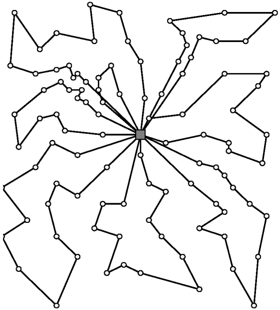
Heuristics for vehicle routing problems: Current challenges and future prospects

Thibaut Vidal

Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro



SOICT'17

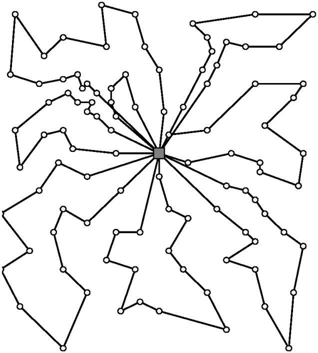
Nha Trang, December 8th, 2017

Contents

Contents

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.

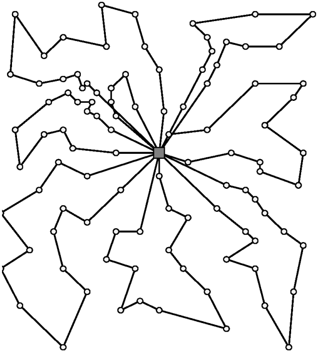


- NP-Hard problem
- recent breakthrough in exact methods enable to solve problems of moderate size with up to 300-400 customers (Uchoa et al., 2013).
- A Scopus search “Vehicle Routing” for 2007-2011 returns 1258 publications, including 566 journal papers.
- Massive research on heuristics

Multi-attribute vehicle routing problems (MAVRPs)

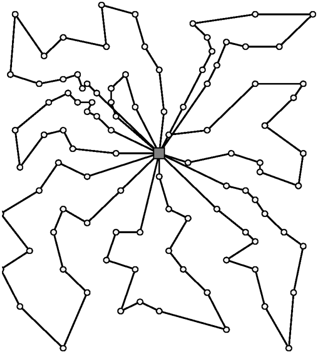
- Capacitated vehicle routing problems (VRP)
  - **Combinatorial explosion:** For a problem with **n=100 customers** and **a single vehicle**, the number of feasible solutions is:

    n! = 93326215443944152681699
    23885626670049071596826438 16
    21468592963895217599993229 91
    56089414639761565182862536 97
    92082722375825118521091686 40
    000000000000000000000000 $\approx 10^{158}$



Multi-attribute vehicle routing problems (MAVRPs)

- Even with a grid of computers which:
  - Contains as many CPUs as the estimated number of atoms in the universe : $n_{\mathrm{CPU}} = 10^{80}$
  - Does one operation per Planck time: $t_P = 5.39 \times 10^{-44}$ seconds



- We would need $T = 10^{158} \times 5.39 \times 10^{-44}/1080 = 5.39 \times 10^{34}$ seconds to enumerate all solutions.
- Compare this to the estimated age of Universe : 4.33×10^{17} seconds

Multi-attribute vehicle routing problems (MAVRPs)

- **Vehicle routing "attributes":** Supplementary decisions, constraints and objectives which complement the classic VRP formulation.
  - modeling the specificities of application cases, customer requirements, network and vehicle specificities, operators abilities...
  - e.g., service time windows, multiple periods of planning, multiple depots and facilities, heterogeneous fleet, 2D-3D loading, time-dependent travel times...

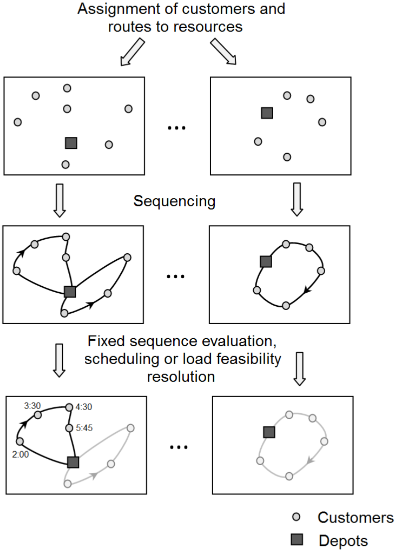
Multi-attribute vehicle routing problems (MAVRPs)

- **Vehicle routing "attributes":** Supplementary decisions, constraints and objectives which complement the classic VRP formulation.
  - modeling the specificities of application cases, customer requirements, network and vehicle specificities, operators abilities...
  - e.g., service time windows, multiple periods of planning, multiple depots and facilities, heterogeneous fleet, 2D-3D loading, time-dependent travel times...

- **Multi-Attribute Vehicle Routing Problems (MAVRP)**
  - Challenges: **VARIETY** of attributes
  - Challenges: **COMBINATION** of attributes
  - Plethora of attribute-specific methods in the literature, but highly problem specific
  - More **unified methods**, which can be extended to new problems without significant development, are necessary to answer the industrial needs in a timely manner.

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)

- More than 200 *problem attributes* have been proposed to this date
- These attributes often need to be combined together, leading to possibly 2^{200} problems... 2^{200} different methods, and 2^{200} papers ?!
- **Double combinational explosion**: from the variety of problems, from the number of solutions.

⇒ A need for more flexible and general purpose solvers

⇒ 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.

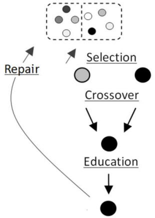
Contents

Unified Hybrid Genetic Search

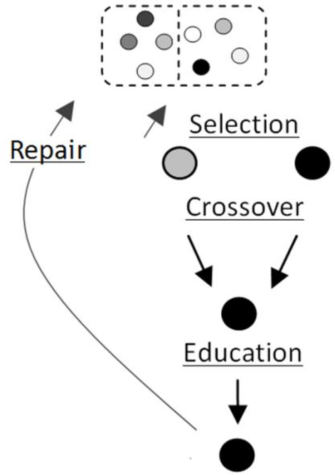
- **To solve these problems in a unified manner**, we proposed the **Unified Hybrid genetic Search** (UHGS) (Vidal et al., 2014)

- 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**

General HGA Methodology : Evolving a population of solutions with genetic operators, selection, crossover and mutation. The latter is replaced by a local search procedure.

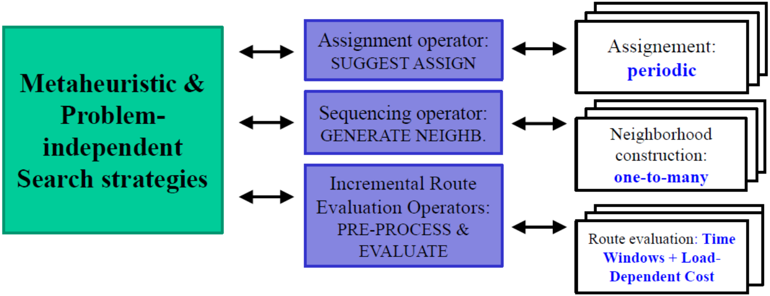


- **To solve these problems in a unified manner**, we proposed the **Unified Hybrid genetic Search** (UHGS) (Vidal et al., 2014)

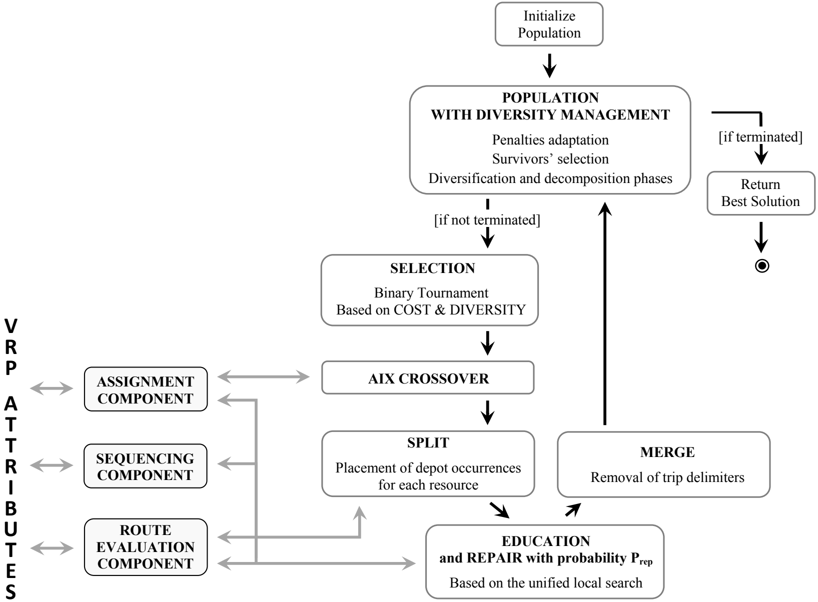


Unified Hybrid Genetic Search

- **To solve these problems in a unified manner**, we proposed the **Unified Hybrid genetic Search** (UHGS) (Vidal et al., 2014)
  - Relying on assignment, sequencing & route evaluation (RE) operators to do attribute-dependent tasks. Implemented in a generic way
  - Attribute-dependent modules are selected and combined by the method, relatively to the problem structure, to implement the assignment, sequencing and route evaluations.



Unified Hybrid Genetic Search



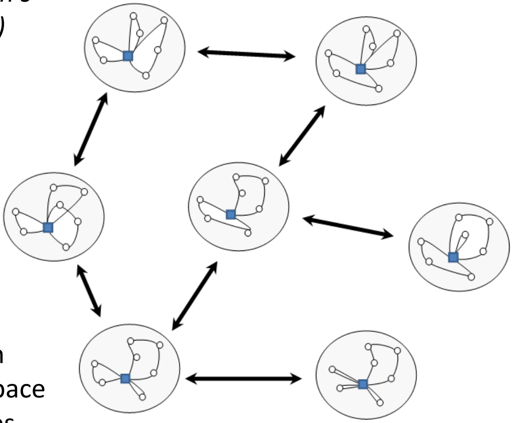
Unified Hybrid Genetic Search

- **Local-improvement procedures** :

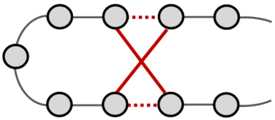
  - From an *incumbent solution s* define a *neighborhood N(s)* of solutions obtained by applying some changes

  - The set of solutions, linked by neighborhood relationships = *search space.*

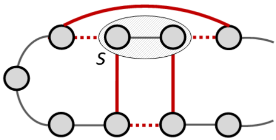
  - LS-improvement method progress from one solution to another in this search space as long at the cost improves.



- For optimizing a single route (TSP tour);
  - in the terminology of Lin (1965), *λ-opt* neighborhood = subset of moves obtained by deleting and reinserting λ 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.



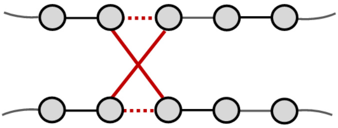
2-opt



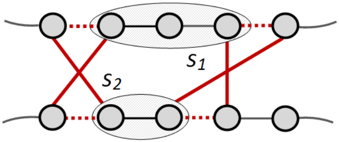
Or-exchange

Unified Hybrid Genetic Search

- 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)



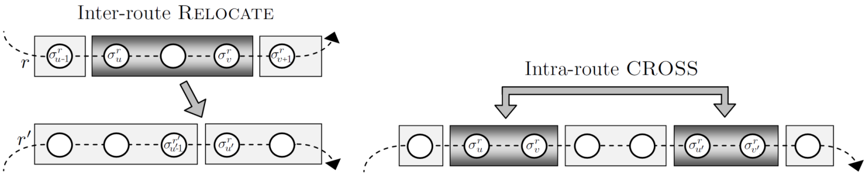
2-opt*



CROSS

Unified Hybrid Genetic Search

- One important structural property of local search, which considerably helps to progress towards **unified** and **efficient** metaheuristics:
  - Any local-search move involving a bounded number of node relocations or arc exchanges can be assimilated to a concatenation of a bounded number of sub-sequences from the incumbent solution



  - Data preprocessing: compute auxiliary data on subsequences to speed up the search
  - Can be computed by induction on the concatenation operator ($\oplus$).

- Example 1) Distance and capacity constraints

Auxiliary data structures:

Partial loads $L(\sigma)$ and distance $D(\sigma)$

Initialization

For a sequence σ_0 with a single visit v_i , $L(\sigma_0) = q_i$ and $D(\sigma_0) = 0$

Induction Step:

$Q(\sigma_1 \oplus \sigma_2) = Q(\sigma_1) + Q(\sigma_2)$

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

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 σ_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 and route duration constraints

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:

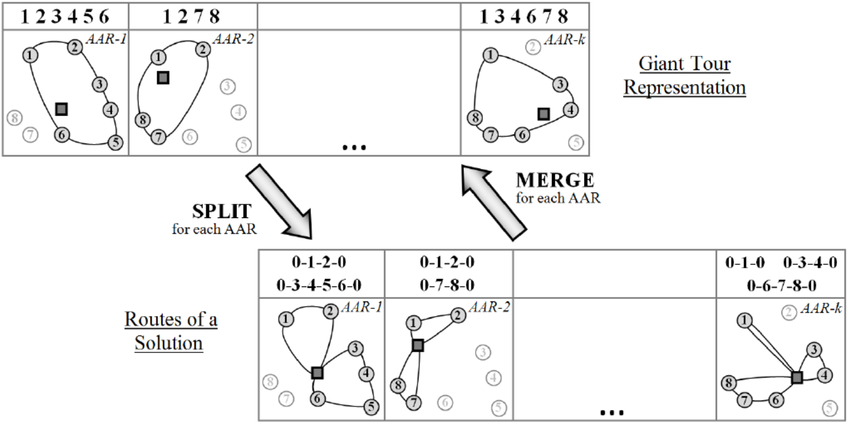
For a sequence σ_0 with a single visit v_i , $T(\sigma_0) = s_i$, $E(\sigma_0) = e_i + s_i$, $L(\sigma_0) = l_i$ and $F(\sigma_0) = true$.

Induction Step:

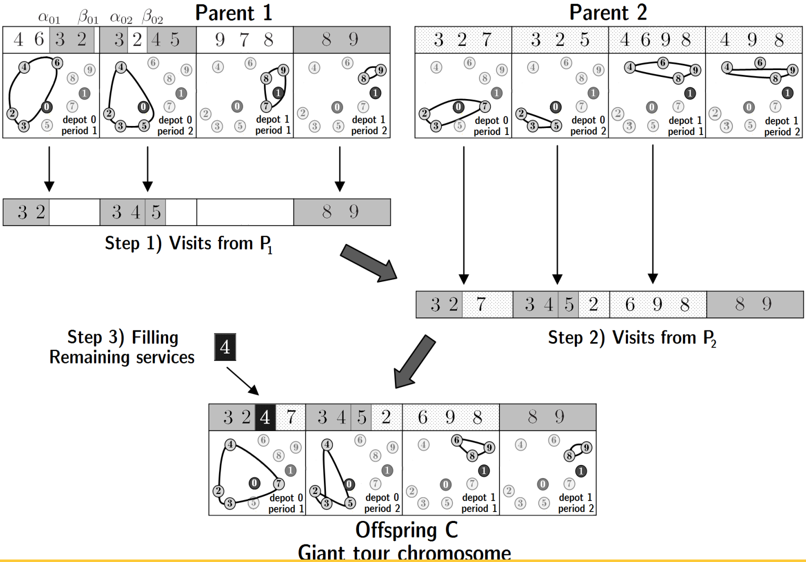
$$
\begin{aligned}
T(\sigma_1 \oplus \sigma_2) &= T(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} + T(\sigma_2) \\
E(\sigma_1 \oplus \sigma_2) &= \max\{E(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} + T(\sigma_2), E(\sigma_2)\} \\
L(\sigma_1 \oplus \sigma_2) &= \min\{L(\sigma_1), L(\sigma_2) - d_{\sigma_1(|\sigma_1|)\sigma_2(1)} - T(\sigma_1)\} \\
F(\sigma_1 \oplus \sigma_2) &\equiv F(\sigma_1) \ \wedge \ F(\sigma_2) \ \wedge \ (E(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} \leq L(\sigma_2))
\end{aligned}
$$

Unified Hybrid Genetic Search

- Unified solution representation and **Split**:



- Unified crossover operator:

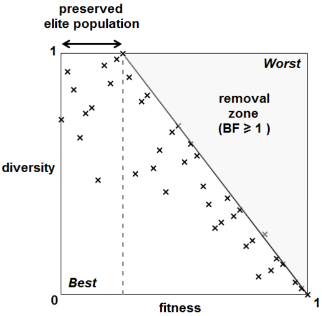


Unified Hybrid Genetic Search

- **Biased Fitness** is a tradeoff between ranks in terms of **solution penalized cost** *cost(I)*, and **contribution to the diversity** *dc(I)*, measured as a distance to others individuals in the population.

$$BF(I) = fit(I) + (1 - \frac{nbElit}{nbIndiv - 1}) \times dc(I)$$

- Used during selection of the parents
  - Balancing strength with innovation during reproduction, and thus favoring exploration of the search space.
- and during selection of the survivors:
  - Removing the individual *I* with worst *BF(I)* also guarantees some elitism in terms of solution value.



Unified Hybrid Genetic Search

- **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

Variant	Bench.	n	Obj.	State-of-the-art methods				
				Author	Avg.%	Best%	T(min)	CPU
CVRP	CMT79	[50,199]	C	GG11:	—	+0.03%	8×2.38	8×Xe 2.3G
				MB07:	+0.03%	—	2.80	P-IV 2.8G
				UHGS*:	**+0.02%**	**+0.00%**	11.90	Opt 2.4G
CVRP	GWKC98	[200,483]	C	GG11:	—	+0.29%	8×5	8×Xe 2.3G
				NB09:	+0.27%	+0.16%	21.51	Opt 2.4G
				UHGS*:	**+0.15%**	**+0.02%**	71.41	Opt 2.4G
VRPB	GJ89	[25,200]	C	ZK12:	+0.38%	+0.00%	1.09	T5500 1.67G
				GA09:	+0.09%	+0.00%	1.13	Xe 2.4G
				UHGS:	**+0.01%**	**+0.00%**	0.99	Opt 2.4G
CCVRP	CMT79	[50,199]	C	NPW10:	+0.74%	+0.28%	5.20	Core2 2G
				RL12:	+0.37%	+0.07%	2.69	Core2 2G
				UHGS:	**+0.01%**	**-0.01%**	1.42	Opt 2.2G
CCVRP	GWKC98	[200,483]	C	NPW10:	+2.03%	+1.38%	94.13	Core2 2G
				RL12:	+0.34%	+0.07%	21.11	Core2 2G
				UHGS:	**-0.14%**	**-0.23%**	17.16	Opt 2.2G
VRPSDP	SN99	[50,199]	C	SDBOF10:	+0.16%	+0.00%	256×0.37	256×Xe 2.67G
				ZTK10:	—	+0.11%	—	T5500 1.66G
				UHGS:	**+0.01%**	**+0.00%**	2.79	Opt 2.4G
VRPSDP	MG06	[100,400]	C	SDBOF10:	+0.30%	+0.17%	256×3.11	256×Xe 2.67G
				UHGS:	+0.20%	+0.07%	12.00	Opt 2.4G
				S12 :	**+0.08%**	**+0.00%**	7.23	I7 2.93G

Unified Hybrid Genetic Search

Variant	Bench.	n	Obj.	State-of-the-art methods				
				Author	Avg.%	Best%	T(min)	CPU
VFMP-F	G84	[20,100]	C	ISW09:	—	+0.07%	8.34	P-M 1.7G
				SPUO12:	+0.12%	+0.01%	0.15	I7 2.93G
				UHGS:	**+0.04%**	**+0.01%**	1.13	Opt 2.4G
VFMP-V	G84	[20,100]	C	ISW09:	—	+0.02%	8.85	P-M 1.7G
				SPUO12:	+0.17%	+0.00%	0.06	I7 2.93G
				UHGS:	**+0.03%**	**+0.00%**	0.85	Opt 2.4G
VFMP-FV	G84	[20,100]	C	P09:	—	+0.02%	0.39	P4M 1.8G
				UHGS:	+0.01%	+0.00%	0.99	Opt 2.4G
				SPUO12:	**+0.01%**	**+0.00%**	0.13	I7 2.93G
LDVRP	CMT79	[50,199]	C	XZKX12:	+0.48%	+0.00%	1.3	NC 1.6G
				UHGS:	**-0.28%**	**-0.33%**	2.34	Opt 2.2G
LDVRP	GWKC98	[200,483]	C	XZKX12:	+0.66%	+0.00%	3.3	NC 1.6G
				UHGS:	**-1.38%**	**-1.52%**	23.81	Opt 2.2G
PVRP	CGL97	[50,417]	C	HDH09:	+1.69%	+0.28%	3.09	P-IV 3.2G
				UHGS*:	+0.43%	+0.02%	6.78	Opt 2.4G
				CM12:	**+0.24%**	**+0.06%**	64×3.55	64×Xe 3G
MDVRP	CGL97	[50,288]	C	CM12:	+0.09%	+0.03%	64×3.28	64×Xe 3G
				S12:	+0.07%	+0.02%	11.81	I7 2.93G
				UHGS*:	**+0.08%**	**+0.00%**	5.17	Opt 2.4G
GVRP	B11	[16,262]	C	BER11:	+0.06%	—	0.01	Opt 2.4G
				MCR12:	+0.11%	—	0.34	Duo 1.83G
				UHGS:	**+0.00%**	**-0.01%**	1.53	Opt 2.4G

Unified Hybrid Genetic Search

Variant	Bench.	n	Obj.	State-of-the-art methods Author	Avg.%	Best%	T(min)	CPU
OVRP	CMT79 &F94	[50,199]	F/C	RTBI10:	0%/+0.32%	—	9.54	P-IV 2.8G
				S12:	—/+0.16%	0%/+0.00%	2.39	I7 2.93G
				UHGS:	**0%/+0.11%**	**0%/+0.00%**	1.97	Opt 2.4G
OVRP	GWKC98	[200,480]	F/C	ZK10:	0%/+0.39%	0%/+0.21%	14.79	T5500 1.66G
				S12:	0%/+0.13%	0%/+0.00%	64.07	I7 2.93G
				UHGS:	**0%/-0.11%**	**0%/-0.19%**	16.82	Opt 2.4G
VRPTW	SD88	100	F/C	RTI09:	0%/+0.11%	0%/+0.04%	17.9	Opt 2.3G
				UHGS*:	0%/+0.04%	0%/+0.01%	2.68	Xe 2.93G
				NBD10:	**0%/+0.02%**	**0%/+0.00%**	5.0	Opt 2.4G
VRPTW	HG99	[200,1000]	F/C	RTI09b:	—	+0.16%/+3.36%	270	Opt 2.3G
				NBD10:	+0.20%/+0.42%	+0.10%/+0.27%	21,7	Opt 2.4G
				UHGS*:	**+0.18%/+0.11%**	**+0.08%/-0.10%**	141	Xe 2.93G
OVRPTW	SD88	100	F/C	RTI09a:	+0.89%/+0.42%	0%/+0.24%	10.0	P-IV 3.0G
				KTDHS12:	0%/+0.79%	0%/+0.18%	10.0	Xe 2.67G
				UHGS:	**+0.09%/-0.10%**	**0%/-0.10%**	5.27	Opt 2.2G
TDVRPTW	SD88	100	F/C	KTDHS12:	+2.25%	0%	10.0	Xe 2.67G
				UHGS:	**-3.31%**	**-3.68%**	21.94	Opt 2.2G
VFMPTW	LS99	100	D	BDHMG08:	—	+0.59%	10.15	Ath 2.6G
				RT10:	+0.22%	—	16.67	P-IV 3.4G
				UHGS:	**-0.15%**	**-0.24%**	4.58	Opt 2.2G
VFMPTW	LS99	100	C	BDHMG08:	—	+0.25%	3.55	Ath 2.6G
				BPDRT09:	—	+0.17%	0.06	Duo 2.4G
				UHGS:	-0.38%	-0.49%	4.82	Opt 2.2G

Unified Hybrid Genetic Search

Variant	Bench.	n	Obj.	State-of-the-art methods Author	Avg.%	Best%	T(min)	CPU
CVRP	CMT79	[50,199]	C	GG11:	—	+0.03%	8×2.38	8×Xe 2.3G
				MB07:	+0.03%	—	2.80	P-IV 2.8G
				UHGS*:	**+0.02%**	**+0.00%**	11.90	Opt 2.4G
CVRP	GWKC98	[200,483]	C	GG11:	—	+0.29%	8×5	8×Xe 2.3G
				NB09:	+0.27%	+0.16%	21.51	Opt 2.4G
				UHGS*:	**+0.15%**	**+0.02%**	71.41	Opt 2.4G
VRPB	GJ89	[25,200]	C	ZK12:	+0.38%	+0.00%	1.09	T5500 1.67G
				GA09:	+0.09%	+0.00%	1.13	Xe 2.4G
				UHGS:	**+0.01%**	**+0.00%**	0.99	Opt 2.4G
CCVRP	CMT79	[50,199]	C	NPW10:	+0.74%	+0.28%	5.20	Core2 2G
				RL12:	+0.37%	+0.07%	2.69	Core2 2G
				UHGS:	**+0.01%**	**-0.01%**	1.42	Opt 2.2G
CCVRP	GWKC98	[200,483]	C	NPW10:	+2.03%	+1.38%	94.13	Core2 2G
				RL12:	+0.34%	+0.07%	21.11	Core2 2G
				UHGS:	**-0.14%**	**-0.23%**	17.16	Opt 2.2G
VRPSDP	SN99	[50,199]	C	SDBOF10:	+0.16%	+0.00%	256×0.37	256×Xe 2.67G
				ZTK10:	—	+0.11%	—	T5500 1.66G
				UHGS:	**+0.01%**	**+0.00%**	2.79	Opt 2.4G
VRPSDP	MG06	[100,400]	C	SDBOF10:	+0.30%	+0.17%	256×3.11	256×Xe 2.67G
				UHGS:	+0.20%	+0.07%	12.00	Opt 2.4G
				S12 :	**+0.08%**	**+0.00%**	7.23	I7 2.93G

Some References

References and collaborators:

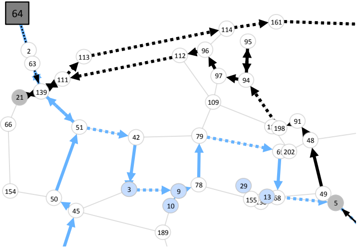
1. T. Vidal, T.G. Crainic, M. Gendreau, N. Lahrichi, W. Rei, A hybrid genetic algorithm for multidepot and periodic vehicle routing problems, Oper. Res. 60 (2012) 611–624.
2. T. Vidal, T.G. Crainic, M. Gendreau, C. Prins, Timing problems and algorithms: Time decisions for sequences of activities, Networks. 65 (2015) 102–128.
3. T. Vidal, T.G. Crainic, M. Gendreau, C. Prins, Heuristics for multi-attribute vehicle routing problems: A survey and synthesis, Eur. J. Oper. Res. 231 (2013) 1–21.
4. T. Vidal, T.G. Crainic, M. Gendreau, C. Prins, A unified solution framework for multi-attribute vehicle routing problems, Eur. J. Oper. Res. 234 (2014) 658–673.

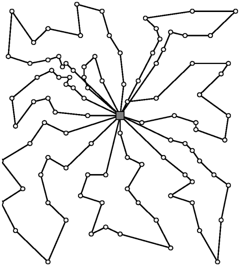
Contents

Contents

Challenges

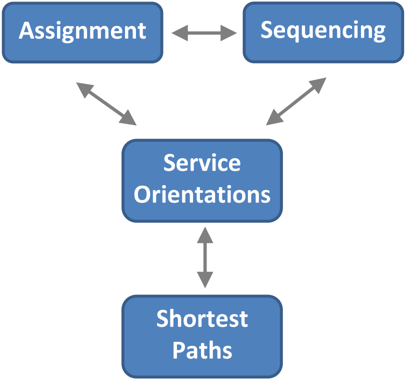
- Arc routing for home delivery, snow plowing, refuse collection, postal services, among others.
- Lead to additional challenges:
  - ⇒ *Deciding* on travel directions for services on edges
  - ⇒ Shortest path between services are *conditioned* by service orientations
    (may also need to include some additional aspects such as turn penalties or delays at intersections).





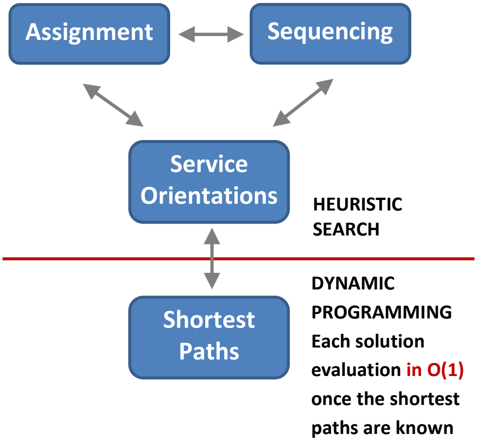
Challenges

- 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

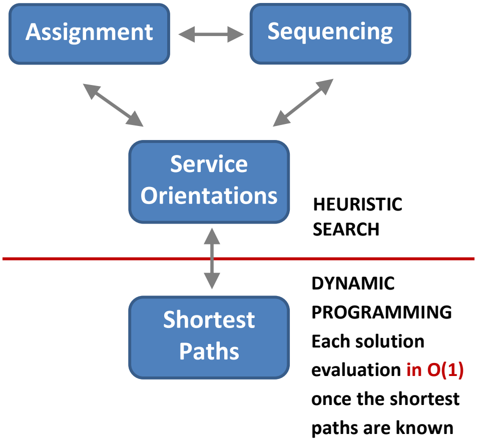
- Most **recent CARP heuristics rely on several enumerative neighborhood classes** to optimize assignment, sequencing and service orientation decisions
  - See, e.g. Brandão and Eglese (2008); Usberti et al. (2013); Dell'Amico et al. (2016)...
  - Shortest paths between node extremities have been pre-processed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



$\Rightarrow$ **This is, however, not the only option.**

A question of neighborhood

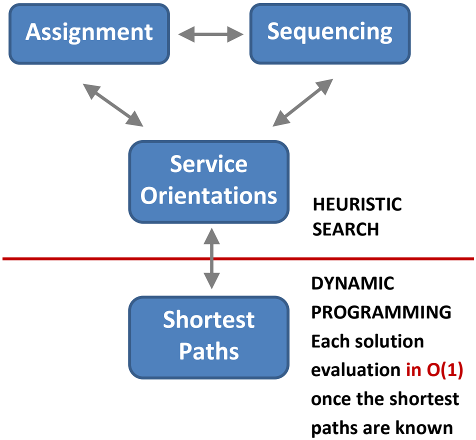
- Most **recent CARP heuristics rely on several enumerative neighborhood classes** to optimize assignment, sequencing and service orientation decisions
  - See, e.g. Brandão and Eglese (2008); Usberti et al. (2013); Dell'Amico et al. (2016)...
  - Shortest paths between node extremities have been pre-processed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



⇒ **This is, however, not the only option.**

A question of neighborhood

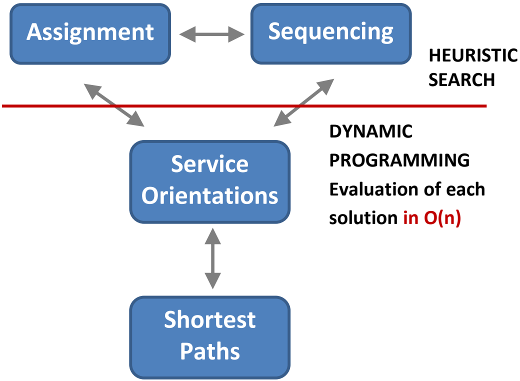
- Most **recent CARP heuristics rely on several enumerative neighborhood classes** to optimize assignment, sequencing and service orientation decisions
  - See, e.g. Brandão and Eglese (2008); Usberti et al. (2013); Dell'Amico et al. (2016)...
  - Shortest paths between node extremities have been pre-processed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



⇒ **This is, however, not the only option.**

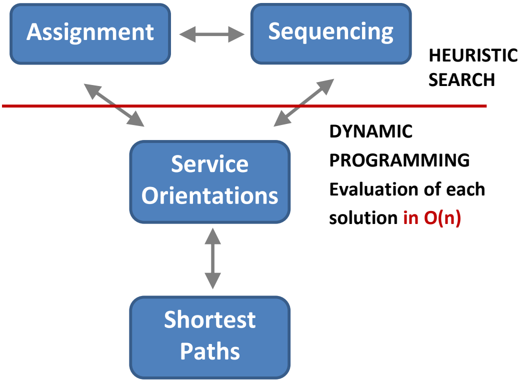
A question of neighborhood

- 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.



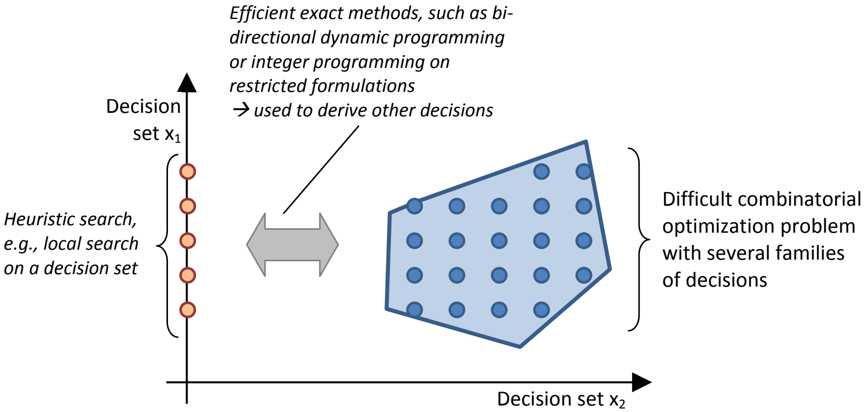
A question of neighborhood

- 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.



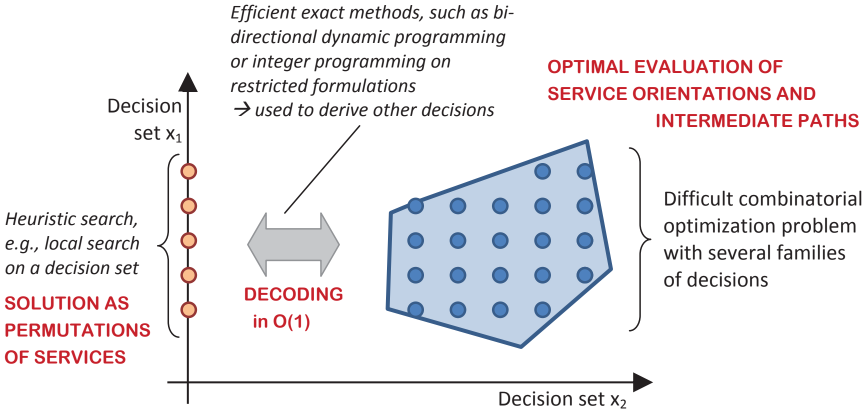
A question of neighborhood

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



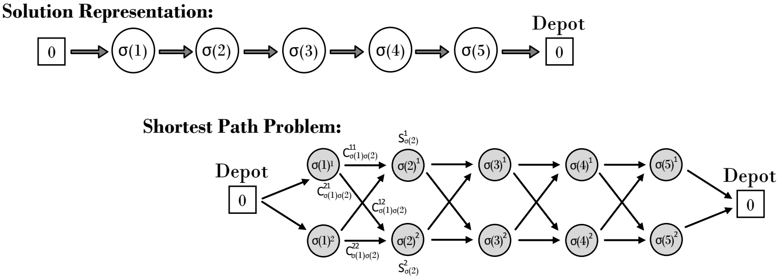
A question of neighborhood

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



Solution representation and decoding

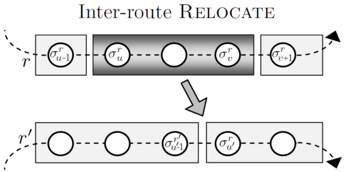
- How to decode/evaluate a solution = deriving optimal orientations for the services ?
  - ⇒ Simple dynamic programming subproblem (Beullens et al., 2003; Wøhlk, 2003, 2004):

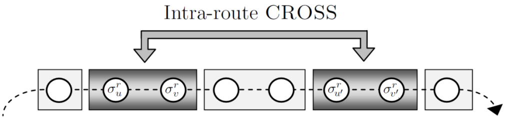


- Each service represented by two nodes, one for each orientation. Travel costs c_{ij}^{kl} between (i, j) are conditioned by the orientations (k, l) for departure and arrival.

Seeking low complexity for solution evaluations

- 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 σ, for any (entry, exit) direction pair (k,l)

Initialization

For σ_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:

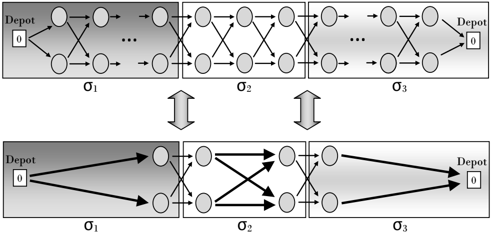
$$C(\sigma_1 \oplus \sigma_2)[k,l] = \min_{x,y} \left\{ C(\sigma_1)[k,x] + c^{xy}_{\sigma_1(|\sigma_1|)\sigma_2(1)} + C(\sigma_2)[y,l] \right\}$$

Seeking low complexity for solution evaluations

- **Pre-processing partial shortest paths in the incumbent solution** – in $\mathcal{O}(n^2)$ 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...
  $\Rightarrow$ 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 σ
- 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 σ 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 $\sigma = \sigma_1 \oplus \dots \oplus \sigma_X$ composed of a concatenation of X sequences:

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

- 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 σ 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 $\sigma = \sigma_1 \oplus \dots \oplus \sigma_X$ composed of a concatenation of X sequences:

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

- 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!**

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:							
GDB	(23)	Golden et al. (1983)	0	[11,55]	0	[11,55]	Random graphs; Only required edges
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
EGL-L	(10)	Brandão and E. (2008)	0	[347,375]	0	[347,375]	Larger winter-gritting application
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

Experimental setting

- To prevent any possible over-tuning
  $\Rightarrow$ using the original parameters of the metaheuristics

- Single core: Xeon 3.07 GHz CPU with 16 GB of RAM

- Single termination criterion on all instances
  $\Rightarrow$ scaled to reach a similar CPU time as previous competitive algorithms.

Experimental setting

- For each benchmark set, we collected the best three solution methods in the literature (some are heavily tailored for specific benchmark sets).

BE08	Brandão and Eglese (2008)	HKSG12	Hasle et al. (2012)	MTY09	Mei et al. (2009)
BLMV14	Bosco et al. (2014)	LPR01	Lacomme et al. (2001)	PDHM08	Polacek et al. (2008)
BMCV03	Beullens et al. (2003)	MLY14	Mei et al. (2014)	TMY09	Tang et al. (2009)
DHDI14	Dell'Amico et al. (2016)	MPS13	Martinelli et al. (2013)	UFF13	Usberti et al. (2013)

- 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).

Experimental setting

- Reporting the average and best solution on 10 runs.
- All Gap(%) values measured from the current best known solutions (BKS)
- Warning – time measures for some previous algorithms: using known optimal solutions to trigger termination, or reporting the time to reach the best solution
  - Dependent on exogenous information
  - Not the complete search time
- Hence, two columns for time measures:
  $\Rightarrow$ "T" for total CPU time when available,
  $\Rightarrow$ "T*" for time to reach final solution.

Comparison with previous literature

Variant	Bench.	n	Author	Runs	Avg.	Best	T	T*	CPU
CARP	GDB	[11,55]	TMY09	*30*	0.009%	0.000%	0.11	—	Xe 2.0G
			BMCV03	*1*	0.000%	—	—	0.03	P-II 500M
			MTY09	*1*	0.000%	—	—	0.01	Xe 2.0G
			ILS	*10*	0.002%	0.000%	0.16	0.03	Xe 3.07G
			UHGS	*10*	**0.000%**	**0.000%**	0.22	0.01	Xe 3.07G
	VAL	[39,97]	MTY09	*1*	0.142%	—	—	0.11	Xe 2.0G
			LPR01	*1*	0.126%	—	2.00	—	P-III 500M
			BMCV03	*1*	0.060%	—	—	1.36	P-II 500M
			ILS	*10*	0.054%	0.024%	0.68	0.16	Xe 3.07G
			UHGS	*10*	**0.048%**	**0.021%**	0.82	0.08	Xe 3.07G
	BMCV	[28,121]	BE08	*1*	0.156%	—	—	1.08	P-M 1.4G
			MTY09	*1*	0.073%	—	—	0.35	Xe 2.0G
			BMCV03	*1*	0.036%	—	2.57	—	P-II 450M
			ILS	*10*	0.027%	0.000%	0.82	0.22	Xe 3.07G
			UHGS	*10*	**0.007%**	**0.000%**	0.87	0.11	Xe 3.07G
	EGL	[51,190]	PDHM08	*10*	0.624%	—	30.0	8.39	P-IV 3.6G
			UFF13	*15*	0.560%	0.206%	13.3	—	I4 3.0G
			MTY09	*1*	0.553%	—	—	2.10	Xe 2.0G
			ILS	*10*	0.236%	0.106%	2.35	1.33	Xe 3.07G
			UHGS	*10*	**0.153%**	**0.058%**	4.76	3.14	Xe 3.07G
	EGL-L	[347,375]	BE08	*1*	4.679%	—	—	17.0	P-M 1.4G
			MPS13	*10*	2.950%	2.523%	20.7	—	I5 3.2G
			MLY14	*30*	1.603%	0.895%	33.4	—	I7 3.4G
			ILS	*10*	0.880%	0.598%	23.6	15.4	Xe 3.07G
			UHGS	*10*	**0.645%**	**0.237%**	36.5	27.5	Xe 3.07G

Comparison with previous literature

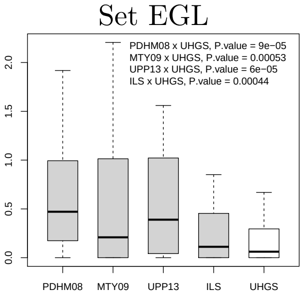
Variant	Bench.	n	Author	Runs	Avg.	Best	T	T*	CPU
MCGRP	MGGDB	[8,48]	BLMV14	*1*	1.342%	—	0.31	—	Xe 3.0G
			DHDI14	*1*	0.018%	—	60.0	0.86	CPU 3G
			ILS	*10*	0.010%	0.000%	0.13	0.03	Xe 3.07G
			UHGS	*10*	**0.015%**	**0.000%**	0.16	0.01	Xe 3.07G
	MGVAL	[36,129]	BLMV14	*1*	2.620%	—	16.7	—	Xe 3.0G
			DHDI14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

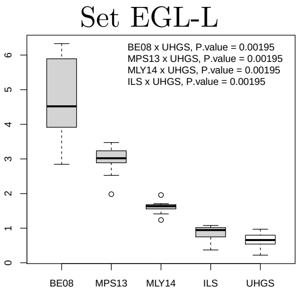
Comparison with previous literature

- New neighborhoods lead to much better solutions.
- ILS already produces better solutions than previous literature, and UHGS goes further in performance $\Rightarrow$ 0.503% and 0.958% improvement on the large instance sets
- Average standard deviation in $[0.000\%, 0.292\%]$
- On the CARP benchmark sets, 187/191 BKS have been matched or improved. 153/155 known optimal solutions were found
- For the MCGRP, 408/409 BKS have been matched or improved. All 217 known optimal solutions found.

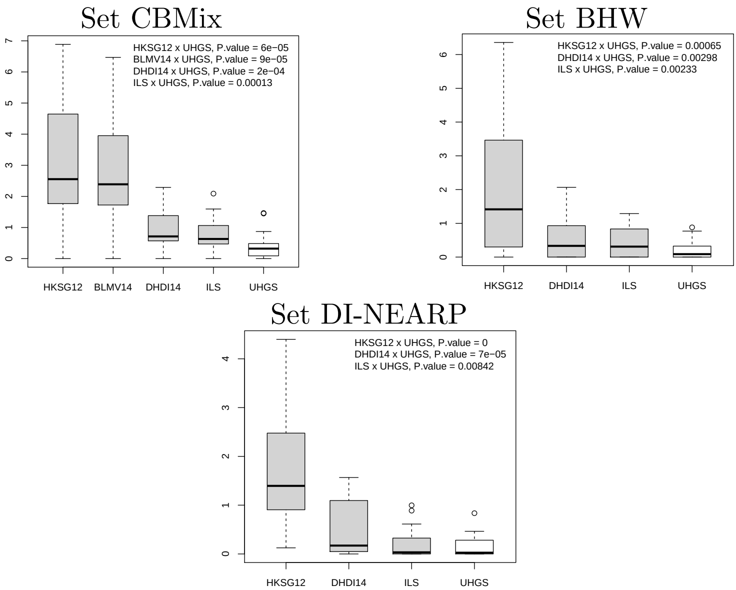
Comparison with previous literature

- 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

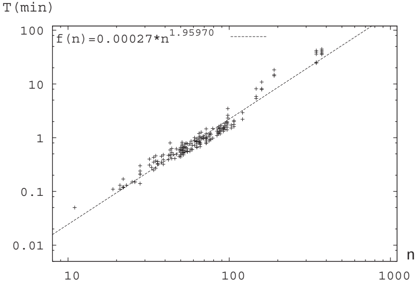




Comparison with previous literature



- 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.



- A linear fit, with a least square regression, has been performed on the sample after logarithmic transformation:
  $\Rightarrow$ CPU time appears to grow in $\mathcal{O}(n^2)$

To reduce or not to reduce

- Previous slides: investigated whether methods using combined neighborhoods – with optimal choices of service orientations – can outperform methods based on more traditional neighborhoods

- Now analyzing whether relying on a problem reduction from CARP to CVRP (Martinelli et al., 2013) with a classical routing metaheuristic can be profitable.

- The reduction increases the number of services by $\times 2$.
  - Half of the edges of a CVRP solution, with a large fixed negative cost, directly determine the service orientations in the associated CARP solution.

To reduce or not to reduce

- Applied the same ILS and UHGS on the transformed instances, now using a classical move evaluation for the CVRP.

	Gap(%)		**T(min)**	
	ILS	ILS_{CVRP}	ILS	ILS_{CVRP}
GDB	0.002%	0.000%	0.16	0.59
VAL	0.054%	0.061%	0.68	2.39
BMCV	0.027%	0.044%	0.82	2.79
EGL	0.236%	0.345%	2.35	8.50
EGL-L	0.880%	1.411%	23.6	60.0

	Gap(%)		**T(min)**	
	UHGS	$\text{UHGS}_{\text{CVRP}}$	UHGS	$\text{UHGS}_{\text{CVRP}}$
GDB	0.000%	0.000%	0.22	0.72
VAL	0.048%	0.048%	0.82	2.98
BMCV	0.007%	0.014%	0.87	3.02
EGL	0.153%	0.200%	4.76	12.65
EGL-L	0.645%	1.001%	36.5	59.7

Some References

References and collaborators:

1. T. Vidal, Node, edge, arc routing and turn penalties : Multiple problems – One neighborhood extension, Oper. Res. 65 (2017) 992–1010.

Contents

Conclusions and Perspectives

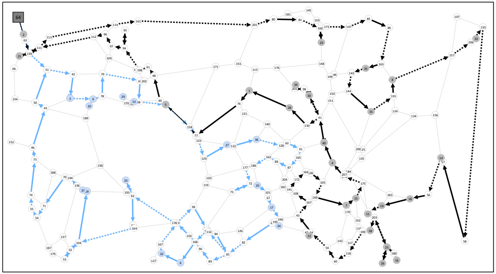
- Unified framework for vehicle routing problems, don't need to reinvent the wheel for each new variant. Generality does not necessarily impede efficiency for a large class of problems.
- **Understanding the structure of the problems** is critical for the design of efficient methods
- Structural problem decompositions allow to relegate difficult decision classes (e.g., customer selection, edge orientations etc...) inside (modular) route-evaluation operators
- Efficient route 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

Conclusions and Perspectives

- Perspectives: keep on focusing on hard VRP variants, focus on **problem structure**, **computational complexity** and **neighborhood search**. Major breakthroughs are still possible around those research lines.
- Perspectives: harnessing the abilities of machine learning techniques to help speeding-up the convergence towards good solutions ?

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`

For further reading I

Bach, L., G. Hasle, S. Wøhlk. 2013. A lower bound for the node, edge, and arc routing problem. *Computers & Operations Research* **40**(4) 943–952.

Balas, E., N. Simonetti. 2001. Linear time dynamic-programming algorithms for new classes of restricted TSPs: A computational study. *INFORMS Journal on Computing* **13**(1) 56–75.

Benavent, E., V. Campos, A. Corberán, E. Mota. 1992. The capacitated arc routing problem: Lower bounds. *Networks* **22**(7) 669–690.

Beullens, P., L. Muyldermans, D. Cattrysse, D. Van Oudheusden. 2003. A guided local search heuristic for the capacitated arc routing problem. *European Journal of Operational Research* **147**(3) 629–643.

Bosco, A., D. Laganà, R. Musmanno, F. Vocaturo. 2012. Modeling and solving the mixed capacitated general routing problem. *Optimization Letters* **7**(7) 1451–1469.

Bosco, A., D. Laganà, R. Musmanno, F Vocaturo. 2014. A matheuristic algorithm for the mixed capacitated general routing problem. *Networks* **64**(4) 262–281.

Brandão, J., R. Eglese. 2008. A deterministic tabu search algorithm for the capacitated arc routing problem. *Computers & Operations Research* **35**(4) 1112–1126.

Dell'Amico, Mauro, José Carlos Díaz Díaz, Geir Hasle, Manuel Iori. 2016. An Adaptive Iterated Local Search for the Mixed Capacitated General Routing Problem. *Transportation Science* .

For further reading II

Golden, B.L., J.S. DeArmon, E.K. Baker. 1983. Computational experiments with algorithms for a class of routing problems. *Computers & Operations Research* **10**(1) 47–59.

Hasle, G., O. Kloster, M. Smedsrud, K. Gaze. 2012. Experiments on the node, edge, and arc routing problem. Tech. rep., SINTEF, Oslo, Norway.

Irnich, S. 2008. Solution of real-world postman problems. *European Journal of Operational Research* **190**(1) 52–67.

Lacomme, P., C. Prins, W. Ramdane-Chérif. 2001. A genetic algorithm for the capacitated arc routing problem and its extensions. *Applications of Evolutionary Computing* 473–483.

Li, L.Y.O., R.W. Eglese. 1996. An interactive algorithm for vehicle routeing for winter-gritting. *Journal of the Operational Research Society* **47**(2) 217–228.

Martinelli, R., M. Poggi, A. Subramanian. 2013. Improved bounds for large scale capacitated arc routing problem. *Computers & Operations Research* **40**(8) 2145–2160.

Mei, Y., X. Li, X. Yao. 2014. Cooperative co-evolution with route distance grouping for large-scale capacitated arc routing problems. *IEEE Transactions on Evolutionary Computation* **18**(3) 435–449.

Mei, Y., K. Tang, X. Yao. 2009. A global repair operator for capacitated arc routing problem. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics* **39**(3) 723–734.

For further reading III

Muyldermans, L., P. Beullens, D. Cattrysse, D. Van Oudheusden. 2005. Exploring variants of 2-opt and 3-opt for the general routing problem. *Operations Research* **53**(6) 982–995.

Polacek, M., K.F. Doerner, R.F. Hartl, V. Maniezzo. 2008. A variable neighborhood search for the capacitated arc routing problem with intermediate facilities. *Journal of Heuristics* **14**(5) 405–423.

Prins, C., S. Bouchenoua. 2005. A memetic algorithm solving the VRP, the CARP, and more general routing problems with nodes, edges and arcs. W. Hart, N. Krasnogor, J. Smith, eds., *Recent advances in memetic algorithms*. Springer, 65–85.

Tang, K., Y. Mei, X. Yao. 2009. Memetic algorithm with extended neighborhood search for capacitated arc routing problems. *IEEE Transactions on Evolutionary Computation* **13**(5) 1151–1166.

Uchoa, E., D. Pecin, A. Pessoa, M. Poggi. 2013. Advances on the Exact Approaches for the CVRP. *EURO'13 Conference, Rome, Italy*.

Usberti, F.L., P.M. França, A.L.M. França. 2013. GRASP with evolutionary path-relinking for the capacitated arc routing problem. *Computers & Operations Research* **40**(12) 3206–3217.

Vidal, T., T.G. Crainic, M. Gendreau, C. Prins. 2014. A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research* **234**(3) 658–673.

For further reading IV

Wøhlk, S. 2003. Simulated annealing for the capacitated arc routing problem, using an online formulation. Tech. rep., University of Southern Denmark.

Wøhlk, S. 2004. Combining dynamic programming and simulated annealing. Tech. rep., University of Southern Denmark.