35</td>
<td>1.33</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UHGS</td>
<td>10</td>
<td>0.153%</td>
<td>0.058%</td>
<td>4.76</td>
<td>3.14</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td>EGL</td>
<td>[51,190]</td>
<td></td>
<td>BE08</td>
<td>1</td>
<td>4.679%</td>
<td>—</td>
<td>17.0</td>
<td>—</td>
<td>P-M 1.4G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MPS13</td>
<td>10</td>
<td>2.950%</td>
<td>2.523%</td>
<td>20.7</td>
<td>—</td>
<td>I5 3.2G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MLY14</td>
<td>30</td>
<td>1.603%</td>
<td>0.895%</td>
<td>33.4</td>
<td>—</td>
<td>I7 3.4G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILS</td>
<td>10</td>
<td>0.880%</td>
<td>0.598%</td>
<td>23.6</td>
<td>15.4</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UHGS</td>
<td>10</td>
<td>0.645%</td>
<td>0.237%</td>
<td>36.5</td>
<td>27.5</td>
<td>Xe 3.07G</td>
</tr>
</tbody>
</table>
## Comparison with previous literature

<table>
<thead>
<tr>
<th>Variant</th>
<th>Bench.</th>
<th>n</th>
<th>Author</th>
<th>Runs</th>
<th>Avg.</th>
<th>Best</th>
<th>T</th>
<th>T*</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCGRP</td>
<td>MGGDB</td>
<td>[8,48]</td>
<td>BLMV14</td>
<td>1</td>
<td>1.342%</td>
<td>—</td>
<td>0.31</td>
<td>—</td>
<td>Xe 3.0G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DHDI14</td>
<td>1</td>
<td>0.018%</td>
<td>—</td>
<td>60.0</td>
<td>0.86</td>
<td>CPU 3G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILS</td>
<td>10</td>
<td>0.010%</td>
<td>0.000%</td>
<td>0.13</td>
<td>0.03</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UHGS</td>
<td>10</td>
<td>0.015%</td>
<td>0.000%</td>
<td>0.16</td>
<td>0.01</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td>MCGRP</td>
<td>MGVAL</td>
<td>[36,129]</td>
<td>BLMV14</td>
<td>1</td>
<td>2.620%</td>
<td>—</td>
<td>16.7</td>
<td>—</td>
<td>Xe 3.0G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DHDI14</td>
<td>1</td>
<td>0.071%</td>
<td>—</td>
<td>60.0</td>
<td>3.69</td>
<td>CPU 3G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILS</td>
<td>10</td>
<td>0.067%</td>
<td>0.019%</td>
<td>1.18</td>
<td>0.32</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UHGS</td>
<td>10</td>
<td>0.045%</td>
<td>0.011%</td>
<td>1.20</td>
<td>0.17</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td>MCGRP</td>
<td>CBMix</td>
<td>[20,212]</td>
<td>HKSG12</td>
<td>2</td>
<td>—</td>
<td>3.076%</td>
<td>120</td>
<td>56.9</td>
<td>CPU 3G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BLMV14</td>
<td>1</td>
<td>2.697%</td>
<td>—</td>
<td>44.7</td>
<td>—</td>
<td>Xe 3.0G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DHDI14</td>
<td>1</td>
<td>0.884%</td>
<td>—</td>
<td>60.0</td>
<td>19.6</td>
<td>CPU 3G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILS</td>
<td>10</td>
<td>0.733%</td>
<td>0.363%</td>
<td>2.46</td>
<td>1.48</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UHGS</td>
<td>10</td>
<td>0.381%</td>
<td>0.109%</td>
<td>4.56</td>
<td>3.08</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td>MCGRP</td>
<td>BHW</td>
<td>[20,410]</td>
<td>HKSG12</td>
<td>2</td>
<td>—</td>
<td>1.949%</td>
<td>120</td>
<td>60.1</td>
<td>CPU 3G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DHDI14</td>
<td>1</td>
<td>0.555%</td>
<td>—</td>
<td>60.0</td>
<td>21.4</td>
<td>CPU 3G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILS</td>
<td>10</td>
<td>0.440%</td>
<td>0.196%</td>
<td>5.22</td>
<td>2.90</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UHGS</td>
<td>10</td>
<td>0.208%</td>
<td>0.077%</td>
<td>7.95</td>
<td>5.87</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td>MCGRP</td>
<td>DI-NEARP</td>
<td>[240,833]</td>
<td>HKSG12</td>
<td>2</td>
<td>—</td>
<td>1.639%</td>
<td>120</td>
<td>93.0</td>
<td>CPU 3G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DHDI14</td>
<td>1</td>
<td>0.536%</td>
<td>—</td>
<td>60.0</td>
<td>36.3</td>
<td>CPU 3G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILS</td>
<td>10</td>
<td>0.199%</td>
<td>0.084%</td>
<td>30.0</td>
<td>21.3</td>
<td>Xe 3.07G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UHGS</td>
<td>10</td>
<td>0.139%</td>
<td>0.055%</td>
<td>29.6</td>
<td>16.7</td>
<td>Xe 3.07G</td>
</tr>
</tbody>
</table>
Comparison with previous literature

- Boxplot visualizations of Gap(%) of various methods on large-scale instances:
- Gray colors indicate a significant difference of performance, as highlighted by pairwise Wilcoxon tests with adequate correction for multiplicity.

Set EGL

![Boxplot for Set EGL with comparison results]

Set EGL-L

![Boxplot for Set EGL-L with comparison results]
Comparison with previous literature

Set CBMix

HKSG12 x UHGS, P.value = 6e−05
BLMV14 x UHGS, P.value = 9e−05
DHDI14 x UHGS, P.value = 2e−04
ILS x UHGS, P.value = 0.00013

Set BHW

HKSG12 x UHGS, P.value = 0.00065
DHDI14 x UHGS, P.value = 0.00298
ILS x UHGS, P.value = 0.00233

Set DI-NEARP

HKSG12 x UHGS, P.value = 0
DHDI14 x UHGS, P.value = 7e−05
ILS x UHGS, P.value = 0.00842
Scalability

- Growth of the CPU time of UHGS as a function of the number of services, for the CARP instances (left figure) and MCGRP instances (right figure). Log-log scale.

  \[ f(n) = 0.00027 \cdot n^{1.9597} \]

- A linear fit, with a least square regression, has been performed on the sample after logarithmic transformation:

  \[ T(n) = O(n^2) \]
Real-case application

- Currently being used as the optimization core for a refuse collection application in Rio de Janeiro
  ⇒ Multiple periods, multiple trips, heterogeneous vehicle types, access restrictions, risk areas, congestion...
- 5 minute CPU time for graphs containing thousands of requests
References:

Large Neighborhoods: team-orienteeering problem

• **Team-orienteeering problem:**
  ▶ Each customer \( i \) is associated with a prize \( p_i \). Not all customers are to be serviced.
  ▶ Each route must have a distance of less than \( D \).
  ▶ The goal is to generate \( m \) feasible routes while maximizing the total amount of prizes

• Numerous applications, including:
  ▶ Logistics, third party providers, secondary market (Tricoire et al., 2010; Aksen et al., 2012)
  ▶ Humanitary relief (Campbell et al., 2008)
  ▶ Robotics, maintenance & military surveillance (Falcon et al., 2012; Mufalli et al., 2012).
**Large Neighborhoods: Team-orienteering Problem**

- Large amount of literature on TOP heuristics and metaheuristics

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Authors</th>
<th>Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGW</td>
<td>Chao et al. (1996)</td>
<td>Tabu Search</td>
</tr>
<tr>
<td>TMH</td>
<td>Tang and Miller-Hooks (2005)</td>
<td>Tabu Search</td>
</tr>
<tr>
<td>GTF</td>
<td>Archetti et al. (2007)</td>
<td>Tabu Search &amp; VNS</td>
</tr>
<tr>
<td>ASe</td>
<td>Ke et al. (2008)</td>
<td>Ant colony optimization</td>
</tr>
<tr>
<td>BDM</td>
<td>Bouly et al. (2009)</td>
<td>Memetic Algorithm</td>
</tr>
<tr>
<td>GLS</td>
<td>Vansteenwegen et al. (2009)</td>
<td>Guided Local Search</td>
</tr>
<tr>
<td>SVNS</td>
<td>Vansteenwegen and Souffriau (2009)</td>
<td>Skewed VNS</td>
</tr>
<tr>
<td>SPR</td>
<td>Souffriau et al. (2010)</td>
<td>Path Relinking</td>
</tr>
<tr>
<td>DGM</td>
<td>Dang et al. (2011)</td>
<td>Particle Swarm Optimization</td>
</tr>
<tr>
<td>MSA</td>
<td>Lin (2013)</td>
<td>Multi-Start Simulated Annealing</td>
</tr>
</tbody>
</table>

**Table: Metaheuristics for team-orienteering problems**
• **Main Idea**: always work on a full solution with all visits
  ▶ Q: How will customers be selected?
  ▶ A: Directly during separate route evaluations

• The problem of optimally selecting the customers from a complete solution can be assimilated to a shortest path with maximum profit under distance constraints for each route.

• We propose efficient techniques to solve this problem, combining
  ▶ bi-directional dynamic programming,
  ▶ graph sparsification,
  ▶ and data preprocessing techniques.
• Again a structural decomposition:

Assignment \[\rightarrow\] Sequencing

Customers Selection

HEURISTIC SEARCH
in the space of complete solutions

DYNAMIC PROGRAMMING from each complete solution
Large Neighborhoods: Team-orienteering Problem

- **Main interest:** Classic VRP neighborhoods on the complete solution representation ⇔ large neighborhoods with an exponential number of implicit insertions and removal of visits.
- **SELECT algorithm at each move** ⇔ resource-constrained SP

### Problem Metaheuristics

<table>
<thead>
<tr>
<th>i</th>
<th>$d_{0,i}$</th>
<th>$d_{i-1,i}$</th>
<th>$p_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>-</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>20</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>25</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>25</td>
<td>15</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>25</td>
<td>20</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>35</td>
<td>15</td>
</tr>
</tbody>
</table>

- $D_{max} = 100$
- $d_{7,9} = 25$
- all other distances = $+\infty$

### Sigma

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>$D(\sigma)$</th>
<th>$P(\sigma)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3,4,5,6)</td>
<td>85</td>
<td>52</td>
</tr>
<tr>
<td>(7,9,10)</td>
<td>95</td>
<td>45</td>
</tr>
<tr>
<td>(1,2,3,4)</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>(6,7,8,9)</td>
<td>90</td>
<td>57</td>
</tr>
</tbody>
</table>

- **Relocate:** 6 before 7

---

> Problem Metaheuristics UHGS Decompositions Conclusions References

112/168
Proposition

Let $B$ be an upper bound on the number of labels per node. Then, the Select algorithm is pseudo-polynomial, with a complexity of

$$O(n^2 B).$$

(4.1)

- In practice the number of labels remains very small, i.e., $B \leq 10$. 
Using a particular hierarchical cost function which considers in priority the Team-Orienteering cost (with only selected customers), and then the VRP cost with all customers.

\[ Z' = \max_{\sigma \in \mathcal{R}} \sum_{\sigma \in \mathcal{R}} Z^{\text{SELECT}}(\sigma) - \omega \sum_{\sigma \in \mathcal{R}} \sum_{i \in \{1, \ldots, |\sigma|-1\}} d_{\sigma(i)\sigma(i+1)} \]

As a consequence, when the method is unable to improve the primary objective, moves may still be performed to improve the second objective = the positions of unserviced customers.

This may lead in turn to a new repartition of customers and new opportunities of improvement of the main objective.
Large Neighborhoods: Team-orienteering Problem

• Speed-ups for move evaluations – 1. Graph Sparsification
  ▶ For a given sparsification parameter $H \in \{1, \ldots, n\}$, only the arcs $(i, j)$, with ($i < j$) satisfying Equation (4.2) are kept.

\[
j < i + H \text{ or } i = 0 \text{ or } j = |\sigma|
\] (4.2)

▶ $H$ is a sparsification parameter, usually small, e.g. $H = 3$.
▶ Thus there are only $O(Hn)$ arcs
Speed-ups for move evaluations – 1. Graph Sparsification

**Proposition**

After sparsification, the number of arcs $|A'|$ in the new graph becomes $O(nH)$, and the complexity of SELECT, in terms of number of elementary operations, is $O(nHB)$. (4.3)
Speed-ups for move evaluations – 2. Evaluation by Concatenation

For any sequence $\sigma$ of successive nodes from the incumbent solution, we propose to pre-process the following information:

**Auxiliary data structures in use:**

- Set of labels $S_{ij}(\sigma)$ associated to each resource-constrained path $(i, j)$ between any node among the $H$ first of $\sigma$, and any node among the $H$ last of $\sigma$.

- Set of labels $S^\text{END}_i(\sigma)$ associated to each resource-constrained path between any node among the $H$ first nodes of $\sigma$ and the ending depot.

- Set of labels $S^\text{BEG}_j(\sigma)$ associated to each resource-constrained path between the beginning depot and any node among the $H$ last of $\sigma$.

- Best profit $Z(\sigma)$ of a inside resource-feasible path in $\sigma$, starting from the depot, visiting a subset of customers in $\sigma$, and coming back to the depot.
Initialization and Pre-processing:

Preprocessing these values for a sequence $\sigma$ requires $O(n^2 HB)$ elementary operations.

\[
\sigma_1 \rightarrow \sigma_2 \rightarrow \sigma_3 \rightarrow \sigma_4
\]

$S_{ij}$ $S_{\text{BEG}j}$ $S_{\text{END}i}$
Initialization and Pre-processing:

Preprocessing these values for a sequence \( \sigma \) requires \( O(n^2 HB) \) elementary operations.

The resulting *reduced* multi-graph \( G'' = (V'', A'') \) is such that \( |A''| = O(MH^2) \) arcs and \( |V''| = O(MH) \) nodes. \( M \) is the number of subsequences.
Proposition (Concatenation – general)

The optimal profit $Z(\sigma_1 \oplus \cdots \oplus \sigma_M)$ of Select, for a combination of $M$ sequences is the maximum between the profit $\bar{Z}(\sigma_1 \oplus \cdots \oplus \sigma_M)$ of the resource-constrained shortest path in $G''$, and the maximum profit $Z(\sigma_i)$ of an inside resource-feasible path in $\sigma_i$ for $i \in \{1, \ldots, M\}$. Furthermore, $\bar{Z}(\sigma_1 \oplus \cdots \oplus \sigma_M)$ can be evaluated in

$$\Phi_{C-M} = O(MH^2B^2).$$

(4.4)
Proposition (Concatenation – 2 or 3 subsequences)

The optimal cost \( Z(\sigma_1 \oplus \sigma_0 \oplus \sigma_2) \) of SELECT, for a route assimilated to a recombination of three subsequences \( \sigma_1, \sigma_0 \) and \( \sigma_2 \) such that \( \sigma_0 \) contains a bounded number of customers can be evaluated using bi-directional dynamic programming for a complexity of

\[
\Phi_{C-3} = O(H^2 B). \tag{4.5}
\]

- The same complexity is achieved for a concatenation of two sequences \( \sigma_1 \) and \( \sigma_2 \).
Computational Experiments

- Experimental analysis of three heuristics and metaheuristics based on our large-neighborhood concepts
  - A simple local search (LS), restarted 100 times.
  - An Iterated Local Search (ILS), based on Prins (2009)
  - Unified Hybrid Genetic Search (UHGS) of Vidal et al. (2014)
- Benchmark instances:
  - Chao et al. (1996) for the TOP : 7 groups of instances. Groups 4-7 are the largest with 64 to 102 customers.
  - Bolduc et al. (2008) for a variant called VRP with private fleet and common carrier. These instances are derived from the CVRP instances of Christofides et al. (1979) and Golden et al. (1998).
- Tests conducted on a single Xeon 3.0GHz processor.
- Method performance evaluated relatively to Gap to Best Known Solutions BKS and CPU time.
Table: Summary of results on TOP benchmark instances

<table>
<thead>
<tr>
<th></th>
<th>CGW</th>
<th>TMH</th>
<th>GTF</th>
<th>SVF</th>
<th>ASe</th>
<th>SVNS</th>
<th>SPR</th>
<th>MSA</th>
<th>UHGS</th>
<th>ILS</th>
<th>LI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Gap 4</td>
<td>4.36%</td>
<td>1.99%</td>
<td>0.48%</td>
<td>0.06%</td>
<td>0.30%</td>
<td>1.46%</td>
<td>0.11%</td>
<td>0.06%</td>
<td>0.01%</td>
<td>0.05%</td>
<td>0.09%</td>
</tr>
<tr>
<td>Best Gap 5</td>
<td>1.36%</td>
<td>1.38%</td>
<td>0.01%</td>
<td>0.03%</td>
<td>0.04%</td>
<td>0.61%</td>
<td>0.05%</td>
<td>0.01%</td>
<td>0.00%</td>
<td>0.01%</td>
<td>0.01%</td>
</tr>
<tr>
<td>Best Gap 6</td>
<td>0.37%</td>
<td>0.79%</td>
<td>0.04%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.52%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Best Gap 7</td>
<td>2.68%</td>
<td>1.15%</td>
<td>0.29%</td>
<td>0.06%</td>
<td>0.00%</td>
<td>1.31%</td>
<td>0.04%</td>
<td>0.03%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.07%</td>
</tr>
<tr>
<td>Avg Time 4</td>
<td>796.70</td>
<td>105.30</td>
<td>22.50</td>
<td>11.40</td>
<td>32.00</td>
<td>36.70</td>
<td>367.40</td>
<td>81.00</td>
<td>298.57</td>
<td>301.54</td>
<td>76.72</td>
</tr>
<tr>
<td>Avg Time 5</td>
<td>71.30</td>
<td>69.50</td>
<td>34.20</td>
<td>3.50</td>
<td>15.10</td>
<td>11.20</td>
<td>119.90</td>
<td>6.60</td>
<td>222.92</td>
<td>193.97</td>
<td>11.31</td>
</tr>
<tr>
<td>Avg Time 6</td>
<td>45.70</td>
<td>66.30</td>
<td>8.70</td>
<td>4.30</td>
<td>14.10</td>
<td>9.00</td>
<td>89.60</td>
<td>1.40</td>
<td>184.60</td>
<td>138.25</td>
<td>6.86</td>
</tr>
<tr>
<td>Avg Time 7</td>
<td>432.60</td>
<td>160.00</td>
<td>10.30</td>
<td>12.10</td>
<td>24.60</td>
<td>27.30</td>
<td>272.80</td>
<td>32.20</td>
<td>306.35</td>
<td>309.62</td>
<td>50.22</td>
</tr>
</tbody>
</table>

- Equaled or improved 380 of the 387 best known solutions.
- 4 new BKS, quite surprising since the problems have been studied by dozens of previous papers.
### Computational Experiments

**Table: Highlight of the results on some of the most difficult problems**

<table>
<thead>
<tr>
<th>Inst</th>
<th>CGW</th>
<th>TMH</th>
<th>GTF</th>
<th>SVF</th>
<th>ASe</th>
<th>SVNS</th>
<th>SPR</th>
<th>MSA</th>
<th>UHGS</th>
<th>ILS</th>
<th>LI</th>
<th>BKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>p4.2.o</td>
<td>1147</td>
<td>1175</td>
<td>1192</td>
<td><strong>1218</strong></td>
<td>1215</td>
<td>1195</td>
<td><strong>1218</strong></td>
<td>1217</td>
<td><strong>1218</strong></td>
<td><strong>1218</strong></td>
<td><strong>1218</strong></td>
<td>1218</td>
</tr>
<tr>
<td>p4.2.p</td>
<td>1199</td>
<td>1208</td>
<td>1239</td>
<td>1241</td>
<td><strong>1242</strong></td>
<td>1237</td>
<td><strong>1242</strong></td>
<td>1241</td>
<td>1241</td>
<td>1241</td>
<td>1241</td>
<td>1242</td>
</tr>
<tr>
<td>p4.2.q</td>
<td>1242</td>
<td>1255</td>
<td>1255</td>
<td>1263</td>
<td>1263</td>
<td>1239</td>
<td>1263</td>
<td>1259</td>
<td><strong>1267</strong></td>
<td>1265</td>
<td>1265</td>
<td>1265</td>
</tr>
<tr>
<td>p4.2.r</td>
<td>1199</td>
<td>1277</td>
<td>1283</td>
<td>1285</td>
<td>1288</td>
<td>1279</td>
<td>1286</td>
<td><strong>1290</strong></td>
<td>1286</td>
<td>1281</td>
<td>1285</td>
<td>1290</td>
</tr>
<tr>
<td>p4.2.s</td>
<td>1286</td>
<td>1294</td>
<td>1299</td>
<td>1301</td>
<td><strong>1304</strong></td>
<td>1295</td>
<td>1296</td>
<td>1300</td>
<td>1302</td>
<td>1297</td>
<td>1301</td>
<td>1304</td>
</tr>
<tr>
<td>p4.2.t</td>
<td>1299</td>
<td><strong>1306</strong></td>
<td><strong>1306</strong></td>
<td><strong>1306</strong></td>
<td>1305</td>
<td><strong>1306</strong></td>
<td><strong>1306</strong></td>
<td><strong>1306</strong></td>
<td>1306</td>
<td>1306</td>
<td>1306</td>
<td>1306</td>
</tr>
<tr>
<td>p4.3.o</td>
<td>1078</td>
<td>1151</td>
<td>1157</td>
<td><strong>1172</strong></td>
<td>1170</td>
<td>1136</td>
<td>1170</td>
<td>1170</td>
<td><strong>1172</strong></td>
<td><strong>1172</strong></td>
<td>1170</td>
<td>1172</td>
</tr>
<tr>
<td>p4.3.p</td>
<td>1115</td>
<td>1218</td>
<td>1221</td>
<td><strong>1222</strong></td>
<td>1221</td>
<td>1200</td>
<td>1220</td>
<td><strong>1222</strong></td>
<td><strong>1222</strong></td>
<td><strong>1222</strong></td>
<td><strong>1222</strong></td>
<td><strong>1222</strong></td>
</tr>
<tr>
<td>p4.3.q</td>
<td>1222</td>
<td>1249</td>
<td>1241</td>
<td>1245</td>
<td>1252</td>
<td>1236</td>
<td><strong>1253</strong></td>
<td>1251</td>
<td><strong>1253</strong></td>
<td><strong>1253</strong></td>
<td>1251</td>
<td>1253</td>
</tr>
<tr>
<td>p4.3.r</td>
<td>1225</td>
<td>1265</td>
<td>1269</td>
<td>1273</td>
<td>1267</td>
<td>1250</td>
<td>1272</td>
<td>1265</td>
<td><strong>1273</strong></td>
<td>1272</td>
<td>1269</td>
<td>1272</td>
</tr>
<tr>
<td>p4.3.s</td>
<td>1239</td>
<td>1282</td>
<td>1294</td>
<td><strong>1295</strong></td>
<td>1293</td>
<td>1280</td>
<td>1287</td>
<td>1293</td>
<td><strong>1295</strong></td>
<td><strong>1295</strong></td>
<td><strong>1295</strong></td>
<td><strong>1295</strong></td>
</tr>
<tr>
<td>p4.3.t</td>
<td>1285</td>
<td>1288</td>
<td>1304</td>
<td>1304</td>
<td><strong>1305</strong></td>
<td>1299</td>
<td>1299</td>
<td>1299</td>
<td>1305</td>
<td>1305</td>
<td>1299</td>
<td>1305</td>
</tr>
<tr>
<td>p4.4.o</td>
<td>995</td>
<td>1014</td>
<td>1057</td>
<td><strong>1061</strong></td>
<td>1036</td>
<td>1030</td>
<td>1057</td>
<td><strong>1061</strong></td>
<td>1061</td>
<td>1061</td>
<td>1061</td>
<td>1061</td>
</tr>
<tr>
<td>p4.4.p</td>
<td>996</td>
<td>1056</td>
<td>1120</td>
<td>1120</td>
<td>1111</td>
<td>1120</td>
<td>1122</td>
<td><strong>1124</strong></td>
<td>1124</td>
<td>1124</td>
<td>1124</td>
<td>1124</td>
</tr>
<tr>
<td>p4.4.q</td>
<td>1084</td>
<td>1124</td>
<td>1157</td>
<td><strong>1161</strong></td>
<td>1145</td>
<td>1149</td>
<td>1160</td>
<td><strong>1161</strong></td>
<td>1161</td>
<td>1161</td>
<td>1157</td>
<td>1161</td>
</tr>
<tr>
<td>p4.4.r</td>
<td>1155</td>
<td>1165</td>
<td>1211</td>
<td>1207</td>
<td>1200</td>
<td>1193</td>
<td>1213</td>
<td><strong>1216</strong></td>
<td>1216</td>
<td>1216</td>
<td>1211</td>
<td>1216</td>
</tr>
<tr>
<td>p4.4.s</td>
<td>1230</td>
<td>1243</td>
<td>1256</td>
<td>1260</td>
<td>1249</td>
<td>1213</td>
<td>1250</td>
<td>1256</td>
<td><strong>1260</strong></td>
<td><strong>1260</strong></td>
<td><strong>1260</strong></td>
<td><strong>1260</strong></td>
</tr>
<tr>
<td>p4.4.t</td>
<td>1253</td>
<td>1255</td>
<td><strong>1285</strong></td>
<td><strong>1285</strong></td>
<td>1281</td>
<td>1281</td>
<td>1280</td>
<td><strong>1285</strong></td>
<td><strong>1285</strong></td>
<td><strong>1285</strong></td>
<td><strong>1285</strong></td>
<td><strong>1285</strong></td>
</tr>
</tbody>
</table>

| Best Gap | 4.36% | 1.99% | 0.48% | 0.06% | 0.30% | 1.46% | 0.11% | 0.06% | 0.01% | 0.05% | 0.09% |
| Avg Time | 796.70 | 105.30 | 22.50 | 457.90 | 32.00 | 36.70 | 367.40 | 81.00 | 298.57 | 301.54 | 76.72 |
References:

Contents

1 Multi-attribute Vehicle Routing Problems

2 Heuristics and Metaheuristics
   • A quick guided tour of CVRP metaheuristics
   • Successful strategies – Analysis

3 Unified Hybrid Genetic Search
   • General description
   • Tricks of the trade

4 Structural Problem Decompositions
   • Arc Routing Problems
   • Team-Orienteering Problems
   • CVRP – Sequence or Set optimization?

5 Conclusions and Perspectives
CVRP = Assignment + Sequencing

• Most state-of-the-art CVRP metaheuristics built on a combination of inter- and intra-route neighborhoods, usually simple variations of Swap, Relocate, Cross-Exchanges, 2-opt and 2-opt*

• These neighborhoods alone are generally sufficient to obtain TSP-optimal routes for classical benchmark instances (rarely contain over 20 customers per route)
⇒ Larger intra-route neighborhoods are not commonly used

• Does this mean that we should consider Sequencing optimization a “solved case” and focus on Assignment optimization in majority?
CVRP = Assignment + Sequencing

• Does this mean that we should consider Sequencing optimization a “solved case” and focus on Assignment optimization in majority?

• Inter-route moves often lead to TSP-suboptimal tours which are rejected due to their higher cost, but could be accepted if the tours were simultaneously optimized ⇒ (see, e.g., GENI – Gendreau et al. 1994).
CVRP = **Assignment** + **Sequencing**

- The two decision sets – **Assignments** and **Sequencing** – can be used to decompose the problem and define search space $S^A$
- **Assignments** decisions are decoded into complete solutions by a TSP solver:
Question 1: Is it practical and worthwhile to search in $S^A$ rather than $S$?

Question 2: If searching in $S^A$ requires too much effort, can we define an intermediate search space with some properties of $S^A$ but easier to explore?
Small example with 3 customers

- From $S$ to $S^A$: a much smaller search space

```
```

Solution quality
Small example with 3 customers

- From $\mathcal{S}$ to $\mathcal{S}^A$: a much smaller search space
Using Concorde to search in $S^A$

- The Concorde solver was used for TSP optimizations (Applegate et al., 2006)
- Experiments consider a single local search execution. Results consider 100 executions for each instance.
- The initial solution was produced by the savings algorithm of Clarke and Wright (1964).
Computational experiments

• Idea seems promising!
• Local search on the space of assignments ($S^A$) resulted in improved solutions

- However... runtime was prohibitive (even with efficient exploration strategies)
- We went from less than a second to over an hour
Computational experiments

- Idea seems promising!
- Local search on the space of assignments ($S^A$) resulted in improved solutions

- However... runtime was prohibitive (even with efficient exploration strategies)
- We went from less than a second to over an hour
An intermediate approach?

• In a recent conference presentation, Irnich (2013) proposed using the B&S neighborhood *in combination* with some classical CVRP moves.

• B&S Neighborhood (Balas and Simonetti, 2001)
  - Given a range parameter $k$ and an initial tour, the B&S algorithm finds, in $O(k^22^{k-2}n)$ operations, the vertex sequence with minimum cost such that no vertex is displaced by more than $k$ positions.

• $\Rightarrow$ $B^k$–optimal tour
  - A tour $\sigma$ is $B^k$–optimal if there exists no other permutation of its visits $\pi \circ \sigma$ with a shorter total distance such that $\pi(1) = 1$ and $\pi(i) \leq \pi(j)$ for all $i, j \in \{1, \ldots, n\}$ with $i + k \leq j$. 
An intermediate approach?

- In a recent conference presentation, Irnich (2013) proposed using the B&S neighborhood *in combination* with some classical CVRP moves.

- **B&S Neighborhood** (Balas and Simonetti, 2001)
  - Given a range parameter $k$ and an initial tour, the B&S algorithm finds, in $O(k^22^{k-2}n)$ operations, the vertex sequence with minimum cost such that no vertex is displaced by more than $k$ positions.

- $\Rightarrow$ **B$^k$-optimal tour**
  - A tour $\sigma$ is B$^k$-optimal if there exists no other permutation of its visits $\pi \circ \sigma$ with a shorter total distance such that $\pi(1) = 1$ and $\pi(i) \leq \pi(j)$ for all $i, j \in \{1, \ldots, n\}$ with $i + k \leq j$. 
An alternative search space

- We decided to investigate a systematic use of B&S in combination with *every* move of a LS.

- **Search Space** $S^B_k$:
  - Set of primitive solutions $Y$ is a subset of the complete solutions, those containing only $B^k$-optimal tours;
  - A nontrivial decoder $f$ is used, consisting of applying B&S multiple times to each route with a fixed $k$-range until tours are $B^k$-optimal;

**Properties:**
- From an initial solution containing $B^k$-optimal tours, a local search in the space $S^B_k$ explores only $B^k$-optimal tours.
- For a fixed range $k$, each move evaluation and subsequent solution decoding is done in polynomial time as a function of $n$ and the number of applications of B&S.
- The search space $S^B_k$ is such that $S^B_0 = S$ and $S^B_{n-1} = S^A$. 
• We decided to investigate a systematic use of B&S in combination with every move of a LS.

• **Search Space** $S^B_k$:
  
  ▶ Set of primitive solutions $Y$ is a subset of the complete solutions, those containing only $B^k$-optimal tours;
  
  ▶ A nontrivial decoder $f$ is used, consisting of applying B&S multiple times to each route with a fixed $k$-range until tours are $B^k$-optimal;

**Properties:**

▶ From an initial solution containing $B^k$-optimal tours, a local search in the space $S^B_k$ explores only $B^k$-optimal tours.

▶ For a fixed range $k$, each move evaluation and subsequent solution decoding is done in polynomial time as a function of $n$ and the number of applications of B&S.

▶ The search space $S^B_k$ is such that $S^B_0 = S$ and $S^B_{n-1} = S^A$. 

---

> Problem Metaheuristics UHGS Decompositions Conclusions References 135/168
An alternative search space

- We decided to investigate a systematic use of B&S in combination with every move of a LS.

- **Search Space $S^B_k$:**
  - Set of primitive solutions $Y$ is a subset of the complete solutions, those containing only $B^k$–optimal tours;
  - A nontrivial decoder $f$ is used, consisting of applying B&S multiple times to each route with a fixed $k$-range until tours are $B^k$–optimal;

**Properties:**
- From an initial solution containing $B^k$–optimal tours, a local search in the space $S^B_k$ explores only $B^k$–optimal tours.
- For a fixed range $k$, each move evaluation and subsequent solution decoding is done in polynomial time as a function of $n$ and the number of applications of B&S.
- The search space $S^B_k$ is such that $S^B_0 = S$ and $S^B_{n-1} = S^A$. 
An alternative search space

- We decided to investigate a systematic use of B&S in combination with every move of a LS.

**Search Space $S^B_k$:**

- Set of primitive solutions $Y$ is a subset of the complete solutions, those containing only $B^k$–optimal tours;
- A nontrivial decoder $f$ is used, consisting of applying B&S multiple times to each route with a fixed $k$-range until tours are $B^k$–optimal;

**Properties:**

- From an initial solution containing $B^k$–optimal tours, a local search in the space $S^B_k$ explores only $B^k$–optimal tours.
- For a fixed range $k$, each move evaluation and subsequent solution decoding is done in polynomial time as a function of $n$ and the number of applications of B&S.
- The search space $S^B_k$ is such that $S^B_0 = S$ and $S^B_{n-1} = S^A$. 
An alternative search space

• We decided to investigate a systematic use of B&S in combination with every move of a LS.

• Search Space $S^B_k$:
  ▶ Set of primitive solutions $Y$ is a subset of the complete solutions, those containing only $B^k$–optimal tours;
  ▶ A nontrivial decoder $f$ is used, consisting of applying B&S multiple times to each route with a fixed $k$-range until tours are $B^k$–optimal;

Properties:
  ▶ From an initial solution containing $B^k$–optimal tours, a local search in the space $S^B_k$ explores only $B^k$–optimal tours.
  ▶ For a fixed range $k$, each move evaluation and subsequent solution decoding is done in polynomial time as a function of $n$ and the number of applications of B&S.
  ▶ The search space $S^B_k$ is such that $S^B_0 = S$ and $S^B_{n-1} = S^A$. 

An alternative search space

• We decided to investigate a systematic use of B&S in combination with every move of a LS.

• **Search Space** $S^B_k$:
  - Set of primitive solutions $Y$ is a subset of the complete solutions, those containing only $B^k$–optimal tours;
  - A nontrivial decoder $f$ is used, consisting of applying B&S multiple times to each route with a fixed $k$-range until tours are $B^k$–optimal;

**Properties:**

- From an initial solution containing $B^k$–optimal tours, a local search in the space $S^B_k$ explores only $B^k$–optimal tours.
- For a fixed range $k$, each move evaluation and subsequent solution decoding is done in polynomial time as a function of $n$ and the number of applications of B&S.
- The search space $S^B_k$ is such that $S^B_0 = S$ and $S^B_{n-1} = S^A$. 
Small example with 3 customers

- Beginning from $S = S_0^B$, then $S_1^B$, and finally $S_A = S_2^B$:
Small example with 3 customers

- Beginning from $S = S^B_0$, then $S^B_1$, and finally $S^A = S^B_2$:
Small example with 3 customers

- Beginning from $\mathcal{S} = \mathcal{S}_0^B$, then $\mathcal{S}_1^B$, and finally $\mathcal{S}_A = \mathcal{S}_2^B$: 

```
<table>
<thead>
<tr>
<th>[2][1,3]</th>
<th>[2][3,1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[2,1,3]</td>
<td>[2,3,1]</td>
</tr>
<tr>
<td>[1,3,2]</td>
<td>[3,2,1]</td>
</tr>
<tr>
<td>[1][2][3]</td>
<td>[3][1,2]</td>
</tr>
<tr>
<td>[1][2,3]</td>
<td>[3][2,1]</td>
</tr>
<tr>
<td>[1][3,2]</td>
<td>[3][1,2]</td>
</tr>
</tbody>
</table>
```

solution quality
Efficient Local Search

- LS based on classical RELOCATE and SWAP, for single vertices or generalized to consecutive vertex pairs, along with 2-OPT and 2-OPT* moves.

- Speedup techniques to reduce the search effort: static neighborhood restrictions, dynamic move filters, concatenation techniques and memory structures.
As detailed in Vidal et al. (2013a), and in a similar way as Johnson and McGeoch (1997) and Toth and Vigo (2003): we restrict the search to the subset of moves that reconnect at least one vertex $i$ with a vertex $j$ belonging to the $\Gamma$ closest vertices of $i$.

⇒ Number of moves is $\mathcal{O}(\Gamma n)$
Dynamic move filters

- Decoding solutions (by applying B&S multiple times) is time consuming.
- It is thus important to reduce the number of decoded solutions, filtering infeasible and non-promising moves.

- Dynamic move filters:
  - Only solutions resulting from moves that increased the distance by a factor $1 + \phi$ or less are decoded:
    \[ z(\phi(x^t)) \leq (1 + \psi) \times z(x^t) \]
  - Parameter $\phi$ plays an important role defining the percentage of evaluated moves.
Dynamic move filters

• Decoding solutions (by applying B&S multiple times) is **time consuming**.

• It is thus important to **reduce** the number of decoded solutions, filtering **infeasible** and **non-promising** moves.

• **Dynamic move filters:**
  
  ▶ Only solutions resulting from moves that increased the distance by a factor $1 + \phi$ or less are decoded:

  $$z(\phi(x^t)) \leq (1 + \psi) \times z(x^t)$$

  ▶ Parameter $\phi$ plays an important role defining the percentage of evaluated moves.
Dynamic move filters

• Parameter $\phi$ is dynamically adjusted given a target range $[\xi^-, \xi^+]$ for the fraction of filtered moves.

• After each 1,000 move evaluations, the fraction $\xi$ of filtered moves is collected and $\phi$ is updated.

$$\psi = \begin{cases} 
\psi \times \alpha & \text{if } \xi \leq \xi^-, \\
\psi / \alpha & \text{if } \xi \geq \xi^+, \\
\psi & \text{otherwise.}
\end{cases}$$
Constant-time Evaluations

- Constant-time feasibility checks and computations of hash functions play an important role in the algorithm, exploiting the same concepts as Vidal et al. (2014):

\[
Q(\sigma^1 \oplus \sigma^2) = Q(\sigma^1) + Q(\sigma^2)
\]
\[
C(\sigma^1 \oplus \sigma^2) = C(\sigma^1) + d_{\sigma^1(|\sigma^1|,\sigma^2(1)} + C(\sigma^2)
\]
\[
H^p(\sigma^1 \oplus \sigma^2) = H^p(\sigma^1) + \rho^{\sigma^1} \times H^p(\sigma^2)
\]
\[
H^s(\sigma^1 \oplus \sigma^2) = H^s(\sigma^1) + H^s(\sigma^2).
\]
Moreover, can we transform the search space $S$ or $S^B_k$ so that it converges towards $S^A$ as the optimization is run?

▶ Definitely, using long-term memories to implement a **tunneling** strategy!
Tunneling strategy

Input route $\sigma$

Has $\sigma$ already been decoded in the past search

- YES: Return the best known route $\pi$ for this visit set
- NO: Apply B&S iteratively on $\sigma$ to generate a $B^k$-optimal route $\pi'$

If $\pi'$ is better than the best known route for this visit set, update $\pi$

“Decoded” route $\pi$
Small example with 3 customers

- Tunneling effect on search space $S_1^B$: in this example, search space converges to $S^A$ when solution $[3,1,2]$ is discovered.
Small example with 3 customers

- Tunneling effect on search space $S_1^B$: in this example, search space converges to $S^A$ when solution [3,1,2] is discovered.
Algorithm 1: Efficient local search in the space $S^B_k$
Input: An initial complete solution $x^0$, an evaluation threshold $\psi$ and a granularity threshold $\Gamma$

1. $t \leftarrow 0$
2. repeat
   // Enumerating $O(\Gamma n)$ moves - candidate lists based on vertex proximity
   for each move $\phi(x^t) \in \mathcal{N}(x^t)$ involving a vertex pair $(i, j)$, $j \in \Gamma(i)$
   3. The move $\phi$ modifies up to two routes of $x^t$. Let $z_{\text{before}}$ be the sum of the costs of these two routes, and let $(\sigma^1_1, \ldots, \sigma^1_{b_1})$ and $(\sigma^2_1, \ldots, \sigma^2_{b_2})$ be the new routes in $\phi(x^t)$.
   // First, filter infeasible moves with respect to capacity constraints in $O(1)$:
   if $Q(\sigma^1_1 \oplus \cdots \oplus \sigma^1_{b_1}) > Q$ or $Q(\sigma^2_1 \oplus \cdots \oplus \sigma^2_{b_2}) > Q$ then
      continue.
   // Second, consider the cost of the classical CVRP move to filter non-promising solutions in $O(1)$:
   if $z(x^t) + C(\sigma^1_1 \oplus \cdots \oplus \sigma^1_{b_1}) + C(\sigma^2_1 \oplus \cdots \oplus \sigma^2_{b_2}) - z_{\text{before}} > (1 + \psi) \times z(x^t)$ then
      continue.
   // Third, decode the routes $\sigma^1$ and $\sigma^2$ to evaluate the move $\phi$ in $S^B_k$:
   $z_{\text{move}} \leftarrow 0$
   for each route $\sigma^i$ with $i \in \{1, 2\}$
      // Compute hash key in $O(1)$ and check memory in $O(1)$:
      $(\bar{\sigma}^i, \bar{z}_i) \leftarrow \text{LOOKUP}(H(\sigma^i_1 \oplus \cdots \oplus \sigma^i_{b_1}))$
      if $(\bar{\sigma}^i, \bar{z}_i) = \text{NOT FOUND}$ then
         $(\bar{\sigma}^i, \bar{z}_i) \leftarrow \text{BALAS-SIMONETTI}(\sigma^i_1 \oplus \cdots \oplus \sigma^i_{b_1})$
         STORE($(\bar{\sigma}^i, \bar{z}_i), H(\sigma^i_1 \oplus \cdots \oplus \sigma^i_{b_1}))$
      $z_{\text{move}} \leftarrow z_{\text{move}} + \bar{z}_i$
      // Filter non-improving moves:
      if $z_{\text{move}} \geq z_{\text{before}}$ then
         continue.
      // At this stage, apply $\phi$ since it is an improving move in $S^B_k$:
      Set $x^{t+1} = \phi(x)$ ; $t = t + 1$
      Replace the routes $(\sigma^1, \sigma^2)$ by $(\bar{\sigma}^1, \bar{\sigma}^2)$ in $x^{t+1}$
   until $x^t$ is a local minimum
3. return $x^t$
Computational experiments

- Experiments with simple local search – on all instances
Computational experiments

- Experiments with simple local search – instances with route cardinality in range [3.0, 4.55]:

![Box plots showing gap and CPU time](image-url)
Computational experiments

- Experiments with simple local search – instances with route cardinality in range $[16.47, 24.43]$: 

![Graphs showing gap and CPU time for different instances and decompositions.](image-url)
Computational experiments

- Results with different target target intervals $[\xi^-, \xi^+]$ (desired quantity of filtered moves):
Computational experiments

- Experiments with UHGS-BS and different values for parameter $k$:

![Box plots showing gap and CPU time for different parameter values.](image)

- Wilcoxon tests show that statistically significant differences exist between the results of $S$, $S_1^B$ and $S_2^B$ (p-values < 0.05)
Computational experiments

- Experiments with UHGS-BS and different values for parameter $k$:

- Wilcoxon tests show that statistically significant differences exist between the results of $S$, $S_1^B$ and $S_2^B$ (p-values $< 0.05$)
<table>
<thead>
<tr>
<th>#</th>
<th>Instance</th>
<th>ILS</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Time</td>
<td>Average</td>
<td>Best</td>
<td>Time</td>
<td>Average</td>
<td>Best</td>
<td>Time</td>
<td>Average</td>
<td>Best</td>
<td>Time</td>
<td>Average</td>
<td>Best</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>X-n101-k25</td>
<td>0.1</td>
<td>27591</td>
<td>27591</td>
<td>1.4</td>
<td>27591</td>
<td>27591</td>
<td>2.4</td>
<td>27591</td>
<td>27591</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>X-n106-k14</td>
<td>2.0</td>
<td>26375.9</td>
<td>26362</td>
<td>4.0</td>
<td>26381.8</td>
<td>26378</td>
<td>17.6</td>
<td>26374.3</td>
<td>26362</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>X-n110-k13</td>
<td>0.2</td>
<td>14971</td>
<td>14971</td>
<td>1.6</td>
<td>14971</td>
<td>14971</td>
<td>3.9</td>
<td>14971</td>
<td>14971</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>X-n115-k10</td>
<td>0.2</td>
<td>12747</td>
<td>12747</td>
<td>1.8</td>
<td>12747</td>
<td>12747</td>
<td>6.4</td>
<td>12747</td>
<td>12747</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>X-n120-k6</td>
<td>1.7</td>
<td>13337</td>
<td>13332</td>
<td>2.3</td>
<td>13332</td>
<td>13332</td>
<td>38.3</td>
<td>13332</td>
<td>13332</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>X-n125-k30</td>
<td>1.4</td>
<td>55673.8</td>
<td>55539</td>
<td>2.7</td>
<td>55542.1</td>
<td>55539</td>
<td>6.1</td>
<td>55540</td>
<td>55539</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>X-n129-k18</td>
<td>1.9</td>
<td>28998</td>
<td>28948</td>
<td>2.7</td>
<td>28948.5</td>
<td>28940</td>
<td>8.4</td>
<td>28940</td>
<td>28940</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>X-n134-k13</td>
<td>2.1</td>
<td>10947</td>
<td>10916</td>
<td>3.3</td>
<td>10934.9</td>
<td>10916</td>
<td>20.2</td>
<td>10916</td>
<td>10916</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>X-n139-k10</td>
<td>1.6</td>
<td>13603</td>
<td>13590</td>
<td>2.3</td>
<td>13590</td>
<td>13590</td>
<td>8.9</td>
<td>13590</td>
<td>13590</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>X-n143-k7</td>
<td>1.6</td>
<td>15745.2</td>
<td>15726</td>
<td>3.1</td>
<td>15700.2</td>
<td>15700</td>
<td>33.1</td>
<td>15700</td>
<td>15700</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>X-n148-k46</td>
<td>0.8</td>
<td>43452.1</td>
<td>43448</td>
<td>3.2</td>
<td>43448</td>
<td>43448</td>
<td>5.1</td>
<td>43448</td>
<td>43448</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>X-n153-k22</td>
<td>0.5</td>
<td>21400</td>
<td>21340</td>
<td>5.5</td>
<td>21226.3</td>
<td>21220</td>
<td>15.2</td>
<td>21225.6</td>
<td>21225</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>X-n157-k13</td>
<td>0.8</td>
<td>16876</td>
<td>16876</td>
<td>3.2</td>
<td>16876</td>
<td>16876</td>
<td>28.4</td>
<td>16876</td>
<td>16876</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>X-n162-k11</td>
<td>0.5</td>
<td>14160.1</td>
<td>14138</td>
<td>3.3</td>
<td>14141.3</td>
<td>14138</td>
<td>12.2</td>
<td>14138</td>
<td>14138</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>X-n167-k10</td>
<td>0.9</td>
<td>20608.7</td>
<td>20562</td>
<td>3.7</td>
<td>20563.2</td>
<td>20557</td>
<td>36.1</td>
<td>20557</td>
<td>20557</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>X-n172-k51</td>
<td>0.6</td>
<td>45616.1</td>
<td>45607</td>
<td>3.8</td>
<td>45607</td>
<td>45607</td>
<td>5.7</td>
<td>45607</td>
<td>45607</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>X-n176-k26</td>
<td>1.1</td>
<td>48249.8</td>
<td>48140</td>
<td>7.6</td>
<td>47957.2</td>
<td>47812</td>
<td>16.2</td>
<td>47830.7</td>
<td>47812</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>X-n181-k23</td>
<td>1.6</td>
<td>25571.5</td>
<td>25569</td>
<td>6.3</td>
<td>25591.1</td>
<td>25569</td>
<td>13.9</td>
<td>25569.4</td>
<td>25569</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>X-n186-k15</td>
<td>1.7</td>
<td>24186.0</td>
<td>24145</td>
<td>5.9</td>
<td>24147.2</td>
<td>24145</td>
<td>20.2</td>
<td>24145.0</td>
<td>24145</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>X-n190-k8</td>
<td>2.1</td>
<td>17143.1</td>
<td>17085</td>
<td>12.1</td>
<td>16987.9</td>
<td>16980</td>
<td>161.9</td>
<td>16985.3</td>
<td>16980</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>X-n195-k51</td>
<td>0.9</td>
<td>44234.3</td>
<td>44225</td>
<td>6.1</td>
<td>44244.1</td>
<td>44225</td>
<td>9.3</td>
<td>44283.8</td>
<td>44225</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>X-n200-k36</td>
<td>7.5</td>
<td>58697.2</td>
<td>58626</td>
<td>8.0</td>
<td>58626.4</td>
<td>58578</td>
<td>12.0</td>
<td>58615.1</td>
<td>58578</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>X-n204-k19</td>
<td>1.1</td>
<td>19625.2</td>
<td>19570</td>
<td>5.4</td>
<td>19571.5</td>
<td>19565</td>
<td>15.7</td>
<td>19567.0</td>
<td>19565</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>X-n209-k16</td>
<td>3.8</td>
<td>30765.4</td>
<td>30667</td>
<td>8.6</td>
<td>30680.4</td>
<td>30656</td>
<td>35.7</td>
<td>30671.3</td>
<td>30656</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>X-n214-k11</td>
<td>2.3</td>
<td>11126.9</td>
<td>10985</td>
<td>10.2</td>
<td>10877.4</td>
<td>10856</td>
<td>52.3</td>
<td>10872.1</td>
<td>10856</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>X-n219-k7</td>
<td>0.9</td>
<td>117595.0</td>
<td>117595</td>
<td>7.7</td>
<td>117604.9</td>
<td>117595</td>
<td>18.2</td>
<td>117600.5</td>
<td>117595</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>X-n223-k34</td>
<td>8.5</td>
<td>40533.5</td>
<td>40471</td>
<td>8.3</td>
<td>40499.0</td>
<td>40437</td>
<td>18.6</td>
<td>40478.4</td>
<td>40437</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>X-n228-k23</td>
<td>2.4</td>
<td>25795.8</td>
<td>25743</td>
<td>9.8</td>
<td>25779.3</td>
<td>25742</td>
<td>29.0</td>
<td>25768.0</td>
<td>25743</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>X-n233-k16</td>
<td>3.0</td>
<td>19336.7</td>
<td>19266</td>
<td>6.8</td>
<td>19288.4</td>
<td>19230</td>
<td>34.0</td>
<td>19276.5</td>
<td>19230</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 2: Results for the instances with up to 331 customers from Uchoa et al. (2017)

<table>
<thead>
<tr>
<th>Instance</th>
<th>ILS</th>
<th>UHGS</th>
<th>UHGS-BS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Average</td>
<td>Best</td>
<td>Time</td>
</tr>
<tr>
<td>30 X-n237-k14</td>
<td>3.5</td>
<td>27078.8</td>
<td><strong>27042</strong></td>
</tr>
<tr>
<td>31 X-n242-k48</td>
<td>17.8</td>
<td><strong>82874.2</strong></td>
<td>82774</td>
</tr>
<tr>
<td>32 X-n247-k47</td>
<td>2.1</td>
<td>37507.2</td>
<td>37289</td>
</tr>
<tr>
<td>33 X-n251-k28</td>
<td>10.8</td>
<td>38840.0</td>
<td>38727</td>
</tr>
<tr>
<td>34 X-n256-k16</td>
<td>2.0</td>
<td>18883.9</td>
<td>18880</td>
</tr>
<tr>
<td>35 X-n261-k13</td>
<td>6.7</td>
<td>26869.0</td>
<td>26706</td>
</tr>
<tr>
<td>36 X-n266-k58</td>
<td>10.0</td>
<td><strong>75563.3</strong></td>
<td><strong>75478</strong></td>
</tr>
<tr>
<td>37 X-n270-k35</td>
<td>9.1</td>
<td>35363.4</td>
<td>35324</td>
</tr>
<tr>
<td>38 X-n275-k28</td>
<td>3.6</td>
<td>21256.0</td>
<td><strong>21245</strong></td>
</tr>
<tr>
<td>39 X-n280-k17</td>
<td>9.6</td>
<td>33769.4</td>
<td>33624</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance</th>
<th>ILS</th>
<th>UHGS</th>
<th>UHGS-BS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Average</td>
<td>Best</td>
<td>Time</td>
</tr>
<tr>
<td>40 X-n284-k15</td>
<td>8.6</td>
<td>20448.5</td>
<td>20295</td>
</tr>
<tr>
<td>41 X-n289-k60</td>
<td>16.1</td>
<td>95450.6</td>
<td>95315</td>
</tr>
<tr>
<td>42 X-n294-k50</td>
<td>12.4</td>
<td><strong>47254.7</strong></td>
<td>47190</td>
</tr>
<tr>
<td>43 X-n298-k31</td>
<td>6.9</td>
<td>34356.0</td>
<td>34239</td>
</tr>
<tr>
<td>44 X-n303-k21</td>
<td>14.2</td>
<td>21895.8</td>
<td>21812</td>
</tr>
<tr>
<td>45 X-n308-k13</td>
<td>9.5</td>
<td>26101.1</td>
<td>25901</td>
</tr>
<tr>
<td>46 X-n313-k71</td>
<td>17.5</td>
<td>94297.3</td>
<td>94192</td>
</tr>
<tr>
<td>47 X-n317-k53</td>
<td>8.6</td>
<td><strong>78356.0</strong></td>
<td><strong>78355</strong></td>
</tr>
<tr>
<td>48 X-n322-k28</td>
<td>14.7</td>
<td>29991.3</td>
<td>29877</td>
</tr>
<tr>
<td>49 X-n327-k20</td>
<td>19.1</td>
<td>27812.4</td>
<td>27599</td>
</tr>
<tr>
<td>50 X-n331-k15</td>
<td>15.7</td>
<td>31235.5</td>
<td>31105</td>
</tr>
</tbody>
</table>

Average gap: 0.37% 0.13% 0.14% 0.02% 0.10% 0.00%
## Table 3: Results for the instances with more than 331 customers from Uchoa et al. (2017)

<table>
<thead>
<tr>
<th># Instance</th>
<th>ILS</th>
<th>UHGS</th>
<th>UHGS-BS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Average</td>
<td>Best</td>
</tr>
<tr>
<td>51 X-n336-k84</td>
<td>21.4</td>
<td>139461.0</td>
<td><strong>139197</strong></td>
</tr>
<tr>
<td>52 X-n344-k43</td>
<td>22.6</td>
<td>42284.0</td>
<td>42146</td>
</tr>
<tr>
<td>53 X-n351-k40</td>
<td>25.2</td>
<td>26150.3</td>
<td>26021</td>
</tr>
<tr>
<td>54 X-n359-k29</td>
<td>48.9</td>
<td>52076.5</td>
<td>51706</td>
</tr>
<tr>
<td>55 X-n367-k17</td>
<td>13.1</td>
<td>23003.2</td>
<td>22902</td>
</tr>
<tr>
<td>56 X-n376-k94</td>
<td>7.1</td>
<td><strong>147713.0</strong></td>
<td><strong>147713</strong></td>
</tr>
<tr>
<td>57 X-n384-k52</td>
<td>34.5</td>
<td>66372.5</td>
<td>66116</td>
</tr>
<tr>
<td>58 X-n393-k38</td>
<td>20.8</td>
<td>38457.4</td>
<td>38298</td>
</tr>
<tr>
<td>59 X-n401-k29</td>
<td>60.4</td>
<td>66715.1</td>
<td>66453</td>
</tr>
</tbody>
</table>

### Average gap:
- ILS: 0.74%
- UHGS: 0.42%
- UHGS-BS: 0.30%
- Average gap: 0.06%
## Final results (large instances)

<table>
<thead>
<tr>
<th># Instance</th>
<th>ILS</th>
<th>UHGS</th>
<th>UHGS-BS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Average</td>
<td>Best</td>
</tr>
<tr>
<td>80 X-n641-k35</td>
<td>140.4</td>
<td>64606.1</td>
<td>64462</td>
</tr>
<tr>
<td>81 X-n655-k131</td>
<td>47.2</td>
<td><strong>106782.0</strong></td>
<td><strong>106780</strong></td>
</tr>
<tr>
<td>82 X-n670-k126</td>
<td>61.2</td>
<td>147676.0</td>
<td>147045</td>
</tr>
<tr>
<td>83 X-n685-k75</td>
<td>73.9</td>
<td>68988.2</td>
<td>68646</td>
</tr>
<tr>
<td>84 X-n701-k44</td>
<td>210.1</td>
<td>83042.2</td>
<td>82888</td>
</tr>
<tr>
<td>85 X-n716-k35</td>
<td>225.8</td>
<td>44171.6</td>
<td>44021</td>
</tr>
<tr>
<td>86 X-n733-k159</td>
<td>111.6</td>
<td>137045.0</td>
<td>136832</td>
</tr>
<tr>
<td>87 X-n749-k98</td>
<td>127.2</td>
<td>78275.9</td>
<td>77952</td>
</tr>
<tr>
<td>88 X-n766-k71</td>
<td>242.1</td>
<td>115738.0</td>
<td>115443</td>
</tr>
<tr>
<td>89 X-n783-k48</td>
<td>235.5</td>
<td>73722.9</td>
<td>73447</td>
</tr>
<tr>
<td>90 X-n801-k40</td>
<td>432.6</td>
<td>74005.7</td>
<td>73830</td>
</tr>
<tr>
<td>91 X-n819-k171</td>
<td>148.9</td>
<td>159425.0</td>
<td>159164</td>
</tr>
<tr>
<td>92 X-n837-k142</td>
<td>173.2</td>
<td>195027.0</td>
<td>194804</td>
</tr>
<tr>
<td>93 X-n856-k95</td>
<td>153.7</td>
<td>89277.6</td>
<td>89060</td>
</tr>
<tr>
<td>94 X-n876-k59</td>
<td>409.3</td>
<td>100417.0</td>
<td>100177</td>
</tr>
<tr>
<td>95 X-n895-k37</td>
<td>410.2</td>
<td>54958.5</td>
<td>54713</td>
</tr>
<tr>
<td>96 X-n916-k207</td>
<td>226.1</td>
<td>330948.0</td>
<td>330639</td>
</tr>
<tr>
<td>97 X-n936-k151</td>
<td>202.5</td>
<td>134530.0</td>
<td>133592</td>
</tr>
<tr>
<td>98 X-n957-k87</td>
<td>311.2</td>
<td>85936.6</td>
<td>85697</td>
</tr>
<tr>
<td>99 X-n979-k58</td>
<td>687.2</td>
<td>120253.0</td>
<td>119994</td>
</tr>
<tr>
<td>100 X-n1001-k43</td>
<td>792.8</td>
<td>73985.4</td>
<td>73776</td>
</tr>
</tbody>
</table>

Average gap: 0.74% 0.42% 0.30% 0.06% 0.24% 0.03%
Contents

1 Multi-attribute Vehicle Routing Problems

2 Heuristics and Metaheuristics
   • A quick guided tour of CVRP metaheuristics
   • Successful strategies – Analysis

3 Unified Hybrid Genetic Search
   • General description
   • Tricks of the trade

4 Structural Problem Decompositions
   • Arc Routing Problems
   • Team-Orienteering Problems
   • CVRP – Sequence or Set optimization?

5 Conclusions and Perspectives
Conclusions and Perspectives

- Unified methods for vehicle routing problems, no need to reinvent the wheel for each new variant. **Generality does not necessarily impede efficiency for a large class of problems.**

- **Understanding the structure of the problems** is critical for the design of efficient methods

- Structural problem decompositions allow to relegate difficult decision classes (e.g., customer selection, edge orientations etc...) inside (modular) route-evaluation operators

- Efficient move evaluation strategies (e.g., pre-processing and dynamic programming) can lead to **considerable speedups.**

- Structural problem decompositions can be used to explore exponential-sized neighborhoods
• Perspectives: keep on focusing **problem structure**, **computational complexity** and **neighborhood search**. **Major breakthroughs are still possible around those research lines.**

• Following the recent advances of Arnold and Sörensen (2018) and Christiaens and Vanden Berghe (2018), design advanced inter-route moves which efficiently optimize the assignment decisions.

• Exploit pattern mining, machine learning and guidance to a larger extent...

• ...and many other promising perspectives
Thank you

THANK YOU FOR YOUR ATTENTION!

Articles, instances, detailed results and slides available at:
http://w1.cirrelt.ca/~vidalt/

Source code (GPL v3.0) available at:
https://github.com/vidalthi/HGS-CARP – Node, edge, and arc routing
https://github.com/vidalthi/HGSADC – Simple CVRP version (soon online)


For further reading III


For further reading V


For further reading VI


For further reading VII


For further reading X


