Multi-attribute vehicle routing : unified metaheuristics and timing sub-problems

Thibaut Vidal

utt – université de technologie Troyes

Université de Montréal

CIRRELT – Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Conseil Regional Champagne Ardenne

Chaire de recherche industrielle du **CRSNG** en management logistique
NSERC Industrial Research Chair in Logistics Management

CORMSIS Seminar, University of Southampton, February 21th, 2013

Context of this research

- Joint work with

  Christian PRINS, *Université de Technologie de Troyes, France*

  Teodor Gabriel CRAINIC, *Université de Québec à Montréal, Canada*

  and Michel GENDREAU, *Ecole Polytechnique, Montréal, Canada*

Outline of the presentation

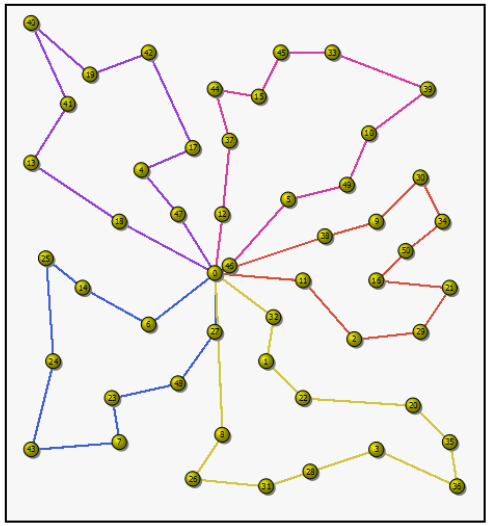
- **I) A general-purpose solver for multi-attribute vehicle routing problems**
  - Multi-attribute vehicle routing problems
  - An efficient and unified local search for MAVRPs
  - A Unified Hybrid Genetic Search (UHGS) for MAVRPs
  - Computational experiments

- **II) Timing problems and algorithms**
  - Several applications presenting similar *timing* issues
  - Classification and notation
  - Reductions
  - Timing Re-optimization

Multi-attribute vehicle routing problems (MAVRPs)

- Classical "vehicle routing problems (VRP)"
  → plethora of exact and heuristic methods

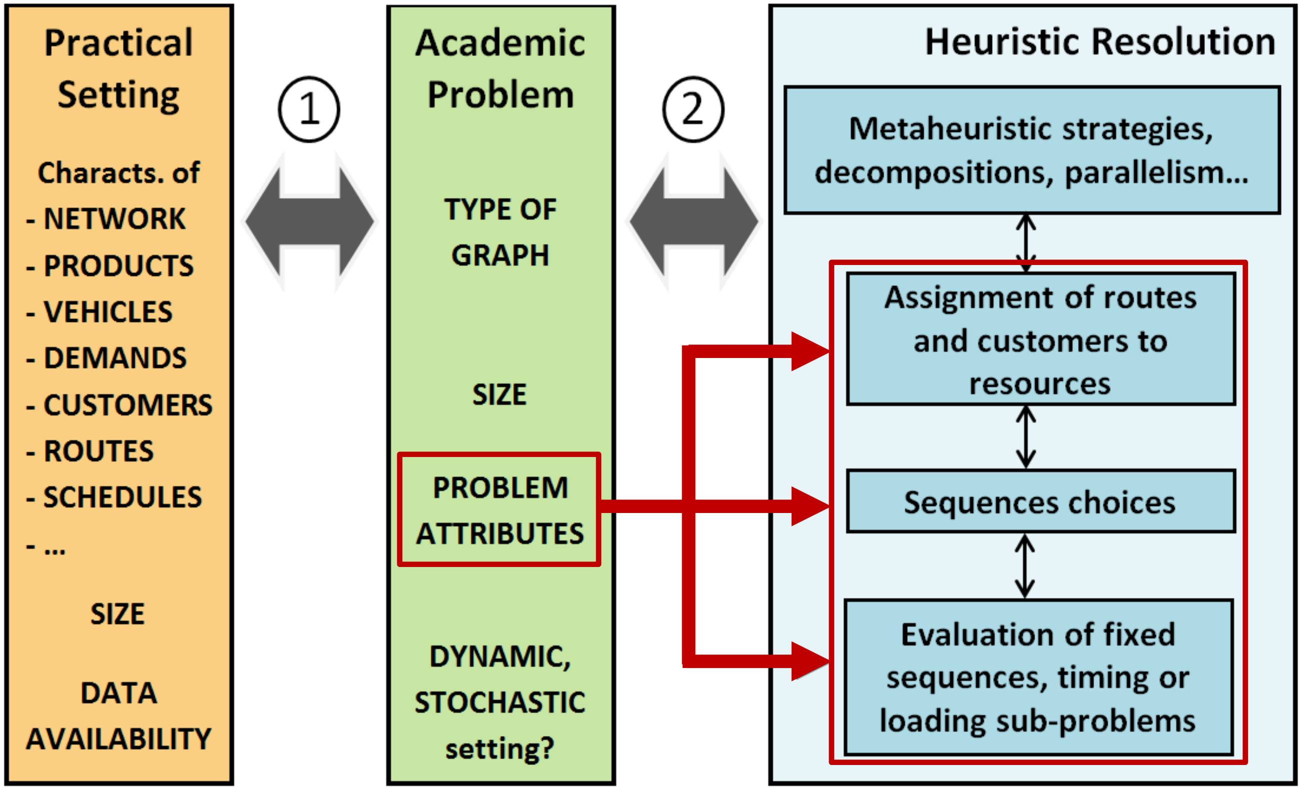
- **Challenges** related to the resolution of **VRP variants with additional *attributes*** (multi-attribute VRPs, MAVRPs)
  - modeling the specificities of application cases, customers requirements, network and vehicle specificities, operators abilities…
  - Combining **several attributes** together can lead to highly complex **rich VRPs.**
  - Dramatic increase in the literature dedicated to specific VRP variants.



Multi-attribute vehicle routing problems (MAVRPs)

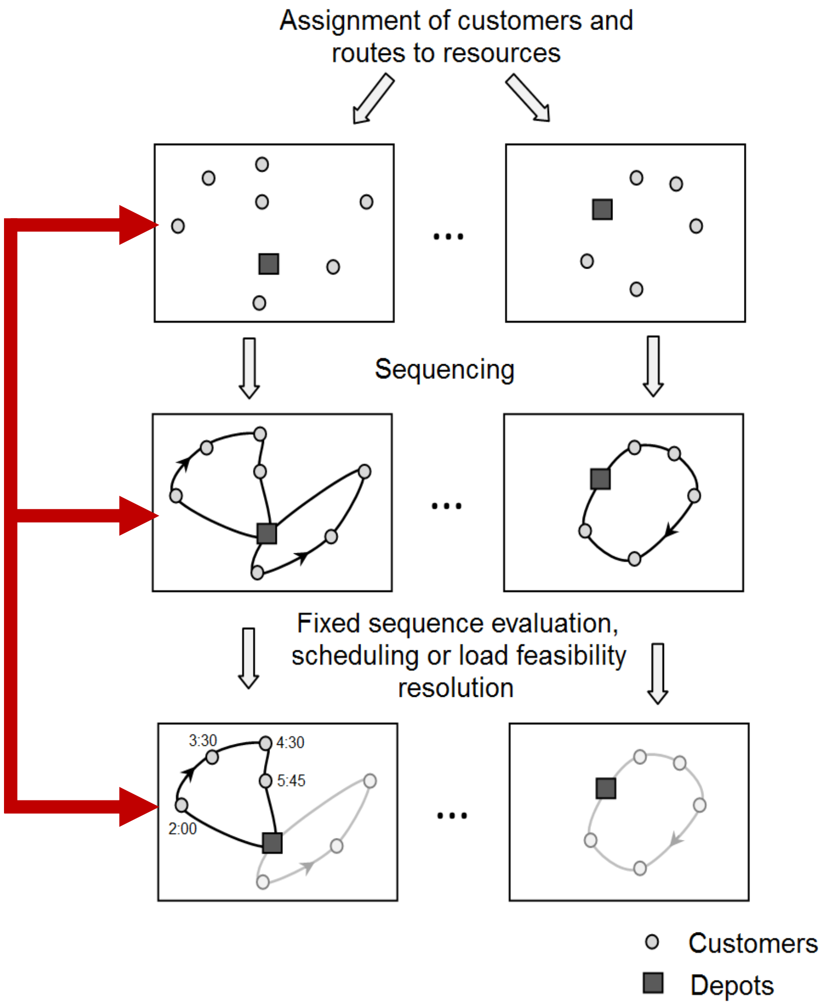
- **General purpose solvers / unified methods:** address a wide range of problems without need for extensive adaptation or user expertise.
  - **Necessary tools** for 1) the timely application of current optimization methods to industrial settings. 2) for assessing the scope of application of elements of methodology
  - Few/none of them in the academic VRP literature. Some algorithms reporting high quality solutions on several MAVRPs: UTS (Cordeau et al. 1997,2001), ALNS (Pisinger and Ropke 2006), ILS (Subramanian et al. 2013).
    - *7 MAVRP with a single code*
    - But "**curse of richness**"

- We classified attributes into three categories related to their impact on VRP resolution methods :



Classification & Proposed Methodology

- **ASSIGN ATTRIBUTES**: impacting the assignment of customers and routes
  - **Periodic, Multi-Depot, Heterogeneous Fleet, Location Routing…**
- **SEQ ATTRIBUTES:** impacting the nature of the network and the sequences
  - **P&D, Backhauls, Two Echelon, Truck-and-Trailer…**
- **EVAL ATTRIBUTES:** impacting the evaluation of fixed routes
  - **Time windows, Time-dep. travel time, Loading constraints, HOS regulations Lunch breaks, Load-Dependent costs…**

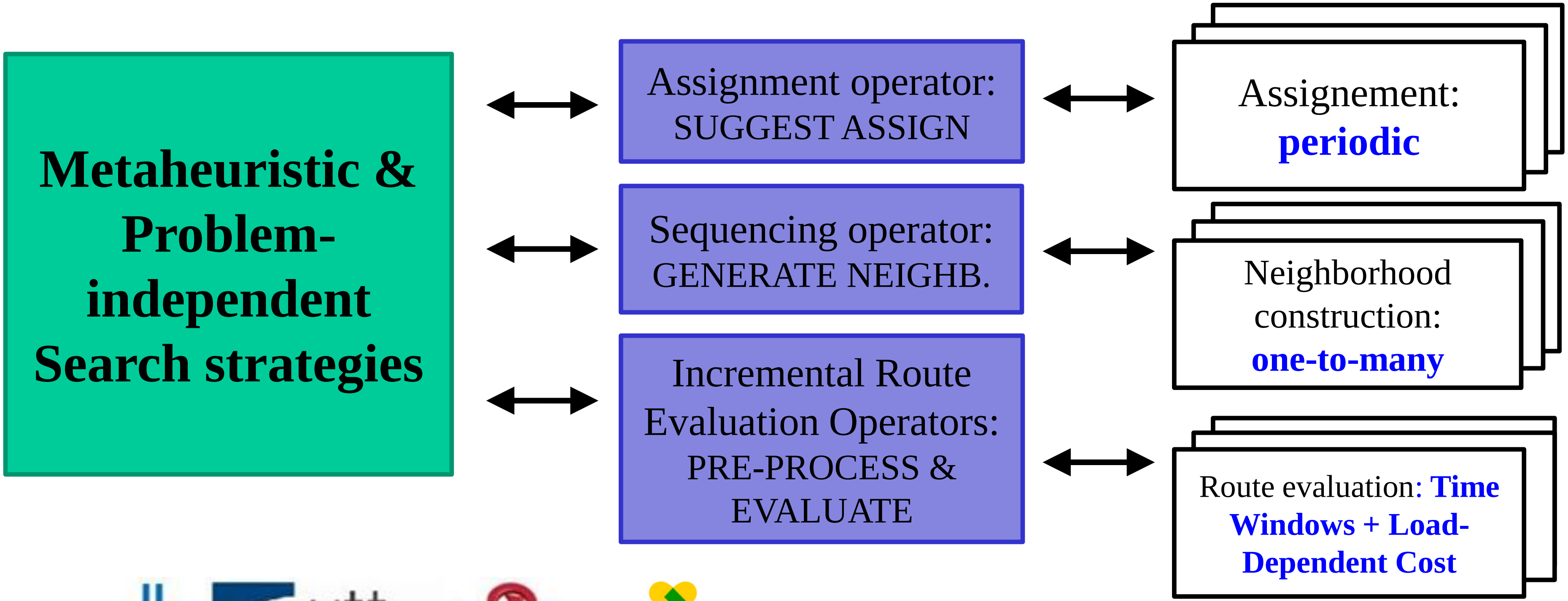


Classification & Proposed Methodology

- Challenge: **Achieving both genericity and efficiency**
  - Still need to address the problem → but relegating problem-specificities to small modular components
  - Each separate MAVRP shall be still addressed with state-of-the-art solution evaluation and search procedures
  - Not dealing with "dummy" attributes

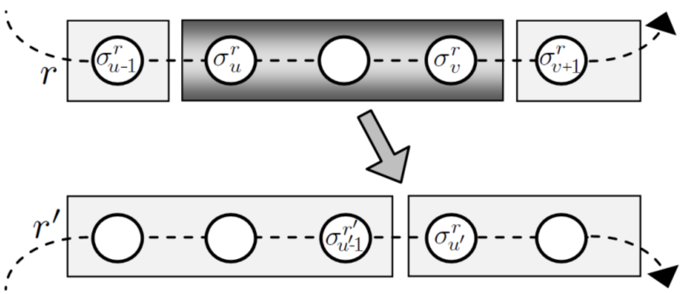
Attribute-based modular design

- Unified framework:
  - 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 RE operators.

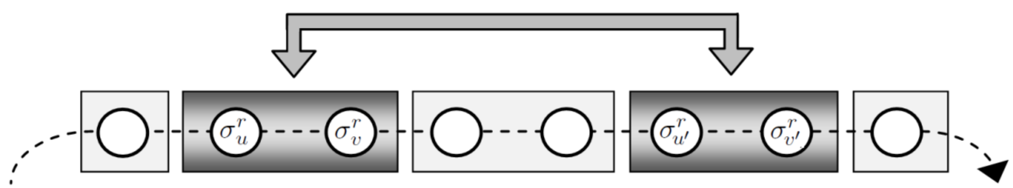


- Route Evaluation Operators based on re-optimization
  - Main Property : Any local-search move involving a bounded number of node relocations or arc exchanges can be assimilated to a concatenation of a bounded number of sub-sequences.
  - The same subsequences appear many times during different moves

Inter-route RELOCATE



Intra-route CROSS



  - Data preprocessing on sub-sequences to speed up the search (Savelsbergh 1985,1992 …)
  - The route evaluation operator must allow for such preprocessing.

Route Evaluation Operators based on re-optimization

- Main Property : 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.
- We compute characteristic information on subsequences by induction on the concatenation operator $\oplus$. Four tasks are necessary:
  - **Init**: Initialize the characteristic information on a single node
  - **Forw:** Append an additional node at the end of an existing sequence, and derive the resulting characteristic information
  - **Back:** Append an additional node at the beginning of an existing sequence, and derive the resulting characteristic information
  - **Eval:** Evaluate a move as a concatenation of a bounded number of subsequences using the characteristic information of each one.

Route evaluation operators examples

- **Example 1)** Route evaluation operators for **distance and capacity constraints**

What is managed ? → Partial loads L(σ) and distance D(σ)

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

Forw and Back → increment L(σ) and D(σ)

Eval → compute the data by induction on the concatenation operator

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

Route evaluation operators examples

- **Example 2)** Route evaluation operators for **cumulated arrival time objectives**

What is managed ? → Travel time D(σ), Cumulated arrival time C(σ), Delay Cost W(σ) associated to one unit of delay in starting time

Init → 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.

Forw & Back & Eval → induction on the concatenation operator:

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

Route evaluation operators examples

- **Example 3)** Route evaluation operators for **time windows (and route duration constraints)**

What is managed ? → Travel time and service time T(σ), earliest feasible completion time E(σ), latest feasible starting date L(σ), statement of feasibility F(σ).

Init → 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 $F(\sigma_0) = \textit{true}$.

Forw & Back & Eval → induction on the concatenation operator:

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

Route evaluation operators examples

- **Example 4)** Route evaluation operators for **lunch break positioning in presence of time-window constraints**

What is managed ? → Same set of data ($T(\sigma)$, $E(\sigma)$, $L(\sigma)$, and $F(\sigma)$) as in the TW case, and it is duplicated to also provide $T'(\sigma)$, $E'(\sigma)$, $L'(\sigma)$, and $F'(\sigma)$ for the sequence where exactly one lunch break was inserted.

Init → As previously for $T(\sigma_0)$, $E(\sigma_0)$, $L(\sigma_0)$, and $F(\sigma_0)$. Furthermore, $T'(\sigma_0) = +\infty$, $E'(\sigma_0) = +\infty$, $L'(\sigma_0) = 0$, and $F'(\sigma_0) = \textit{false}$.

Forw & Back & Eval → induction on the concatenation operator, see next page for the equations.

Route evaluation operators examples

- **Example 4)** Route evaluation operators for **lunch break positioning in presence of time-window constraints**

$$E'(\sigma_1 \oplus \sigma_2) = \min(\{E'_{\text{case i}} | F'_{\text{case i}} = true\} \cup +\infty)$$
$$L'(\sigma_1 \oplus \sigma_2) = \max(\{L'_{\text{case i}} | F'_{\text{case i}} = true\} \cup -\infty)$$
$$F'(\sigma_1 \oplus \sigma_2) = F'_{\text{case 1}} \vee F'_{\text{case 2}} \vee F'_{\text{case 3}}$$

$$E'_{\text{case 1}} = \max\{E'(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} + T(\sigma_2), E(\sigma_2)\}$$
$$E'_{\text{case 2}} = \max\{E(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} + s_{\text{LB}} + T(\sigma_2), e_{\text{LB}} + s_{\text{LB}} + T(\sigma_2), E(\sigma_2)\}$$
$$E'_{\text{case 3}} = \max\{E(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} + T'(\sigma_2), E'(\sigma_2)\}$$
$$L'_{\text{case 1}} = \min\{L'(\sigma_1), L(\sigma_2) - p_{\sigma_1(|\sigma_1|)\sigma_2(1)} - T'(\sigma_1)\}$$
$$L'_{\text{case 2}} = \min\{L(\sigma_1), l_{\text{LB}} - T(\sigma_1), L(\sigma_2) - p_{\sigma_1(|\sigma_1|)\sigma_2(1)} - s_{\text{LB}} - T(\sigma_1)\}$$
$$L'_{\text{case 3}} = \min\{L(\sigma_1), L'(\sigma_2) - p_{\sigma_1(|\sigma_1|)\sigma_2(1)} - T(\sigma_1)\}$$
$$F'_{\text{case 1}} = F'(\sigma_1) \wedge F(\sigma_2) \wedge (E'(\sigma_1) + p_{\sigma_1(|\sigma_1|)\sigma_2(1)} \leq L(\sigma_2))$$
$$F'_{\text{case 2}} = F(\sigma_1) \wedge F(\sigma_2) \wedge (E(\sigma_1) \leq l_{\text{LB}}) \wedge (E(\sigma_1) + s_{\text{LB}} + p_{\sigma_1(|\sigma_1|)\sigma_2(1)} \leq L(\sigma_2))$$
$$F'_{\text{case 3}} = F(\sigma_1) \wedge F'(\sigma_2) \wedge (E(\sigma_1) + p_{\sigma_1(|\sigma_1|)\sigma_2(1)} \leq L'(\sigma_2))$$

Route evaluation operators examples

- **Example 5)** Route evaluation operators for **soft and general time windows**

What is managed ? → Minimum cost F(σ)(t) to process the sequence σ while starting the last service before time t, minimum cost B(σ)(t) to process the sequence σ after time t.

Init → For a sequence σ_0 with a single visit v_i characterized by a service cost function $c_i(t)$, $F(\sigma_0)(t) = \min_{(x \leq t)} ci(x)$ and $B(\sigma_0)(t) = \min_{(x \geq t)} c_i(x)$.

Forw & Back →

$$F(\sigma \oplus v_i)(t) = \min_{0 \leq x \leq t} \{c_i(x) + F(\sigma)(x - s_{\sigma(|\sigma|)} - d_{\sigma(|\sigma|),i})\}$$

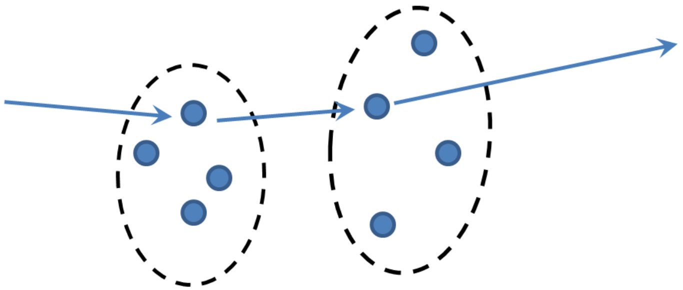
$$B(v_i \oplus \sigma)(t) = \min_{x \geq t}\{c_i(t) + B(\sigma)(x + s_i + d_{i,\sigma(1)})\}$$

Eval 2 →

$$Z^*(\sigma_1 \oplus \sigma_2) = \min_{x \geq 0}\{F(\sigma_1)(x) + B(\sigma_2)(x + s_{\sigma_1(|\sigma_1|)} + d_{\sigma_1(|\sigma_1|)\sigma_2(1)})\}$$

Route evaluation operators examples

- **Example 6)** Route evaluation operators for **the generalized VRP :**



What is managed ? → The shortest path S(σ)[i,j] inside the sequence σ starting at the location *i* of the starting group and finishing at location *j* of the ending group.

Init → For a sequence σ_0 with a single visit v_i , S(σ)[i,j] = +∞ if i ≠ j, and S(σ)[i,i] = 0.

Forw & Back & Eval → induction on the concatenation operator:

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

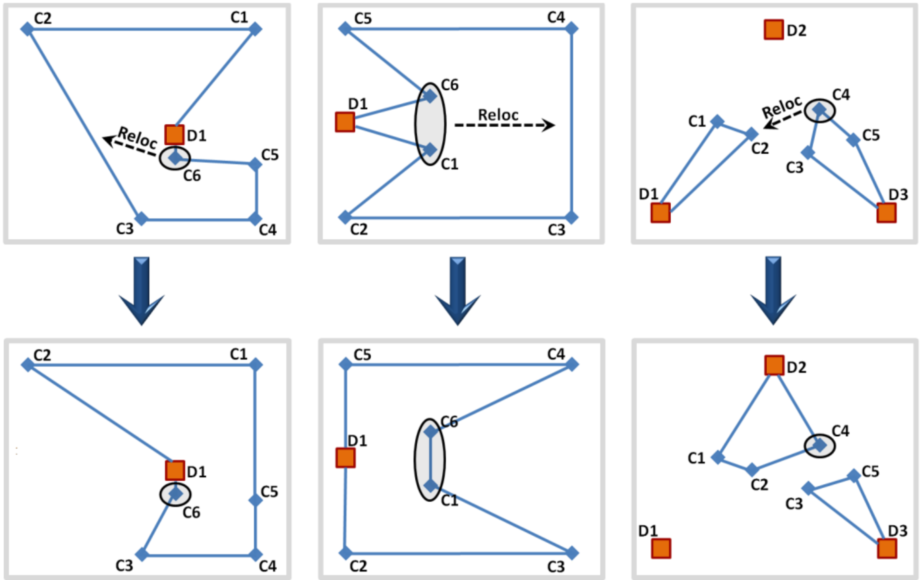
$$\forall i \in \{1,\ldots,\lambda_{\sigma_1(1)}\}, \forall j \in \{1,\ldots,\lambda_{\sigma_2(|\sigma_2|)}\}$$

Route evaluation operators examples

- **Other examples:** many other route evaluation operators have been designed for other vehicle routing variants.

  - Some advanced route evaluation operators, based on dynamic programming, enable to **implicitly and optimally decide the first visit in the route (optimal rotation), the customer-to-depot or customer-to-vehicle type assignment, or the selection of customers in a prize-collecting setting.**

  - See for further examples:
    Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2012). Implicit Depot Assignments and Rotations in Vehicle Routing Heuristics.
    Submitted to EJOR.
    Tech. Rep. 2012, CIRRELT.



An efficient and unified local search for MAVRPs

- Generic local-search based on route evaluation operators

Algorithm 1 Unified local search based on route evaluation operators

Detect the good combination of evaluation operators relatively to the problem attributes
Build re-optimization data on subsequences using the INIT, FORW and BACK operators.
while some improving moves exist in the neighborhood $\mathcal{N}$ **do**
  for each move μ_i in $\mathcal{N}$ **do**
    for each route r_j^μ produced by the move **do**
      Determine the k sub-sequences $[\sigma_1, \ldots, \sigma_k]$ that are concatenated to produce r_j^μ
      if $k = 2$, then NEWCOST(r) = EVAL2(σ_1,σ_2)
      else if $k > 2$, then NEWCOST(r) = EVALN$(\sigma_1,\ldots,\sigma_k)$
    if ACCEPTCRITERIA(μ_i) **then** perform the move μ and update the re-optimization data on for each route r_j^μ using the INIT, FORW and BACK operators.

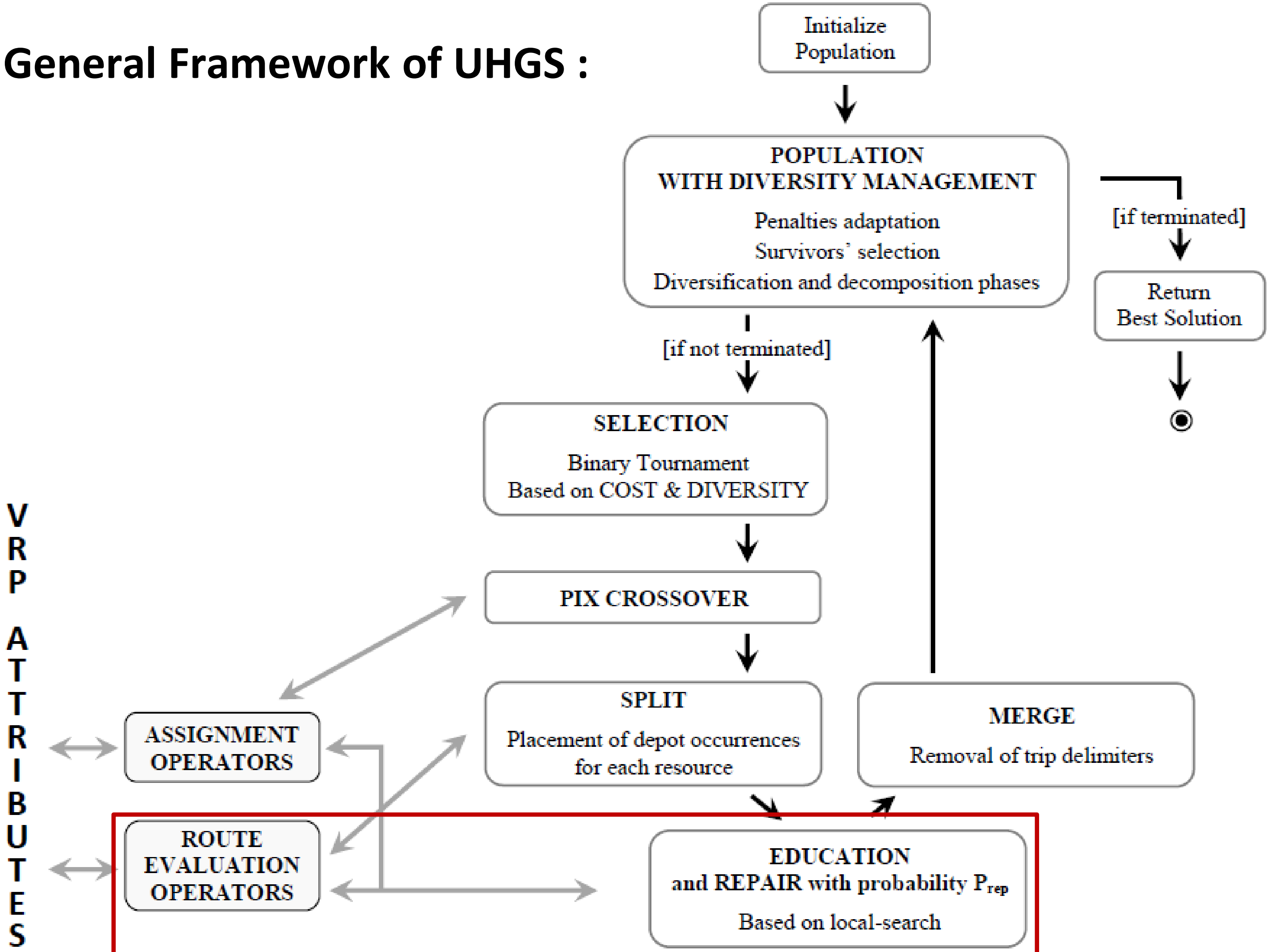
- Can serve as the basis to build any neighborhood-based unified solver based on VNS, Tabu, ILS for MAVRPs with EVAL attributes.
- Going one step further, designing a unified hybrid GA.

A Unified Hybrid Genetic Search (UHGS) for MAVRPs

- UHGS = Classic GA framework + 4 main ingredients (Vidal et al. 2010)
  - Management of penalized infeasible solutions in two subpopulations
  - High-performance local search-based *Education* procedure
  - Solution Representation *without trip delimiters*
  - **Diversity & Cost objective for individuals evaluations**

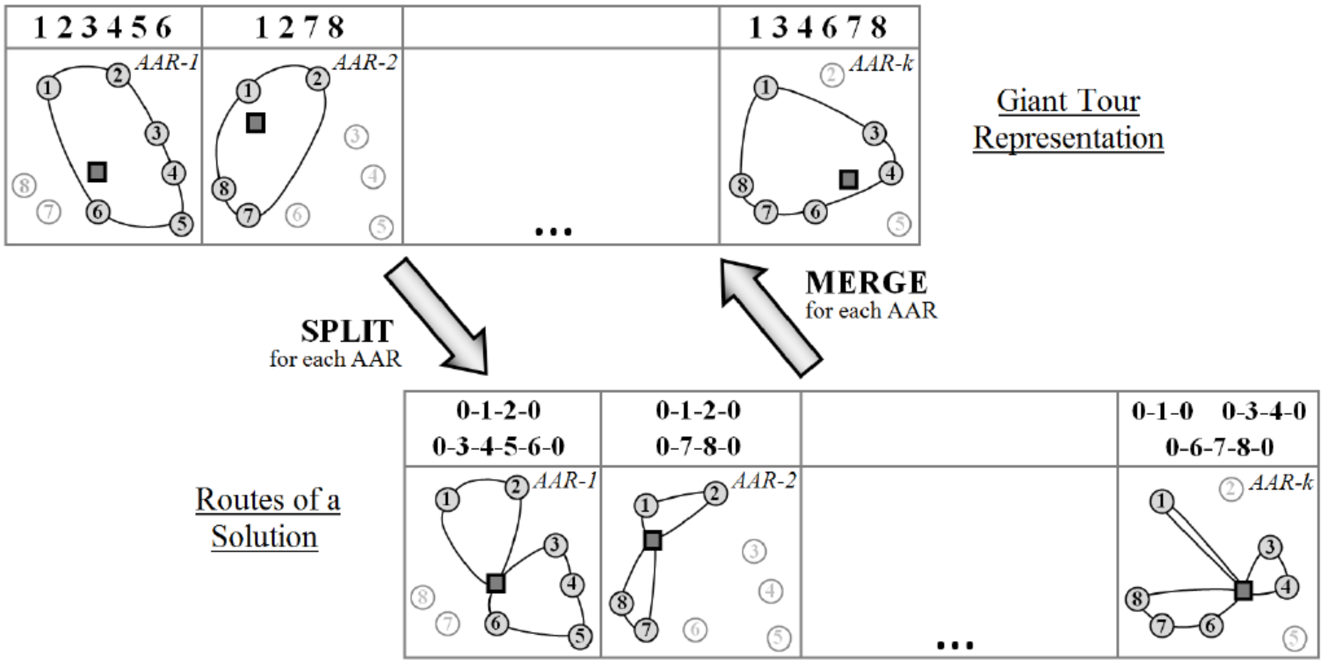
A Unified Hybrid Genetic Search (UHGS) for MAVRPs

- **General Framework of UHGS :**



Unified Solution Representation and Split

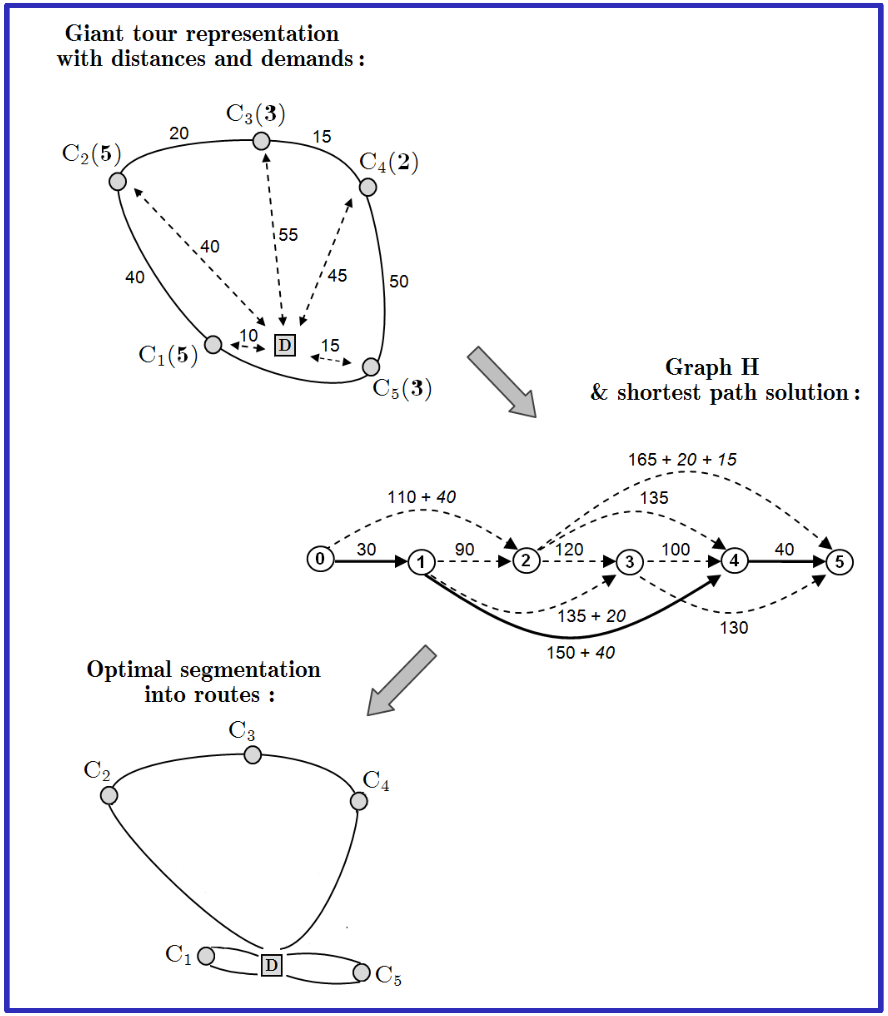
- Now dealing with MAVRPs with both ASSIGN and EVAL attributes: Assignment of customer services to some ASSIGN attributes resources (AARs) + separate optimization of routes for each AARs.
  - Solution representation is designed accordingly.
  - Furthermore, representation without trip delimiters for each AAR.



Unified Solution Representation and Split

- Solution representation as a giant-tour per AAR → requires a Split algorithm (Prins 2004) for optimal segmentation into routes.

- We propose a **unified Split algorithm**
  - As usual, the problem is solved as a m-shortest path
  - The route evaluation operators are used to build the auxiliary graph



Unified Solution Representation and Split

- Solution representation as a giant-tour per AAR → requires a Split algorithm (Prins 2004) for optimal segmentation into routes.

- **Unified Split algorithm.**

Algorithm 2 Generic Split

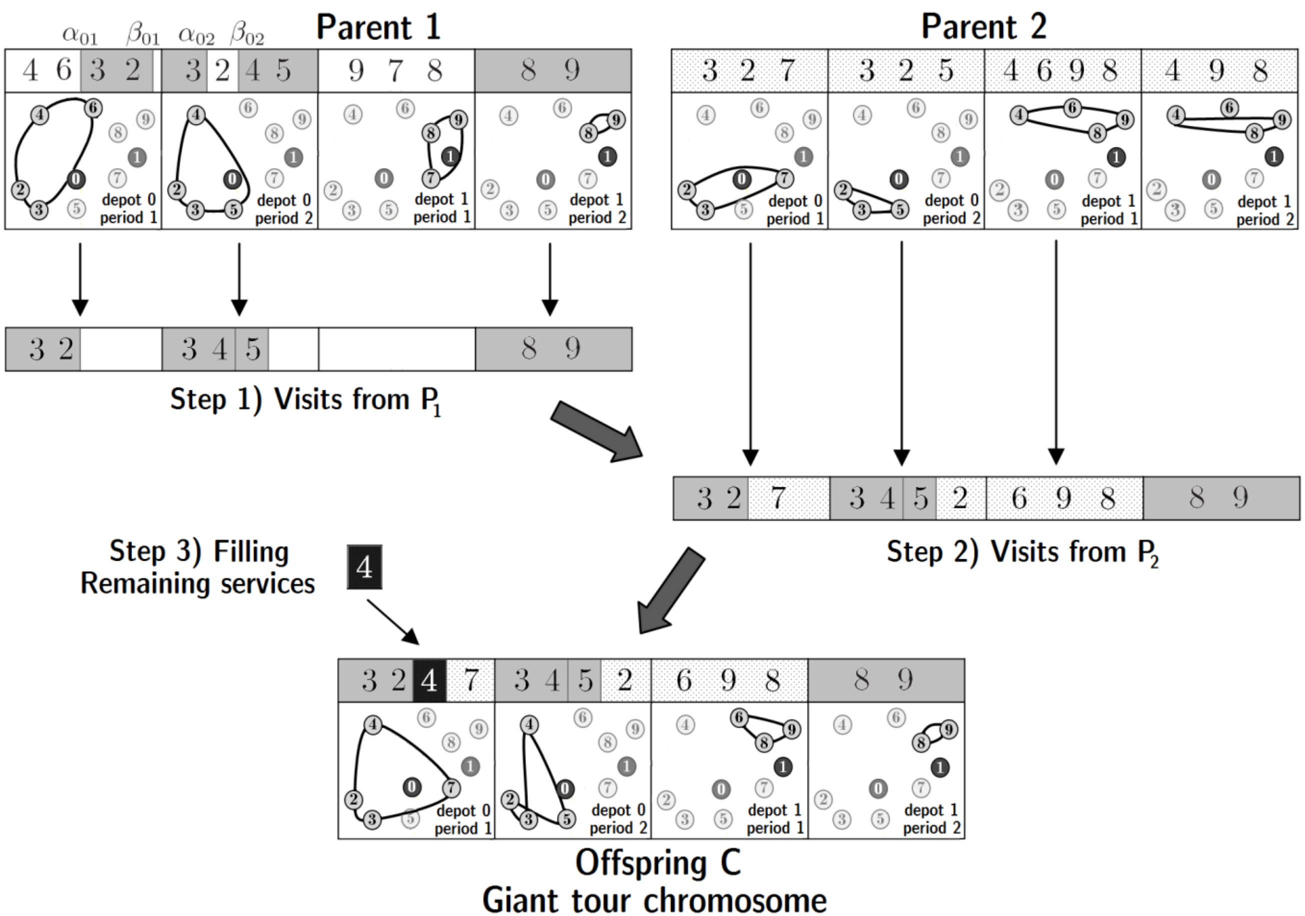
---

for each node $i \in \{0, \ldots, \nu\}$ **do**
    $SeqData(\sigma) = \textsc{Init}(\{v_0\})$ *//Initialize with depot vertex*
    for each node $j \in \{i, \ldots, \min(i + \bar{r}, \nu)\}$ **do**
        $\phi(a_{ij}) = \textsc{Eval2}(\sigma, \{v_0\})$ *//Evaluate the route*
        $SeqData(\sigma) = \textsc{Forw}(\sigma, \{\tau_j\})$ *//Append a new customer to the route end*
Solve the shortest path problem on $\mathcal{G}' = (\mathcal{V}, \mathcal{A})$ with cost $\phi(a_{ij})$ for each arc a_{ij}
Return the set of routes associated to the set of arcs of the shortest path

---

Unified Crossover Operator

- 4 phases **Assignment and Insertion Crossover (AIX),** to produce a single offspring C from two parents P1 and P2.



Unified Education Procedure

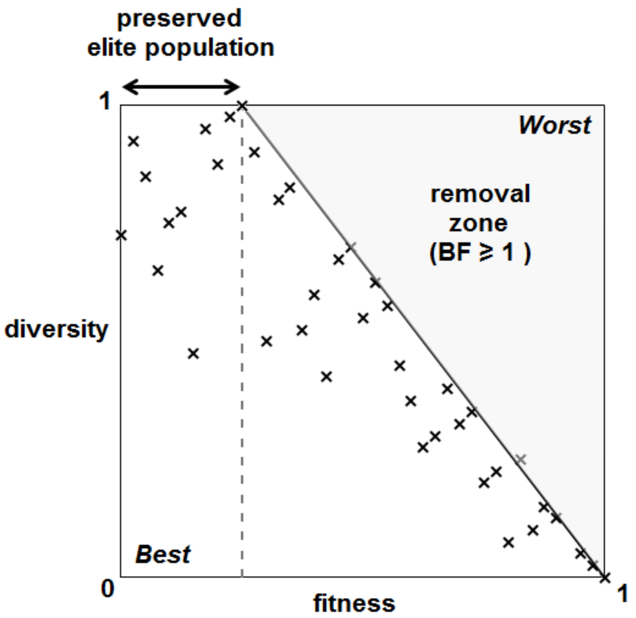
- Unified Local Search to perform route improvement (RI) on separate AAR.
  - Using CROSS, I-CROSS, Relocate, 2-Opt* and 2-Opt neighborhoods
  - Pruning procedures (granular search)
  - Combined with an assignment-improvement (AI) procedure to re-assign customer visits into different resources and routes : RI-AI-RI.

Population management and search guidance

- **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.



Comparison with problem-tailored state-of-the-art methods

- Extensive computational experiments on 26 structurally different VRP variants and 39 sets of benchmark instances.
  - A total of 1008 problem instances.

- Comparing UHGS with the best problem-tailored method for each benchmark and problem. 10 runs on each problem.

- In the following, we indicate for each method
  - % Gap to the BKS of an average run (out of 10 for UHGS).
  - % Gap to the BKS of a best run (out of 10 for UHGS).
  - Computational effort (total work time) for an average run
  - Type of processor used.

Comparison with problem-tailored state-of-the-art methods

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

Comparison with problem-tailored state-of-the-art methods

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

Comparison with problem-tailored state-of-the-art methods

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

Comparison with problem-tailored state-of-the-art methods

Variant	Bench.	n	Obj.	State-of-the-art methods				
				Author	Avg.%	Best%	T(min)	CPU
PVRPTW	CL01	[48,288]	C	PR08:	—	+1.75%	—	Opt 2.2G
				CM12:	+1.10%	+0.76%	64×11.3	64×Xe 3G
				UHGS*:	**+0.63%**	**+0.22%**	32.7	Xe 2.93G
MDVRPTW	CL01	[48,288]	C	PBDH08:	—	+1.37%	147	P-IV 3.6G
				CM12:	+0.36%	+0.15%	64×6.57	64×Xe 3G
				UHGS*:	**+0.19%**	**+0.03%**	6.49	Xe 2.93G
SDVRPTW	CL01	[48,288]	C	B10:	+2.23%	—	2.94	Qd 2.67G
				CM12:	+0.62%	+0.36%	64×5.60	64×Xe 3G
				UHGS*:	**+0.36%**	**+0.10%**	5.48	Xe 2.93G
VRPSTW (type 1, α=100)	SD88	100	F/TW/C	F10:	0%	—	9.69	P-M 1.6G
				UHGS:	**-3.05%**	**-4.42%**	18.62	Opt 2.2G
VRPSTW (type 1, α=1)	SD88	100	C+TW	KTDHS12:	+0.62%	+0.00%	10.0	Xe 2.67G
				UHGS:	**-0.13%**	**-0.18%**	5.82	Opt 2.2G
VRPSTW (type 2, α=100)	SD88	100	F/TW/C	FEL07:	0%	—	5.98	P-II 600M
				UHGS:	**-13.91%**	**-13.91%**	41.16	Opt 2.2G
VRPSTW (type 2, α=1)	SD88	100	C+TW	**UHGS:**	**+0.26%**	**0%**	29.96	Opt 2.2G
MDPVRPTW	New	[48,288]	C	**UHGS:**	**+0.77%**	**0%**	16.89	Opt 2.2G
VRTDSP (E.U. rules)	G09	100	F/C	PDDR10:	0%/0%	0%/0%	88	Opt 2.3G
				UHGS*:	**-0.56%/-0.54%**	**-0.85%/-0.70%**	228	Xe 2.93G

Comparison with problem-tailored state-of-the-art methods

- BKS has been found or improved on 954/1008 problems !
- Strictly improved on 550/1008 problems.
- All known optimal solutions have been retrieved !!
- Run time of a few minutes for average-size instances (n = 200-300)
- Standard deviation below 0.1%
- Outperforming the current best 179/180 problem-dedicated algorithms from the literature. New best method on 28/29 problems and 37/38 benchmarks !!!

Conclusions

- A component-based design & unified hybrid genetic search
  - Method structure designed in accordance with problem structure. Attribute-dependent tasks are identified and addressed by adaptive components.
  - UHGS proof-of-concept : with unified solution representation, Split procedure, genetic operators (Crossover) and population management methods.
  - **Major methodological breakthrough** : **UHGS redefines the state-of-the-art for 26 major VRP variants**, outperforming 179/180 current best problem-tailored methods.
  - **Major impact on current OR practice** : finally an efficient solver for rich VRP, state-of-the-art & ready-to-run.

- **Generality does not necessarily go against performance for the considered MAVRPs.**

Perspectives

- Perspectives :
  - Extend the range of problems (especially SEQ attributes, stochastic and multi-objective settings)
  - UHGS can be now viewed as a “laboratory” on which we can experiment new solution concepts not only on one problem, but on many at the same time
  - Integrate assignment, sequencing and route evaluations (often scheduling) within polynomially-enumerable large neighborhoods.
  - Problems with multiple levels, cross-docking, synchronization, and multiple modes.

Outline of the presentation

- **I) A general-purpose solver for multi-attribute vehicle routing problems**
  - Multi-attribute vehicle routing problems
  - An efficient and unified local search for MAVRPs
  - A Unified Hybrid Genetic Search (UHGS) for MAVRPs
  - Computational experiments

- **II) Timing problems and algorithms**
  - Several applications presenting similar *timing* issues
  - Classification and notation
  - Reductions
  - Timing Re-optimization

Several problems

- General effort dedicated to better address *rich vehicle routing problems* involving many side constraints and *attributes*.

- Observation : several VRP settings deserve their *richness* to the temporal features they involve : Time windows, time-dependent cost and travel times, flexible travel times, stochastic travel times, break scheduling…

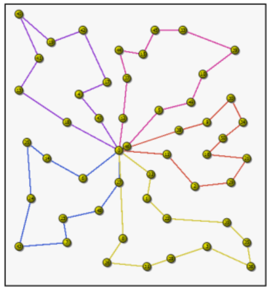
- The same questions are encountered in different domains: vehicle routing, scheduling, PERT, and isotone regression in statistics, among others.

- Leading us to a cross-domain analysis and classification of *timing problems and algorithms.*

Several problems

- Four problems originating from different domains

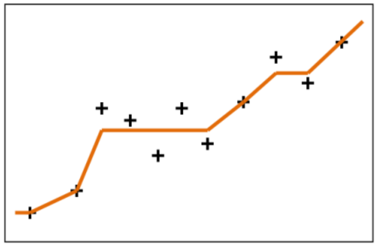
VRPTW



E/T scheduling

ship speed opt.

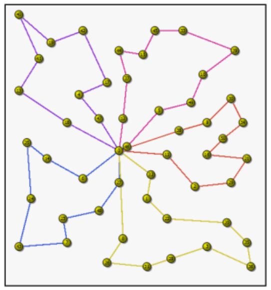
isotonic regression



Several problems

- Four problems originating from different domains:

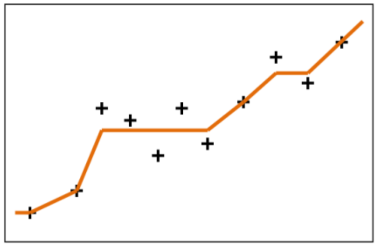
VRPTW



E/T scheduling

ship speed opt.

isotonic regression



<u>When visit sequence is fixed, optimizing on visit dates:</u>

$$\min_{(t_1,\ldots,t_n)\in\Re^{n+}} \sum_{i=1}^{n} \{\alpha(\bar{e}_i - t_i)^+ + \beta(t_i - \bar{l}_i)^+\}$$

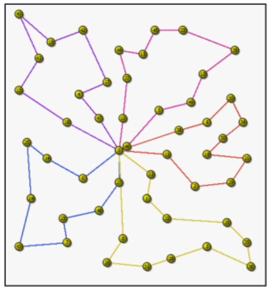
$$s.t. \quad t_i + p_i + d_{i,i+1} \leq t_{i+1} \qquad 1 \leq i < n$$

$$e_i \leq t_i \leq l_i \qquad 1 \leq i \leq n$$

Several problems

- Four problems originating from different domains:

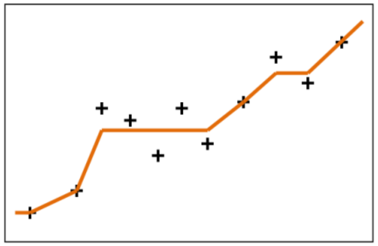
VRPTW



E/T scheduling

ship speed opt.

isotonic regression



When visit sequence is fixed, optimizing on task execution dates:

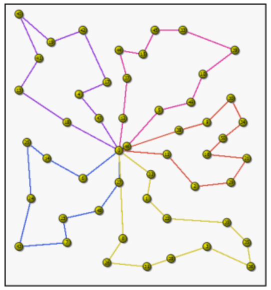
$$\min_{(t_1,\dots,t_n)\in\Re^{n+}} \sum_{i=1}^{n} \{\epsilon_i (d_i - t_i)^+ + \tau_i (t_i - d_i)^+\}$$

$$s.t. \quad t_i + p_i \leq t_{i+1} \qquad 1 \leq i < n$$

Several problems

- Four problems originating from different domains:

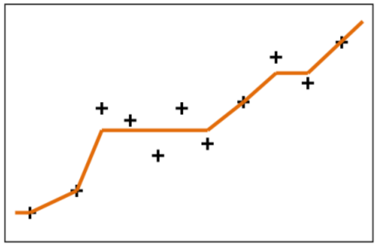
VRPTW



E/T scheduling

ship speed opt.

isotonic regression



When visit sequence is fixed, fuel consumption optimization:

$$\min_{(t_1,\dots,t_n)\in\Re^{n+}} \sum_{i=1}^{n} d_{i,i+1}\hat{c}\left(\frac{d_{i,i+1}}{t_{i+1}-t_i}\right)$$

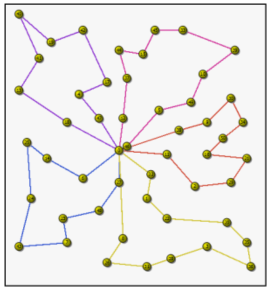
$$\text{s.t.} \quad t_i + p_i + d_{i,i+1}/v_{max} \le t_{i+1} \qquad 1 \le i < n$$

$$e_i \le t_i \le l_i \qquad 1 \le i \le n$$

Several problems

- Four problems originating from different domains:

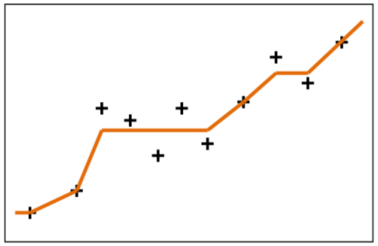
VRPTW



E/T scheduling

ship speed opt.

isotonic regression



$$\min_{\mathbf{t}=(t_1,\ldots,t_n)} \quad ||\mathbf{t} - \mathbf{N}||$$

$$t_i \leq t_{i+1} \qquad 1 \leq i < n$$

… with some characteristics in common

VRPTW

$$\min_{(t_1,\dots,t_n)\in\Re^{n+}} \sum_{i=1}^{n}\{\alpha(\bar{e}_i - t_i)^+ + \beta(t_i - \bar{l}_i)^+\}$$
$$s.t.\quad t_i + p_i + d_{i,i+1} \leq t_{i+1} \qquad 1 \leq i < n$$
$$e_i \leq t_i \leq l_i \qquad 1 \leq i \leq n$$

E/T scheduling

$$\min_{(t_1,\dots,t_n)\in\Re^{n+}} \sum_{i=1}^{n}\{\epsilon_i(d_i - t_i)^+ + \tau_i(t_i - d_i)^+\}$$
$$s.t.\quad t_i + p_i \leq t_{i+1} \qquad 1 \leq i < n$$

Isotonic regression

$$\min_{\mathbf{t}=(t_1,\dots,t_n)} \|\mathbf{t} - \mathbf{N}\|$$
$$t_i \leq t_{i+1} \qquad 1 \leq i < n$$

Ship speed opt.

$$\min_{(t_1,\dots,t_n)\in\Re^{n+}} \sum_{i=1}^{n} d_{i,i+1}\hat{c}\left(\frac{d_{i,i+1}}{t_{i+1} - t_i}\right)$$
$$\text{s.t.}\quad t_i + p_i + d_{i,i+1}/v_{max} \leq t_{i+1} \qquad 1 \leq i < n$$
$$e_i \leq t_i \leq l_i \qquad 1 \leq i \leq n$$

TIMING

$$\min_{\mathbf{t}=(t_1,\dots,t_n)\in\Re^{n+}} \sum_{F^x\in\mathcal{F}^{\text{OBJ}}} \alpha_x \sum_{1\leq y\leq m_x} f_y^x(\mathbf{t})$$
$$s.t.\quad t_i + p_i \leq t_{i+1} \qquad 1 \leq i < n$$
$$f_y^x(\mathbf{t}) \leq 0 \qquad F^x \in \mathcal{F}^{\text{CONS}},\ 1 \leq y \leq m_x$$

Timing problems

TIMING

$$\min_{\mathbf{t}=(t_1,\ldots,t_n)\in\Re^{n+}} \sum_{F^x\in\mathcal{F}^{\text{OBJ}}} \alpha_x \sum_{1\leq y\leq m_x} f_y^x(\mathbf{t})$$

$$s.t.\quad t_i + p_i \leq t_{i+1} \qquad 1 \leq i < n$$

$$f_y^x(\mathbf{t}) \leq 0 \qquad F^x \in \mathcal{F}^{\text{CONS}},\ 1 \leq y \leq m_x$$

- Timing problems seek to determine the execution dates $(t_1,\ldots,t_n)$ for a fixed sequence of activities.
- Totally ordered continuous variables
- Additional *features* F^x characterized by functions f_y^x for $1 \leq y \leq m_x$ that participate either in the objective or as constraints:
  - time windows, time-dependent proc. times, flexible travel times, time lags, no waiting, limited waiting, and so on…

Timing problems

TIMING

$$\min_{\mathbf{t}=(t_1,\dots,t_n)\in\Re^{n+}} \sum_{F^x \in \mathcal{F}^{\text{OBJ}}} \alpha_x \sum_{1\leq y \leq m_x} f_y^x(\mathbf{t})$$

$$s.t. \quad t_i + p_i \leq t_{i+1} \qquad 1 \leq i < n$$

$$f_y^x(\mathbf{t}) \leq 0 \qquad F^x \in \mathcal{F}^{\text{CONS}}, \ 1 \leq y \leq m_x$$

- Several names in the literature: *Scheduling*, *Timing*, *Projections onto Order Simplexes, Optimal service time problem* …
- Few dedicated studies, literature scattered among several research domains despite its relevance to many applications
- Thus motivating a dedicated review and analysis of timing algorithms to fill the gap.

Université de Montréal utt CIRRELT CONSEIL REGIONAL CHAMPAGNE ARDENNE

Timing features from the vehicle routing domain

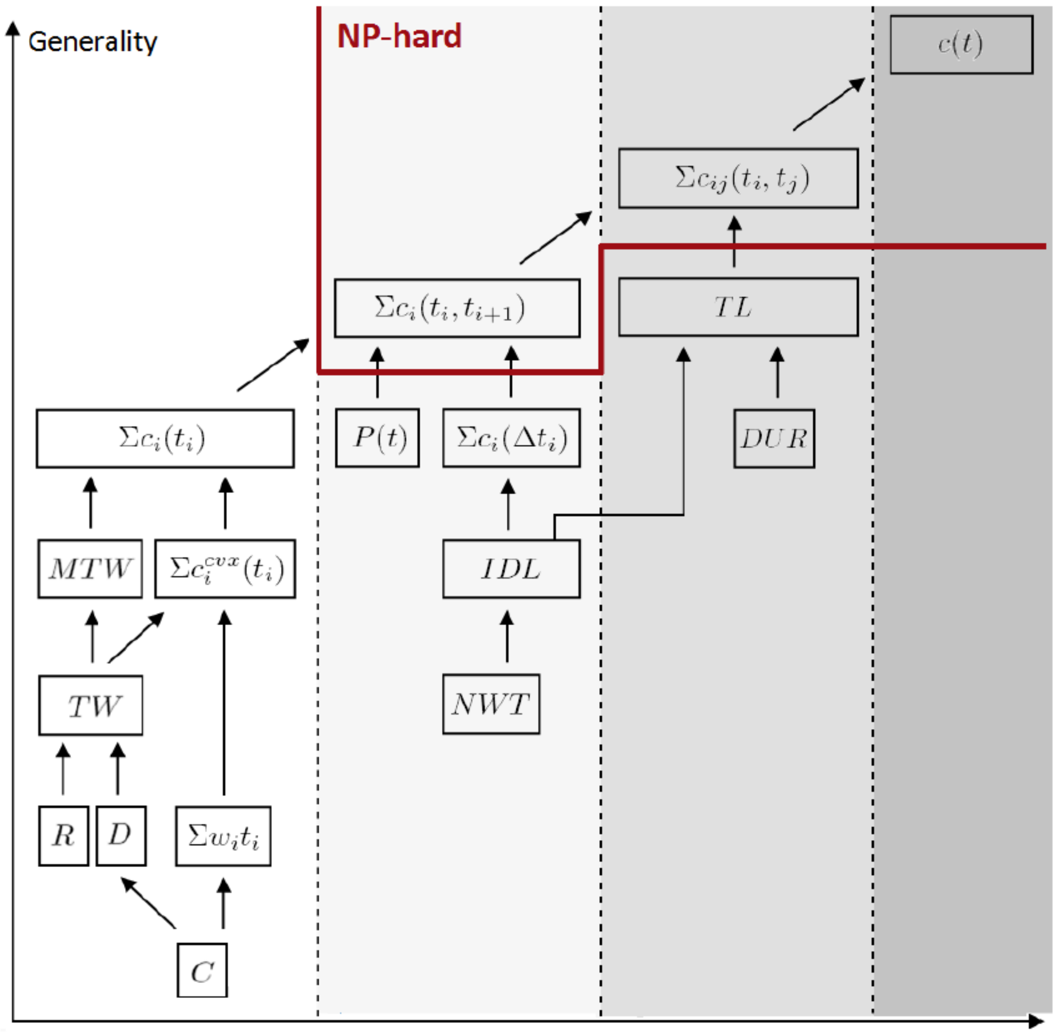
- **Rich vehicle routing problems** can involve various *timing features*

Symbol	Parameters	Char. functions	Most frequent roles
D	due dates d_i	$f_i(\mathbf{t}) = (t_i - d_i)^+$	Service deadlines constraints, tardiness
R	release dates r_i	$f_i(\mathbf{t}) = (r_i - t_i)^+$	Release-dates, earliness.
TW	time windows $TW_i = [e_i, l_i]$	$f_i(\mathbf{t}) = (t_i - l_i)^+ + (e_i - t_i)^+$	Time-window constraints, soft time windows.
MTW	multiple TW $MTW_i = \cup[e_{ik}, l_{ik}]$	$f_i(\mathbf{t}) = \min_k [(t_i - l_{ik})^+ + (e_{ik} - t_i)^+]$	Multiple time-window constraints
$\Sigma c_i(t_i)$	general $c_i(t)$	$f_i(\mathbf{t}) = c_i(t_i)$	Time-dependent service costs
$\Sigma c_i^{\text{cvx}}(t_i)$	convex $c_i^{\text{cvx}}(t_i)$	$f_i(\mathbf{t}) = c_i^{\text{cvx}}(t_i)$	Time-d. convex service costs
DUR	total dur. δ_{max}	$f(\mathbf{t}) = (t_n - \delta_{max} - t_1)^+$	Duration or overall idle time
NWT	no wait	$f_i(\mathbf{t}) = (t_{i+1} - p_i - t_i)^+$	No wait constraints
IDL	idle time ι_i	$f_i(\mathbf{t}) = (t_{i+1} - p_i - \iota_i - t_i)^+$	Limited idle time per stop, min idle time excess
$P(t)$	time-dependent proc. times $p_i(t_i)$	$f_i(\mathbf{t}) = (t_i + p_i(t_i) - t_{i+1})^+$	Time-dependent driving-times
TL	time-lags δ_{ij}	$f_i(\mathbf{t}) = (t_j - \delta_{ij} - t_i)^+$	Time-lag constraints
$\Sigma c_i(\Delta t_i)$	general $c_i(t)$	$f_i(\mathbf{t}) = c_i(t_{i+1} - t_i)$	Flexible travel times
$\Sigma c_{ij}(t_i, t_j)$	general $c_{ij}(t, t')$	$f_{ij}(\mathbf{t}) = c_i(t_i, t_j)$	Separable objectives or constraints by any pairs of variables ...

Timing features hierarchy

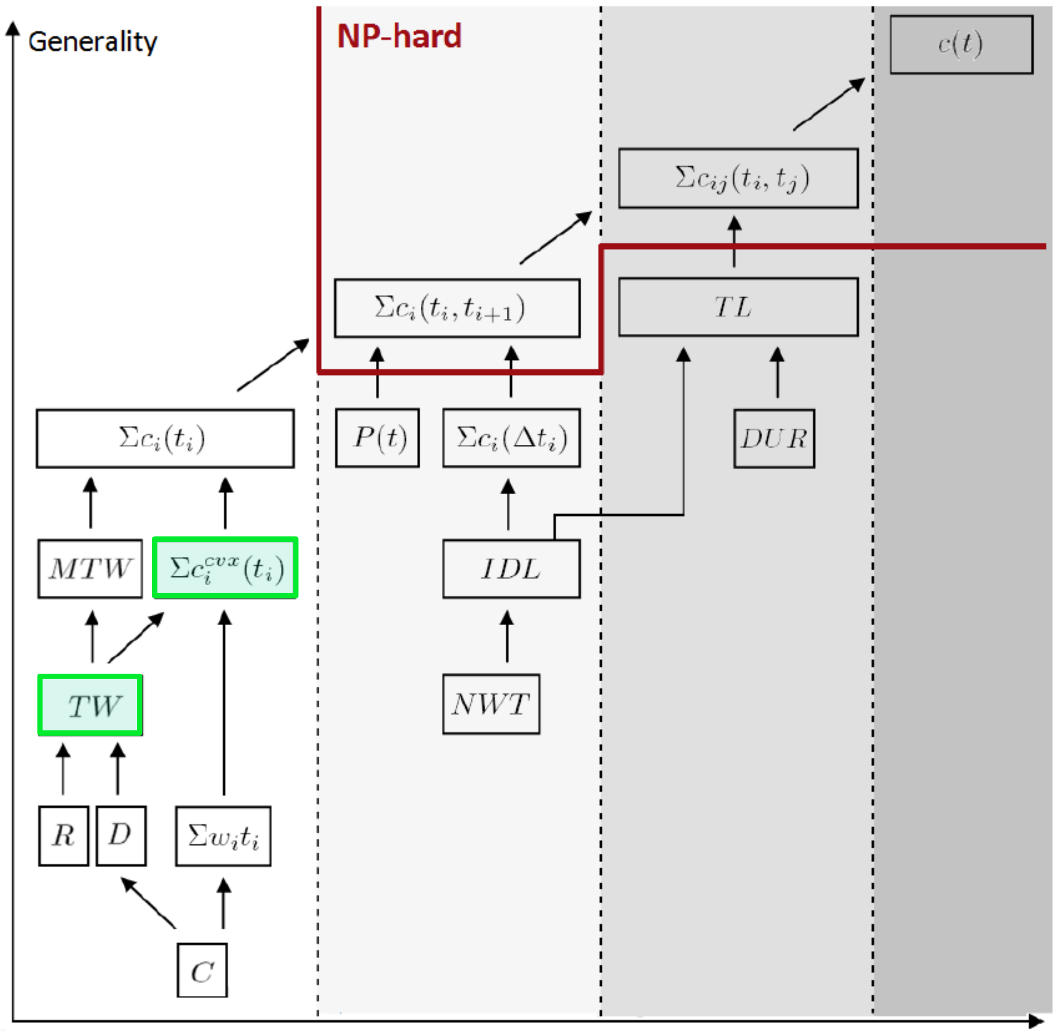
- These features can be classified and hierarchized (many-one linear reduction relationships between the associated timing problems)

- Features in the NP-hard area lead to NP-hard timing problems



Timing features hierarchy

- In this presentation, brief glimpse of the analysis.
- We examine a particular feature as illustrative example
- A similar study has been conducted on other features from this figure.



A feature example: soft time-windows

- Timing problem
  - with soft time-windows (penalized early and late arrival)
  - and generally with any convex separable cost

$$\min_{(t_1,\ldots,t_n)\in\Re^{n+}} \sum_{i=1}^{n} \{\alpha(\bar{e}_i - t_i)^+ + \beta(t_i - \bar{l}_i)^+\}$$
$$s.t. \quad t_i + p_i \leq t_{i+1} \qquad 1 \leq i < n$$

$$\min_{(t_1,\ldots,t_n)\in\Re^{n}} \sum_{i=1}^{n} c_i^{\text{CVX}}(t_i)$$
$$s.t. \quad t_i + p_i \leq t_{i+1}$$

- We inventoried more than 30 algorithms from various domains (routing, scheduling, PERT, statistics…) that address these models.

- The solution block representation / active set framework (Chakravarti 1989, Best & Chakravarti 1990, Best et al. 2000, Ahuja & Orlin 2001) can be used to characterize these methods. But we need to generalize the optimality conditions to the non-smooth case.

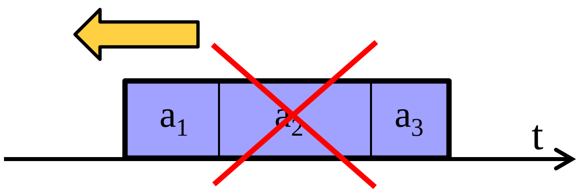
A feature example: soft time-windows

- A block B is defined as a subsequence of activities $(a_{B(1)},\ldots,a_{B(|B|)})$ processed consecutively (such that $t_i + p_i = t_{i+1}$)

- <u>Theorem</u>: Let costs $c_i(t_i)$ be proper convex, eventually non-smooth, functions. **A solution $(t^*_1,\ldots,t^*_n)$ of the timing problem with convex separable costs is optimal if and only if** it can be assimilated to a succession of activity blocks $(B_1,\ldots,B_m)$ such that:

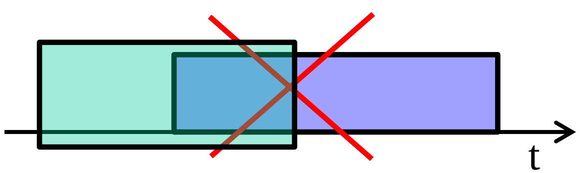
  1) **Blocks are optimally placed**: for each block B_i,

  $$t^*_{Bi(1)} \in \mathit{argmin}\ C_{Bi}(t)$$

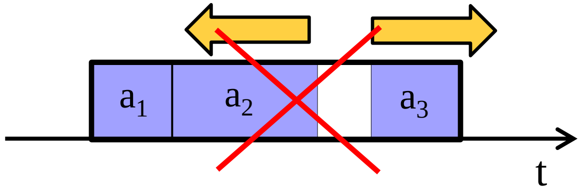
  2) **Blocks are spaced**: for each pair of blocks (B_i,B_{i+1}),

  $$t^*_{Bi(1)} + \Sigma p_{Bi(j)} < t^*_{Bi+1(1)}$$

  3) **Blocks are consistent**: for each block B_i and prefix block B_i^k,

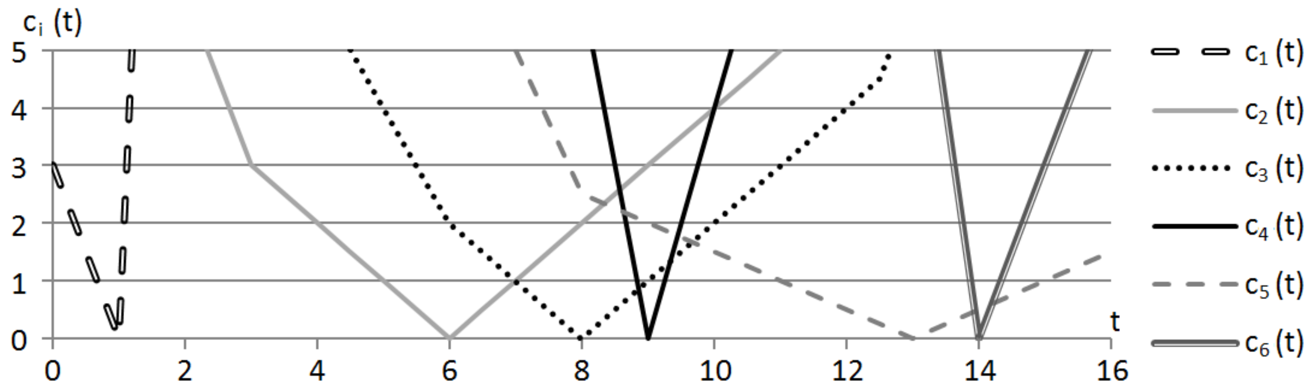
  $$\mathit{max\ argmin}\ C_{Bi}^k(t) \geq t^*_{Bi(1)}$$

A feature example: soft time-windows

- Three main families of algorithms can be identified:
  - Primal feasible, that respect *spacing condition 2*
  - Dual feasible, that respect *consistency condition 3*
  - Dynamic programming

- To illustrate, consider a small problem with 6 activities

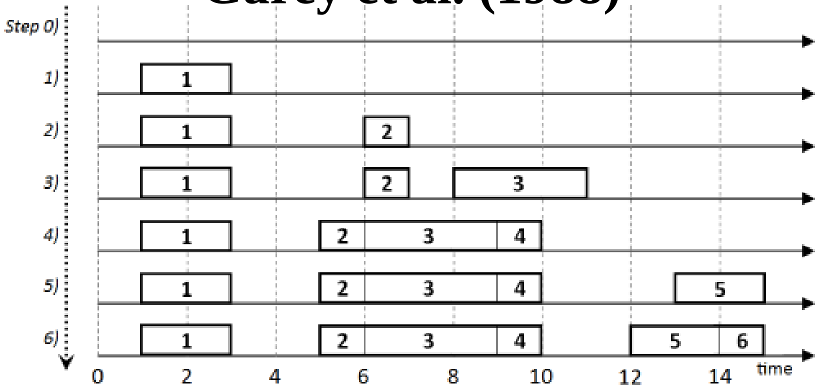


Act. i	p_i
1	2
2	1
3	3
4	1
5	2
6	1

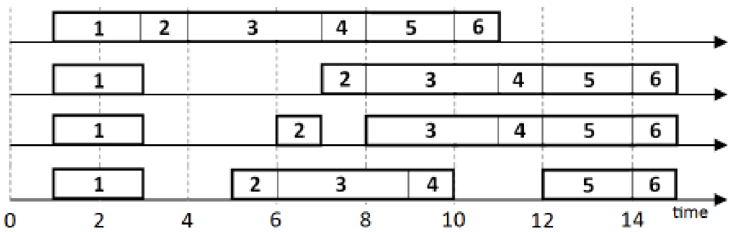
A feature example: soft time-windows

- **Primal feasible method, respecting the spacing condition.**
  - Brunk (1955) : Minimum Lower Set Algorithm : *O (n^2) unimodal minimizations*
  - Extended by Garey et al. (1988) and Best & Chakravarti (1990) :
    O(n) unimodal function minimizations in the general convex case.
    O(n log n) elementary operations in the case of (E/T) scheduling

Garey et al. (1988)



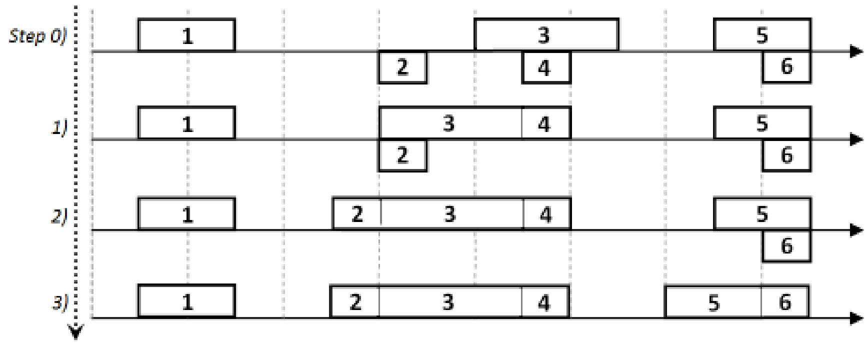
Best & Chakravarti (1990)



  - Other related methods from (E/T) scheduling: Lee and Choi (1995), Davis and Kanet (1993), Wan and Yen (2002), Pan and Shi (2005) in *O(n log n)*
  - For more general cases, Hendel and Sourd (2007) for PL functions, and Chrétienne and Sourd (2003) in the context of PERT with convex costs.

A feature example: soft time-windows

- **Dual feasible method, respecting the consistency condition.**
  - Ayer et al. (1955) : Pool Adjacent Violator Algorithm (PAV).



  - Best et al. (2000) and Ahuja and Orlin (2001) : Extension to the general convex case -> *O(n) unimodal function minimizations*
  - Pardalos (1995) : *O(n log² n) for Isotone Regression with* $|| \; ||_1$ (= E/T scheduling with equal penalties for earliness and tardiness)
  - Grotzinger and Witzgall (1984) and Pardalos and Xue (1999) -> *O(n) elementary operations for the quadratic case.*
  - Dumas et al. (1990) : another application of this principle for the VRP with convex service costs -> *O(n) unimodal function minimizations*

A feature example: soft time-windows

- **Dynamic programming-based methods** (Yano and Kim 1991, Sourd 2005, Ibaraki et al. 2005, 2008, Hendel and Sourd 2007, Hashimoto et al. 2006, 2008)

- Forward dynamic programming

$$F_i(t) = \min_{0 \leq x \leq t} \{c_i(x) + F_{i-1}(x - p_{i-1})\}$$

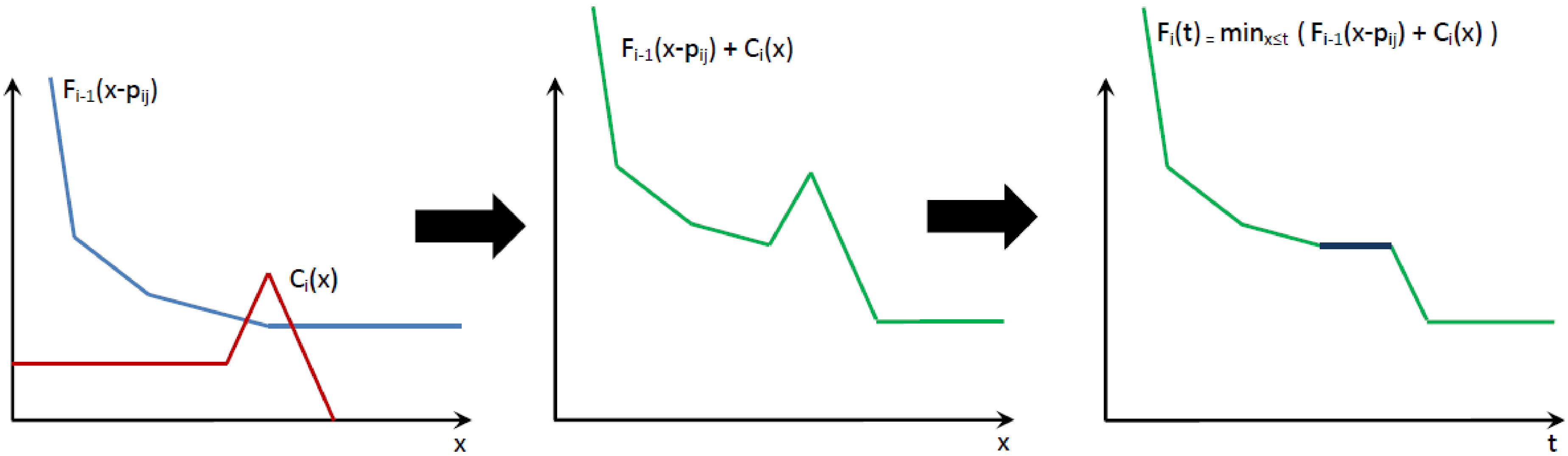
- Backward dynamic programming

$$B_i(t) = \min_{x \geq t} \{c_i(x) + B_{i+1}(x + p_i)\}$$

A feature example: soft time-windows

- Forward dynamic programming

$$F_i(t) = \min_{0 \leq x \leq t} \{c_i(x) + F_{i-1}(x - p_{i-1})\}$$



Timing problems

- Hence, many different methods for this particular feature example. The literature on timing problems is rich, but scattered. All in all, 26 different methods from different domains were classified as variations of 3 main algorithmic ideas.

Timing re-optimization

- Furthermore, when used within LS, solving all timing problems *from scratch* is generally not efficient

- The general goal when exploring neighborhoods is to solve N successive timing problems with different activity permutations σ^k.

$$\min_{\mathbf{t}=(t_1,\dots,t_n)\in\Re^{n+}} \sum_{F^x\in\mathcal{F}^{\mathrm{OBJ}}} \alpha_x \sum_{1\le y\le m_x} f_y^x(\mathbf{t})$$

$$s.t.\quad t_{\sigma^k(i)} + p_{\sigma^k(i)\sigma^k(i+1)} \le t_{\sigma^k(i+1)}$$

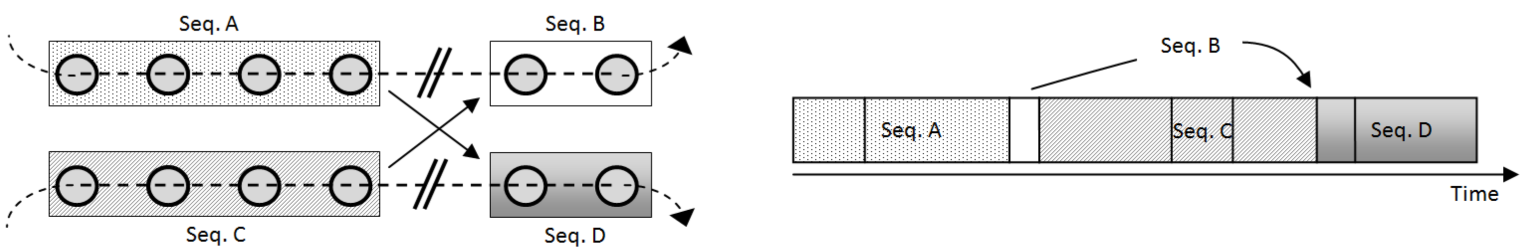
$$f_y^x(\mathbf{t}) \le 0$$

Timing re-optimization

- In classical VRP neighborhoods, the neighborhood size is often rather large: |N| = $\Omega(n^2)$, and permutations are very particular.
  - They have a bounded number (often <= 4) of *breakpoints: integers x such that* $\sigma(x)+1 \neq \sigma(x+1)$,

12**3** 67**8** 45

  - The resulting sequences of activities can be assimilated to recombinations of a bounded number of subsequences.



  - Such that invariants on subsequences can be exploited through the search (Savelsbergh 1985, 1992, Kindervater 1998).

Timing re-optimization

- Management of information of subsequences, efficient *timing re-optimization* by means of a subset of 4 procedures, used within local searches:
  - Initialization of suitable re-optimization data for a single activity
  - Forward (F) or backward (B) computation of data on larger subsequences
  - Evaluation of a concatenation of two (C2) or more (C3+) subsequences

Algorithm 1 Re-optimization

Build re-optimization data on subsequences of the *incumbent timing problem* $\mathcal{T}$, using *initialize*, and *forward extension* or *backward extension*.
For each timing subproblem $\mathcal{T}^k$, $k \in \{1, \ldots, N\}$;
    Determine the breakpoints involved in the permutation function σ^k;
    Evaluate the optimal cost of $\mathcal{T}^k$, as the concatenation of $b(\sigma) + 1$ activity subsequences from $\mathcal{T}$ (see Equation 39).

Timing re-optimization

- Example of soft time-windows: Forward and backward extension to compute data on subsequences, and *evaluate concatenation* of 2 sequences (Ibaraki et al. 2005, 2008):

$$Z^*(A_1 \oplus A_2) = \min_{t \geq 0}\{F(A_1)(t) + B(A_2)(t + p_{A_1(|A_1|)A_2(1)})\}$$

  - In the convex case, the concatenation of 3+ sequences is also addressed efficiently.
  - O(log ϕ) for convex piecewise functions with a total of ϕ pieces.
  - O(log n) move evaluations for soft TW

Conclusions on timing problems

- For other features: Surveying the literature, we classified many re-optimization based methodologies from various domains, and for a large variety of attributes. (Savelsbergh 1985,1992, Kindervater and Savelsbergh 1997, Campbell and Savelsbergh 2004, Ergun and Orlin 2006, Irnich 2008, Hashimoto et al. 2006,2008, Kedad-Sidhoum and Sourd 2010)…

- We could identify a set of state-of-the-art timing methods, which are the key to solve many rich VRP settings:

Conclusions on timing problems

Problem	From Scratch		Re-opt. by concat.	F/B	C2	C3+	Sd	Assumptions
$\{W\|ø\}$	Min idle time	$O(n)$	—	$O(1)$		$O(1)$	✓	
$\{ø\|TW\}$	Min idle time	$O(n)$	Savelsbergh (1985) & Kind. and Sav. (1997)	$O(1)$		$O(1)$	✓	
$\{D\|ø\}$	Min idle time	$O(n)$	Ergun and Orlin (2006)	$O(\log n)$	$O(1)^*$	—		penalty coefficient depending upon act.
$\{D, R(d_i = r_i)\|NWT\}$	Min idle time	$O(n)$	Kedad-Sidhoum and Sourd (2010)	$O(\log n)$	$O(1)^*$	—		penalty coefficient depending upon act.
$\{D, R(d_i = r_i)\|ø\}$	Garey et al. (1988) & Ahuja and Orlin (2001)	$O(n \log n)$	Ibaraki et al. (2008)	$O(\log n)$		$O(\log n)$	✓	
$\{D\|R\}$	Min idle time	$O(n)$	Ibaraki et al. (2008)	$O(\log n)$		$O(\log n)$	✓	
$\{\Sigma c_i^{\text{cvx}}(t_i)\|ø\}$	Ibaraki et al. (2008)	$O(n \log \varphi_c)$	Ibaraki et al. (2008)	$O(\log \varphi_c)$		$O(\log \varphi_c)$	✓	cost f. ≥ 0, p.l. & l.s.c
$\{\Sigma c_i(t_i)\|ø\}$	Ibaraki et al. (2005)	$O(n\varphi_c)$	Ibaraki et al. (2005)	$O(\varphi_c)$		$O(\varphi_c)$	✓	cost f. ≥ 0, p.l. & l.s.c
$\{ø\|MTW\}$	Min idle time	$O(n + \varphi_{\text{MTW}})$	Ibaraki et al. (2005)	$O(\log \varphi_{\text{MTW}})$		—	✓	
$\{DUR\|TW\}$, $\{ø\|DUR, TW\}$	Malcolm et al. (1959)	$O(n)$	Savelsbergh (1992) & Kind. and Sav. (1997)	$O(1)$		$O(1)$	✓	
$\{DUR\|MTW\}$, $\{ø\|DUR, MTW\}$	Tricoire et al. (2010)	$O(n\varphi_{MTW})$	Hashimoto et al. (2006)	$O(\varphi_{\text{MTW}})$		—	✓	
$\{ø\|IDL, TW\}$	Hunsaker and S. (2002)	$O(n)$	—	—	—	—		
$\{\Sigma c_i^{\text{cvx}}(\Delta t_i), \Sigma c_i(t_i)\|ø\}$	Sourd (2005) & Hashimoto et al. (2006)	$O(n(\varphi_c + \widehat{\varphi_c} \times \varphi_c'))$	Sourd (2005) & Hashimoto et al. (2006)	$O(\varphi_c + \widehat{\varphi_c} \times \varphi_c')$		—	✓	cost f. ≥ 0, p.l. & l.s.c
$\{D\|R, P(t)\}$	Min idle time	$O(n)$	—	—	—	—		FIFO assumption
$\{ø\|TW, P(t)\}$	Donati et al. (2008)	$O(n)$	Donati et al. (2008)	$O(1)$		—	✓	FIFO assumption
$\{\Sigma c_i(t_i)\|P(t)\}$	Hashimoto et al. (2008)	$O(n(\varphi_c + \varphi_p))$	Hashimoto et al. (2008)	$O(\varphi_c + \varphi_p)$		—	✓	cost f. ≥ 0, p.l. & l.s.c & HYI assumption
$\{ø\|TL, TW\}$	Hurink and Keuchel (2001)	$O(n^3)$	—	—	—	—		
$\{ø\|TL, TW\}$	Haugland and Ho (2010)	$O(n \log n)$	—	—	—	—		$O(n)$ TL constraints
$\{DUR > D > TL\|R\}$	Cordeau and Laporte (2003)	$O(n^2)$	—	—	—	—		$O(n)$ TL constraints & LIFO assumption
$\{\Sigma c_{ij}^{\text{cvx}}(t_j - t_i), \Sigma c_i^{\text{cvx}}(t_i)\|ø\}$	Ahuja et al. (2003)	$O(n^3 \log n \log(nU))$	—	—	—	—		U is an upper bound of execution dates

Conclusions on timing problems

- Large analysis of a rich body of problems with time characteristics and totally ordered variables. Cross-domain synthesis, considering methods from various fields such as vehicle routing, scheduling, PERT, and isotonic regression. Identification of main resolution principles

- For several "rich" combinatorial optimization settings, the timing sub-problems represent the core of "richness" and deserve particular attention.

- Furthermore, timing sub-problems frequently arise in the context of local search, and thus we analyzed both stand-alone resolution and efficient solving of series of problems.

Perspectives

- Proof of concept : timing procedures have been integrated in a recent Unified Hybrid Genetic Search, yielding state-of-the-art results on 26 variants of vehicle routing problems
- Several features and feature combinations were identified in this work, for which new timing algorithms (including re-optimization procedures) should be sought.
- Generalization to other cumulative resources, multi-objective or stochastic settings.
- Further studies on complexity lower bounds.

Thank you for your attention !

- For further reading, and follow-up works:

- Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., & Rei, W. (2012*). A* Hybrid Genetic Algorithm for Multi-Depot and Periodic Vehicle Routing Problems. *Operations Research*, 60(3), 611–624.
- Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, 40(1), 475–489.
- Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2012). A Unifying View on Timing Problems and Algorithms. *Submitted to C&OR. Tech Rep CIRRELT-2011-43.*
- Vidal T., Crainic T.G., Gendreau M., Prins C. Heuristics for Multi-Attribute Vehicle Routing Problems: A Survey and Synthesis (2012). *Submitted to EJOR. Revised. Tech Rep CIRRELT-2012-05.*
- Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2012). A Unified Solution Framework for Multi-Attribute Vehicle Routing Problems. *Submitted to Operations Research. Tech Rep CIRRELT-2012-23.*
- Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2012). Implicit Depot Assignments and Rotations in Vehicle Routing Heuristics. *Submitted to EJOR. Tech Rep CIRRELT-2012-60.*
- Goel, A., & Vidal, T. (2012). Hours of service regulations in road freight transport : an optimization-based international assessment. *Submitted to Trans. Sci. Revised. Tech Rep CIRRELT-2012-08.*

- **These papers + some others can be found at** http://w1.cirrelt.ca/~vidalt/

Université de Montréal | utt | CIRRELT | CONSEIL REGIONAL CHAMPAGNE ARDENNE | NSERC Industrial Research Chair in Logistics Management / Chaire de recherche industrielle du CRSNG en management logistique

Bibliography

- Ahuja, R. K., & Orlin, J. B. (2001). A Fast Scaling Algorithm for Minimizing Separable Convex Functions Subject to Chain Constraints. Operations Research, 49(5).
- Ayer, M., Brunk, H. D., Ewing, G. M., Reid, W. T., & Silverman, E. (1955). An empirical distribution function for sampling with incomplete information. The Annals of Mathematical Statistics, 26(4), 641–647. JSTOR.
- Best, M. J., & Chakravarti, N. (1990). Active set algorithms for isotonic regression, a unifying framework. Mathematical Programming, 47(1-3), 425-439.
- Best, M. J., Chakravarti, N., & Ubhaya, V. A. (2000). Minimizing separable convex functions subject to simple chain constraints. SIAM Journal on Optimization, 10(3), 658-672.
- Brunk, H. D. (1955). Maximum Likelihood Estimates of Monotone Parameters. The Annals of Mathematical Statistics, 26(4), 607-616.
- Campbell, A. M., & Savelsbergh, M. (2004). Efficient insertion heuristics for vehicle routing and scheduling problems. Transportation science, 38(3), 369–378.
- Chakravarti, N. (1989). Isotonic Median Regression: A Linear Programming Approach. Mathematics of Operations Research, 14(2), 303-308.
- Chrétienne, P., & Sourd, F. (2003). PERT scheduling with convex cost functions. Theoretical Computer Science, 292(1), 145-164.

Bibliography

- Dumas, Y., Soumis, F., & Desrosiers, J. (1990). Optimizing the Schedule for a Fixed Vehicle Path with Convex Inconvenience Costs. Transportation Science, 24(2), 145-152.
- Ergun, O., & Orlin, J. (2006). Fast neighborhood search for the single machine total weighted tardiness problem. Operations Research Letters, 34(1), 41-45.
- Garey, M. R., Tarjan, R. E., & Wilfong, G. T. (1988). One-processor scheduling with symmetric earliness and tardiness penalties. Mathematics of Operations Research, 13(2), 330–348.
- Grotzinger, S. J., & Witzgall, C. (1984). Projections onto Order Simplexes. Applied Mathematics and Optimization, 270(12), 247-270.
- Hashimoto, H., Ibaraki, T., Imahori, S., & Yagiura, M. (2006). The vehicle routing problem with flexible time windows and traveling times. Discrete Applied Mathematics, 154(16), 2271-2290.
- Hashimoto, H., Yagiura, M., & Ibaraki, T. (2008). An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. Discrete Optimization, 5(2), 434-456.
- Hendel, Y., & Sourd, F. (2007). An improved earliness-tardiness timing algorithm. Computers & Operations Research, 34(10), 2931-2938.

Bibliography

- Irnich, S. (2008). Resource extension functions: properties, inversion, and generalization to segments. OR Spectrum, 30, 113-148.
- Kedad-Sidhoum, S., & Sourd, F. (2010). Fast neighborhood search for the single machine earliness-tardiness scheduling problem. Computers & Operations Research, 37(8), 1464–1471.
- Kindervater, G. A. P., & Savelsbergh, M. W. P. (1997). Vehicle routing: Handling edge exchanges. Local Search in Combinatorial Optimization, 337–360.
- Pardalos, P. (1995). Efficient computation of an isotonic median regression. Applied Mathematics Letters, 8(2), 67-70.
- Savelsbergh, M. W. P. (1985). Local Search in Routing Problems with Time Windows. Annals of Operations Research, 4(1), 285-305.
- Savelsbergh, M. W. P. (1992). The Vehicle Routing Problem with Time Windows: Minimizing Route Duration. ORSA Journal on Computing, 4(2), 146-154.
- Sourd, F. (2005). Optimal timing of a sequence of tasks with general completion costs. European Journal of Operational Research, 165(1), 82-96.
- Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., & Rei, W. (2011). A Hybrid Genetic Algorithm for Multi-Depot and Periodic Vehicle Routing Problems. Operations Research, Forthcoming.

Bibliography

- Yano, C. A., & Kim, Y.-D. (1991). Algorithms for a class of single-machine weighted tardiness and earliness problems. European Journal of Operational Research, 52(2), 167–178. Elsevier.

Empirical studies on diversity management methods (1/2)

- Sensitivity analysis on diversity management methods:
  - **HGA** : No diversity management method
  - **HGA-DR** : Dispersal rule on objective space
  - **HGA-PM** : Dispersal rule on solution space
  - **HGSADC** : The proposed approach

Benchmark		HGA	HGA-DR	HGA-PM	HGSADC
PVRP	T	6.86 min	7.01 min	7.66 min	8.17 min
	%	+0.64%	+0.49%	+0.39%	+0.13%
MDVRP	T	7.93 min	7.58 min	9.03 min	8.56 min
	%	+1.04%	+0.87%	+0.25%	-0.04%
MDPVRP	T	25.32 min	26.68 min	28.33 min	40.15 min
	%	+4.80%	+4.07%	+3.60%	+0.44%

Empirical studies on diversity management methods (2/2)

- Behavior of HGSADC during a random run:
  - Higher entropy (average distance between two individuals)
  - Better final solution
  - Diversity can increase during run time