Approaching the Pareto-Frontier of the Multi-Objective Dial-a-Ride Problem

Sophie N. Parragh[1], **Karl F. Doerner**[1]
Xavier Gandibleux[2], Richard F. Hartl[1]

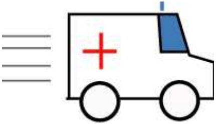
[1] Department of Business Administration, University of Vienna, Austria
[2] Laboratoire d'Informatique de Nantes Atlantique, Université de Nantes, France

May 16, 2007

universität wien

Motivation

- Increasing demand for decision support in health care organizations, especially in patient transportation
- Usually only one focus: cost minimization
- BUT: There exists a tradeoff between cost minimization and passenger's inconvenience
- Almost all previous approaches handle the multiobjective nature of the problem by introducing a weighted sum objective function.
- The major disadvantage of this approach consists in the loss of information wrt other tradeoff solutions.



Outline

A Bi-Objective Optimization Problem

$$\text{“minimize”}_{x \in X} \quad \mathbf{f}(x) = (f_1(x), f_2(x)) \tag{1}$$

where $X \subset \mathbb{R}^n$ is a set of feasible solutions or decision space and $f_i : \mathbb{R}^n \to \mathbb{R}^2$ $(i = 1, 2)$ are the $k \geq 2$ vector-valued objective functions that need to be minimized. $\mathbb{R}^2$ is called the objective space.

Remarks:

1. No single solution simultaneously accomplishes the two objectives
2. Consider the Pareto optimal solutions or the set of efficient solutions

A bi-objective optimization problem

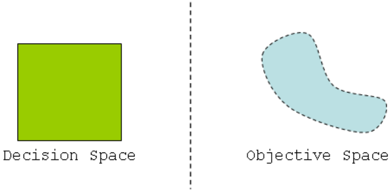


Figure: Bi-objective optimization

A bi-objective optimization problem

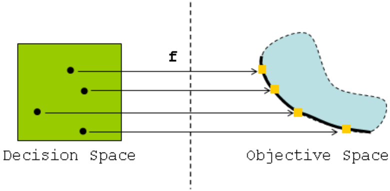


Figure: Bi-objective optimization

A Bi-Objective Optimization Problem

Definition

An element $x \in X$ is an efficient solution or Pareto optimal solution if there does not exist any $x' \in X$ such that $f_i(x') \leq f_i(x)$ for all i and $f_j(x') < f_j(x)$ for some j. The set of all efficient solutions X_E is called an efficient set or Pareto optimal set.

Definition

If x^* is Pareto optimal, then $z^* = \mathbf{f}(x^*)$ is called nondominated point or efficient point. The set of all nondominated points is referred to as nondominated frontier or Pareto front.

Definition

The process of finding the set of efficient solutions is called bi-objective optimization or Pareto optimization.

Pareto Dominance Relations

Pareto Dominance Relations (1)

1. z^1 strictly dominates z^2 if z^1 is better than z^2 in all objectives i.e., $z^1 \prec\prec z^2 \Leftrightarrow z_i^1 < z_i^2 \ \forall i$
2. z^1 dominates z^2 if z^1 is not worse than than z^2 in all objectives and better in at least one objective i.e., $z^1 \prec z^2 \Leftrightarrow z_i^1 \leq z_i^2 \ \forall i$ and $z_j^1 < z_j^2$ for some j
3. z^1 weakly dominates z^2 if z^1 is not worse than than z^2 in all objectives i.e., $z^1 \preceq z^2 \Leftrightarrow z_i^1 \leq z_i^2 \ \forall i$
4. z^1 is incomparable with z^2 if neither z^1 weakly dominates z^2 nor z^2 weakly dominates z^1 i.e., $z^1 \parallel z^2 \Leftrightarrow z^1 \neg \preceq z^2$ and $z^2 \neg \preceq z^1 \ \forall i$

Pareto Dominance Relations

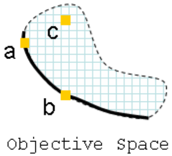


Figure: Examples of dominance relations on objective vectors: $a \prec\prec c$, $a \prec c$, $a \preceq c$, $a \parallel b$, $b \prec c$, and $b \preceq c$

Remark: The idea of Pareto dominance relations can be extended to nondominated frontiers A and B.

Performance Measures

1. hypervolume indicator I_H
   - measures the hypervolume of the objective space that is weakly dominated by an approximation set. Higher I_H is preferable
   - calculated using a point that is dominated by all approximation sets
   - desirable property: whenever an approximation set A is better than approximation set B, then $I_H(A) > I_H(B)$

Performance Measures

2. R3 indicator
   - incorporates decision maker's preference
   - Given a set of weight vectors Λ

$$I_{R3}(A) = I_{R3}(A, R) = \frac{\sum_{\lambda \in \Lambda} \left[u^*(\lambda, R) - u^*(\lambda, A) \right] / u^*(\lambda, A)}{|\Lambda|} \tag{2}$$

Lower value is preferable.

Outline

Notation (1)

The DARP is modeled on a complete graph $G = (V, A)$
$V = \{v_0, v_1, ..., v_{2n}\}$
$A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$

- n number of customer requests
- m size of vehicle fleet
- v_0 is the depot
- v_i $(i = 1, ..., n)$ a pickup point (origin)
- v_i $(i = n + 1, ..., 2n)$ a delivery point (destination)

(number of vertices: $2n + 1$; set of all nodes N)

Notation (2)

every vertex pair (v_i, v_{i+n}) is associated with a transportation request i

for every arc (v_i, v_j)

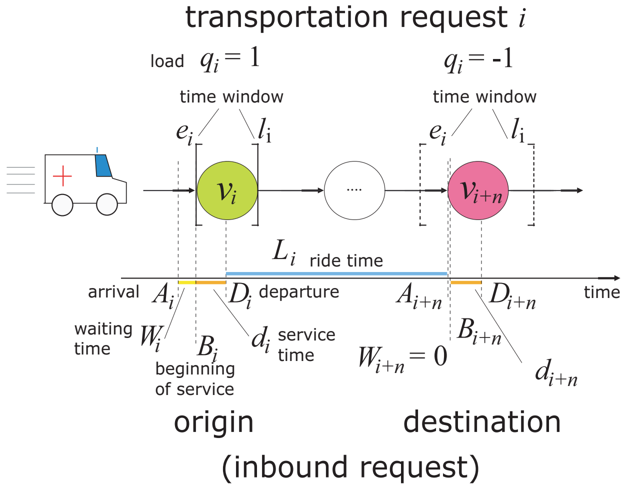
c_{ij} routing cost $(c_{ij} \geq 0)$

t_{ij} travel time $(t_{ij} \geq 0)$

L maximal ride time of a client

T end of the planning horizon

- Outbound request: tight TW on destination
- Inbound request: tight TW on origin



- $L_i = B_{i+n} - D_i$
- $B_i = \max\{e_i, A_i\}$

Objectives

- minimize total routing costs
- minimize mean user ride time

Requirements

1. Every vehicle k (route) has to start and end at v_0
2. For every request i, origin v_i and destination v_{i+n} have to be visited by the same vehicle and v_i is to be visited before v_{i+n}
3. The load q_k must not be greater than vehicle k's capacity Q_k.
4. The total duration of a route k must not exceed T_k
5. The service at v_i has to start within $[e_i, l_i]$ and every vehicle has to depart from and return to the depot within $[e_0, l_0]$
6. The ride time L_i of a client must not exceed L

Outline

Motivation

1. Supported Pareto solutions can be found by means of weighted sum solution methods.
2. Non-supported solutions cannot be found. Non-supported solutions are those that do not lie on the convex hull of the Pareto front.
3. Solution: Path Relinking module departing from weighted sum solutions.

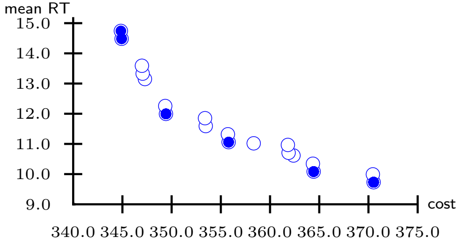


Figure: Supported and non-supported solutions

PR - Algorithm Outline

1. **Initialization**
   - get partial pareto front from exact weighted sum approach or from VNS
2. **Repeat** the following for $itmax$ iterations
   - *Choose* an initial and a guiding solution
   - *Matching*
   - *Path Relinking*
   - *Local Search*

Matching

1. **Edit distance** calculation for all route combinations (Sörensen 2007)
2. **Greedy Matching** (Ho and Gendreau 2006)
   1. Find minimum edit distance e_{ij} (route $i \in$ initial solution, route $j \in$ guiding solution). Forbid selection of all distances e_{ik} with $k \in$ guiding solution and e_{kj} with $k \in$ initial solution.
   2. *Repeat.* until no more routes to match.

Edit Distance Example

route A: abcdefg → route B: bcfamo

Edit Distance Example

route A: abcdefg → route B: bcfamo

step 1: look for b → elim a b cdefg

Edit Distance Example

route A: abcdefg → route B: bcfamo

step 1:	look for b → elim a	b cdefg
step 2:	look for c	bc defg

Edit Distance Example

route A: abcdefg → route B: bcfamo

step 1:	look for b → elim a	b cdefg
step 2:	look for c	bc defg
step 3:	look for f → elim de	bcf g

Edit Distance Example

route A: abcdefg $\longrightarrow$ route B: bcfamo

step 1:	look for b $\longrightarrow$ elim a	b cdefg
step 2:	look for c	bc defg
step 3:	look for f $\longrightarrow$ elim de	bcf g
step 4:	look for a $\longrightarrow$ not found, insert	bcfa g

Edit Distance Example

route A: abcdefg $\longrightarrow$ route B: bcfamo

step 1:	look for b $\longrightarrow$ elim a	b cdefg
step 2:	look for c	bc defg
step 3:	look for f $\longrightarrow$ elim de	bcf g
step 4:	look for a $\longrightarrow$ not found, insert	bcfa g
step 5:	look for m $\longrightarrow$ not found, insert	bcfam g

Edit Distance Example

route A: abcdefg $\longrightarrow$ route B: bcfamo

step 1:	look for b $\longrightarrow$ elim a	b cdefg
step 2:	look for c	bc defg
step 3:	look for f $\longrightarrow$ elim de	bcf g
step 4:	look for a $\longrightarrow$ not found, insert	bcfa g
step 5:	look for m $\longrightarrow$ not found, insert	bcfam g
step 6:	look for o $\longrightarrow$ not found, insert	bcfamo g

Edit Distance Example

route A: abcdefg → route B: bcfamo

step 1:	look for b → elim a	b cdefg
step 2:	look for c	bc defg
step 3:	look for f → elim de	bcf g
step 4:	look for a → not found, insert	bcfa g
step 5:	look for m → not found, insert	bcfam g
step 6:	look for o → not found, insert	bcfamo g
step 7:	elim rest (g)	bcfamo

Edit Distance Example

route A: abcdefg → route B: bcfamo

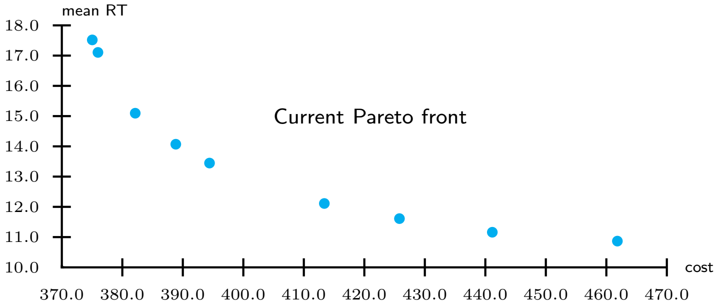
step 1:	look for b → elim a	b cdefg
step 2:	look for c	bc defg
step 3:	look for f → elim de	bcf g
step 4:	look for a → not found, insert	bcfa g
step 5:	look for m → not found, insert	bcfam g
step 6:	look for o → not found, insert	bcfamo g
step 7:	elim rest (g)	bcfamo

Edit Distance: 7

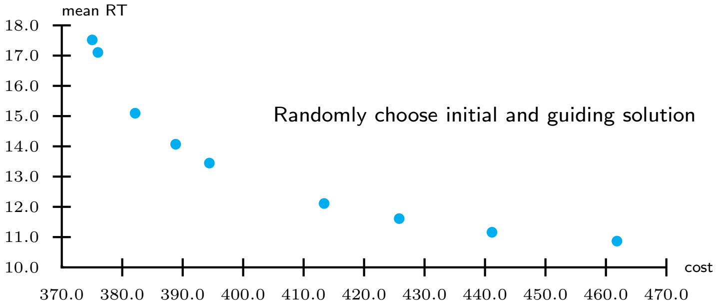
Path Relinking

- **Initialization.** Determine set of requests that are not on the same route in initial and guiding solution.
- **Phase 1.** inter-tour moves (requests)
- **Phase 2.** intra-tour moves (nodes)

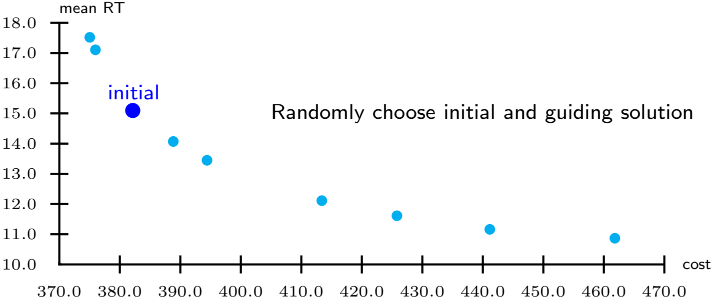
The Path


mean RT
18.0
17.0
16.0
15.0
14.0
13.0
12.0
11.0
10.0
Current Pareto front
cost
370.0
380.0
390.0
400.0
410.0
420.0
430.0
440.0
450.0
460.0
470.0

The Path

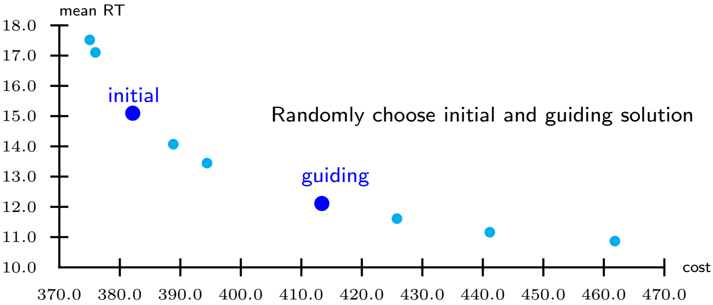

mean RT
18.0
17.0
16.0
15.0
14.0
13.0
12.0
11.0
10.0
cost
370.0
380.0
390.0
400.0
410.0
420.0
430.0
440.0
450.0
460.0
470.0
Randomly choose initial and guiding solution

The Path



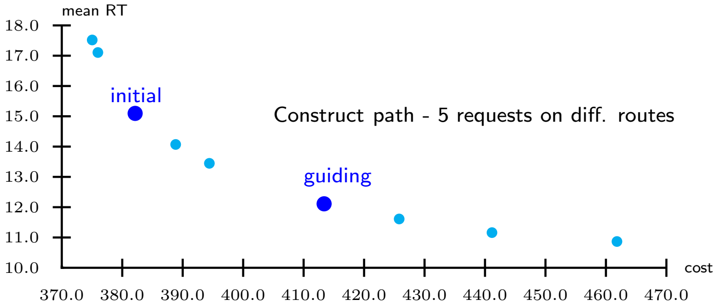
Randomly choose initial and guiding solution

The Path

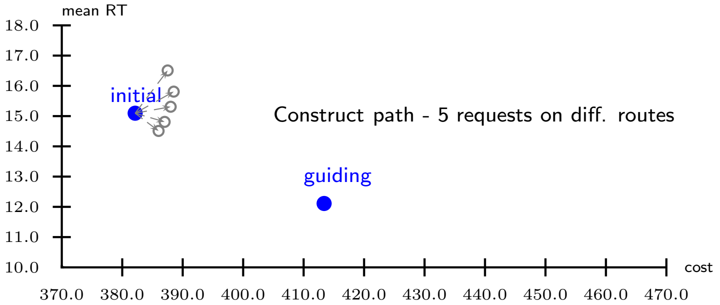


Randomly choose initial and guiding solution

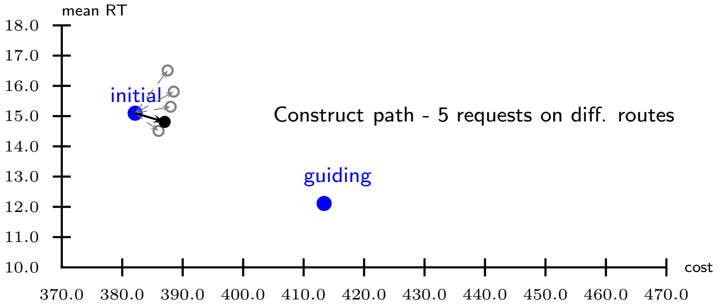
The Path


mean RT
18.0
17.0
16.0
15.0
14.0
13.0
12.0
11.0
10.0
initial
guiding
Construct path - 5 requests on diff. routes
cost
370.0
380.0
390.0
400.0
410.0
420.0
430.0
440.0
450.0
460.0
470.0

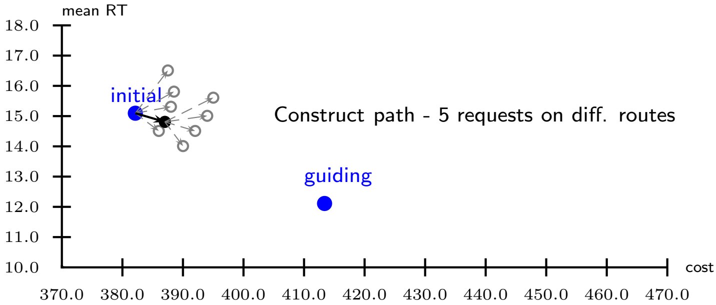
The Path


mean RT
18.0
17.0
16.0
15.0
14.0
13.0
12.0
11.0
10.0
initial
guiding
Construct path - 5 requests on diff. routes
cost
370.0
380.0
390.0
400.0
410.0
420.0
430.0
440.0
450.0
460.0
470.0

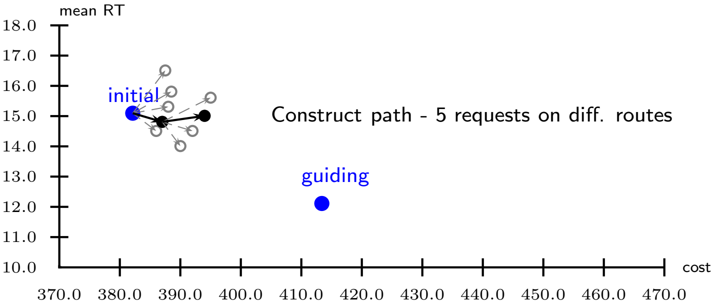
The Path


mean RT
18.0
17.0
16.0
15.0
14.0
13.0
12.0
11.0
10.0
initial
guiding
Construct path - 5 requests on diff. routes
cost
370.0
380.0
390.0
400.0
410.0
420.0
430.0
440.0
450.0
460.0
470.0

The Path

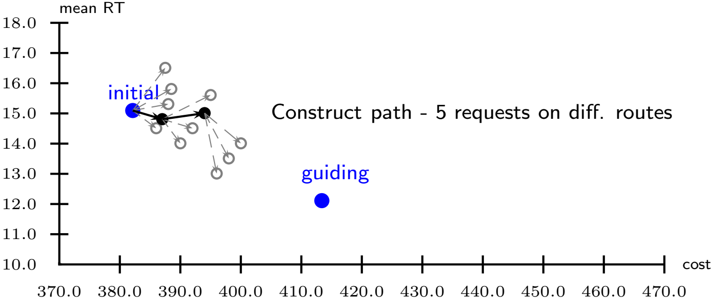

mean RT
18.0
17.0
16.0
15.0
14.0
13.0
12.0
11.0
10.0
initial
guiding
Construct path - 5 requests on diff. routes
cost
370.0
380.0
390.0
400.0
410.0
420.0
430.0
440.0
450.0
460.0
470.0

The Path



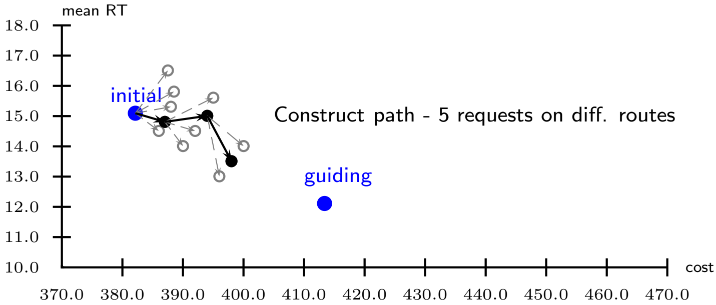
Construct path - 5 requests on diff. routes

The Path

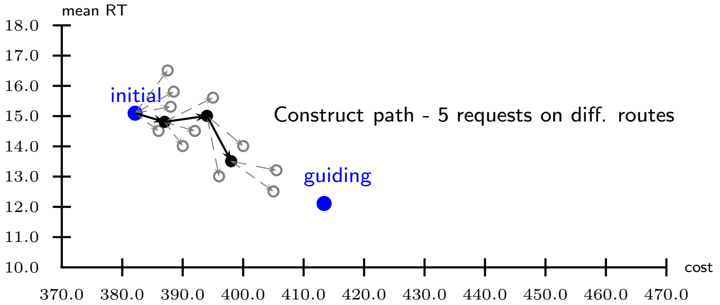


Construct path - 5 requests on diff. routes

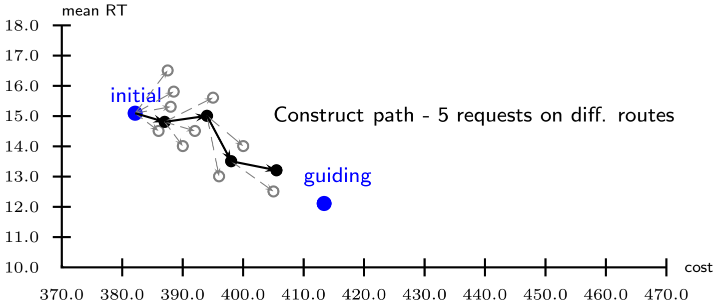
The Path


mean RT
18.0
17.0
16.0
15.0
14.0
13.0
12.0
11.0
10.0
initial
guiding
cost
370.0
380.0
390.0
400.0
410.0
420.0
430.0
440.0
450.0
460.0
470.0
Construct path - 5 requests on diff. routes

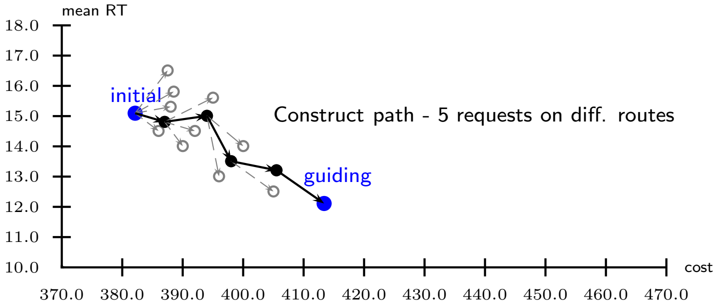
The Path


mean RT
18.0
17.0
16.0
15.0
14.0
13.0
12.0
11.0
10.0
initial
guiding
Construct path - 5 requests on diff. routes
cost
370.0
380.0
390.0
400.0
410.0
420.0
430.0
440.0
450.0
460.0
470.0

The Path

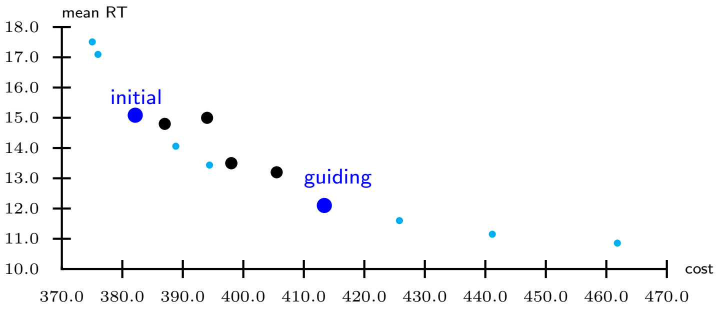

mean RT
18.0
17.0
16.0
15.0
14.0
13.0
12.0
11.0
10.0
initial
guiding
Construct path - 5 requests on diff. routes
cost
370.0
380.0
390.0
400.0
410.0
420.0
430.0
440.0
450.0
460.0
470.0

The Path


mean RT
18.0
17.0
16.0
15.0
14.0
13.0
12.0
11.0
10.0
initial
guiding
Construct path - 5 requests on diff. routes
370.0
380.0
390.0
400.0
410.0
420.0
430.0
440.0
450.0
460.0
470.0
cost

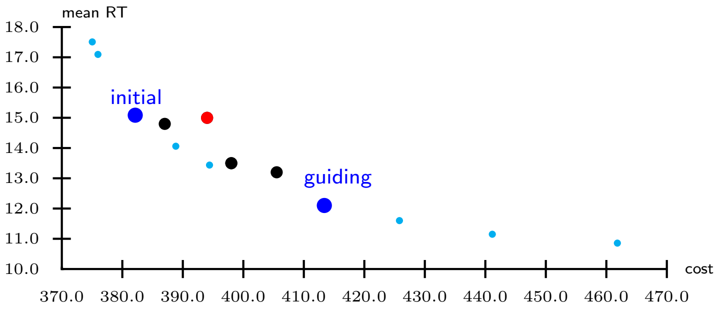
Local Search

- intra-tour moves, first-improvement



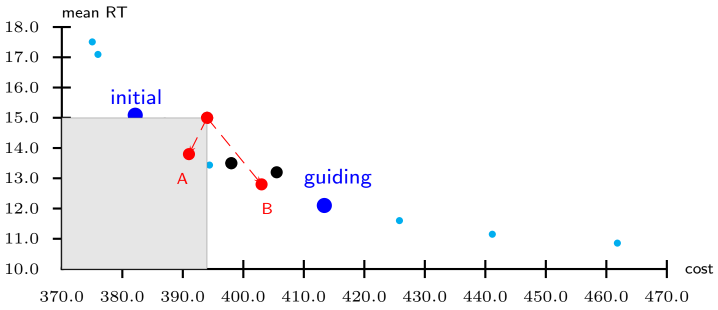
Local Search

- intra-tour moves, first-improvement



Local Search

- intra-tour moves, first-improvement



Local Search - Example

n = 6, start with first origin v_1
0 - **1** - 2 - **7** - 8 - 3 - 9 - 0 RC = 10, mRT = 5
remove origin v_1 and its destination v_7

0 - 2 - 8 - 3 - 9 - 0

insert origin at first possible position wrt TW

0 - **1** - 2 - 8 - 3 - 9 - 0

insert destination v_7 at first possible position wrt v_1
0 - **1** - **7** - 2 - 8 - 3 - 9 - 0 RC = 11, mRT = 4 → incomparable
0 - **1** - 2 - **7** - 8 - 3 - 9 - 0 RC = 10, mRT = 5
0 - **1** - 2 - 8 - **7** - 3 - 9 - 0 RC = 12, mRT = 5
0 - **1** - 2 - 8 - 3 - **7** - 9 - 0 RC = 11, mRT = 3 → TW inf v_7 - STOP
insert origin at next possible position wrt TW

0 - 2 - **1** - 8 - 3 - 9 - 0

insert destination v_7 at first possible position wrt v_1
0 - 2 - **1** - **7** - 8 - 3 - 9 - 0 RC = 9, mRT = 4 → IMPROVEMENT
start with first origin v_2

0 - **2** - 1 - 7 - **8** - 3 - 9 - 0

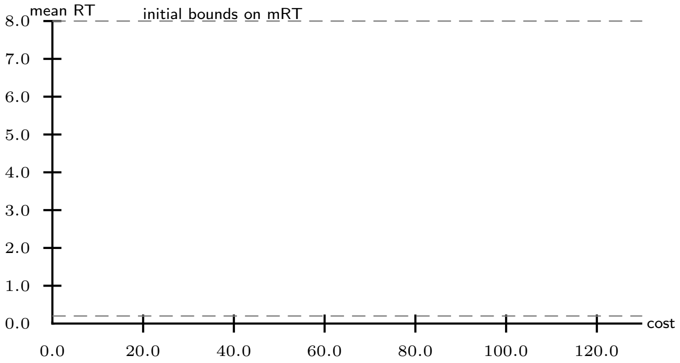
remove origin v_2 and its destination v_8

...

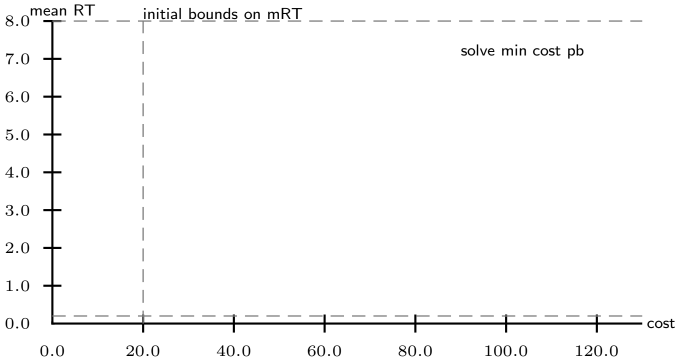
Epsilon Constraint Method

- Branch and cut algorithms for the DARP were developed by
  - Cordeau (2006): 3 index formulation
  - Ropke et al. (2006): more efficient 2 index formulations
- Improved ϵ-constraint method to generate the true pareto frontier proposed by
  - Laumanns et al. (2006)
- B&C algorithm is used in the ϵ-constraint framework to solve the resulting single objective problem instances with additional constraints.

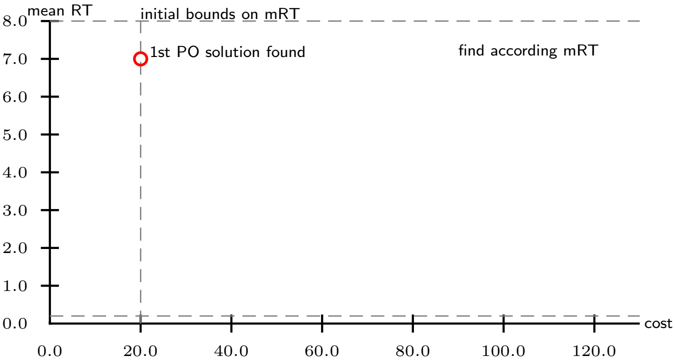
Epsilon Constraint Method



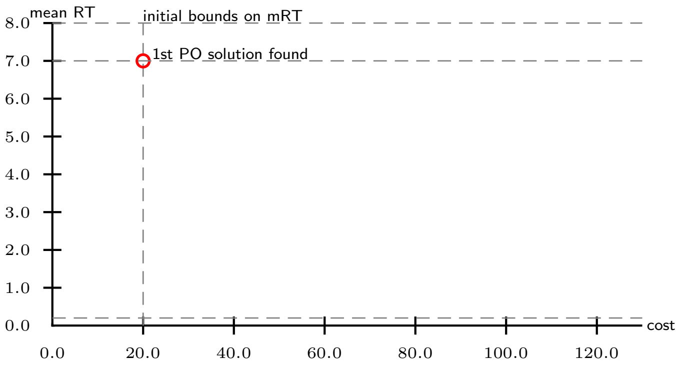
Epsilon Constraint Method



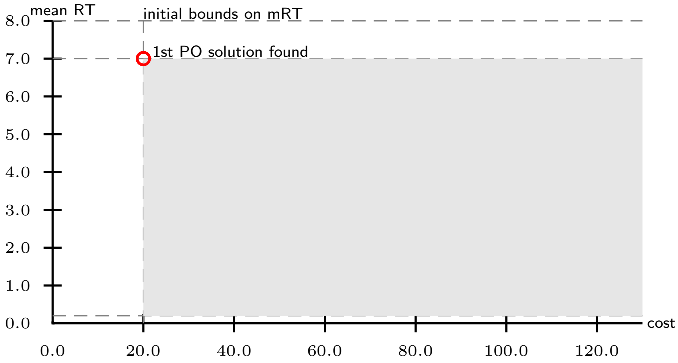
Epsilon Constraint Method



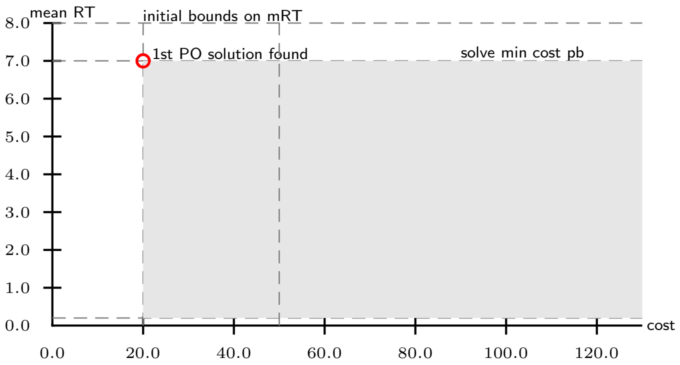
Epsilon Constraint Method



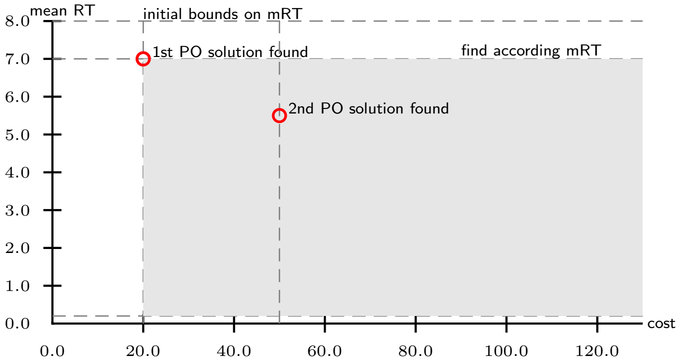
Epsilon Constraint Method



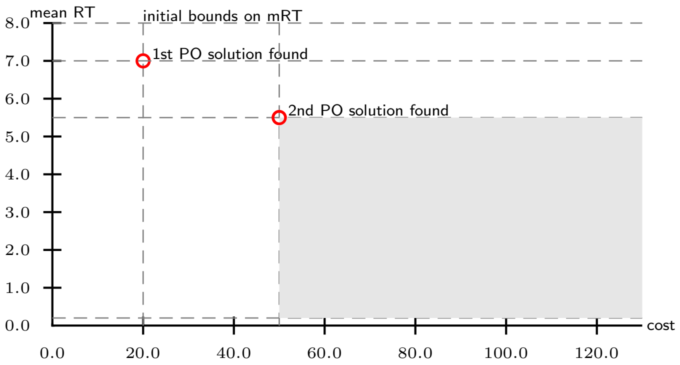
Epsilon Constraint Method



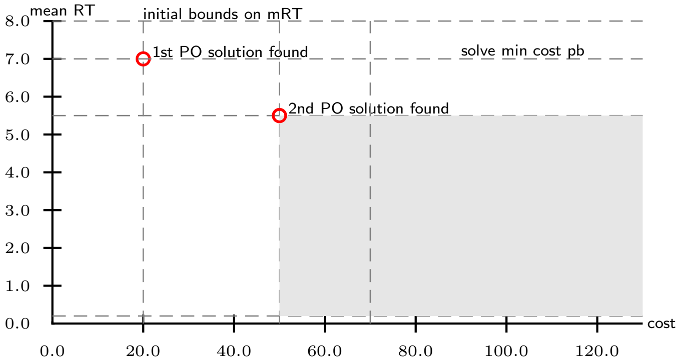
Epsilon Constraint Method



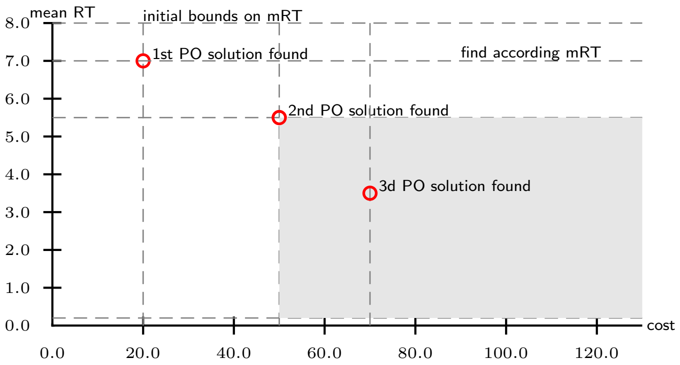
Epsilon Constraint Method



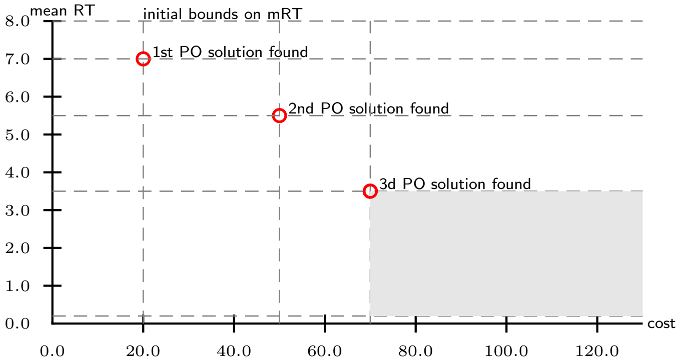
Epsilon Constraint Method



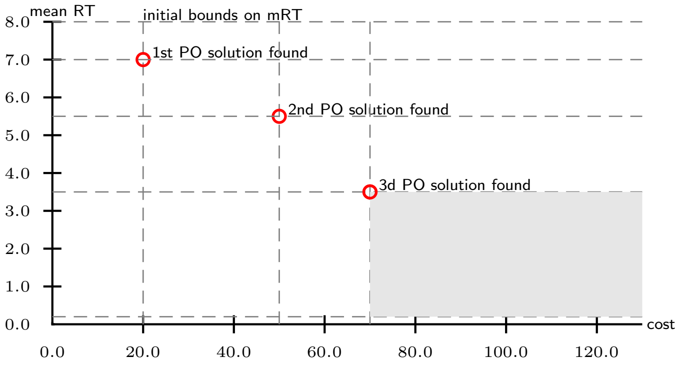
Epsilon Constraint Method



Epsilon Constraint Method



Epsilon Constraint Method



... and so on until no more feasible solution within bounds possible.

Generating Weighted Sum Solutions

- Two methods are applied:
  - Branch and cut algorithm by Ropke et al. (2006)
  - Variable Neighborhood Search
- 100 weight combinations
- Objective function:

$$\min w_1 \frac{1}{C} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} + w_2 \frac{1}{nR} \sum_{i \in P} L_i \tag{3}$$

Variable Neighborhood Search

1. **Initialization.**
   1. do preprocessing (Cordeau 2006)
   2. generate initial solution s
2. **Shaking.** 13 neighborhoods

4 move nh	4 sequence swap nh
4 chain nh	1 zero split nh

3. **Local Search.** only every 5-20 iterations
4. **Move or not.** use ideas from simulated annealing wrt ascending moves.

- $it = 1,000,000$ (first execution); $50,000$ (subsequent).
- Result of previous execution serves as new initial solution.

Outline

Test Instances

- 12 instances of Cordeau (2006)
- 16 - 48 requests
- 2 - 4 vehicles
- load $q_i = 1(-1) \quad \forall i$
- tight time windows
- $L = 30$
- $Q = 3$

... solved by

- Exact Weighted Sum (WS)
- WS + Path Relinking (PR)
- WS + PR + Local Search (LS)
- VNS weighted sum (VNS)
- VNS + PR
- VNS + PR + LS

An Illustrative Example

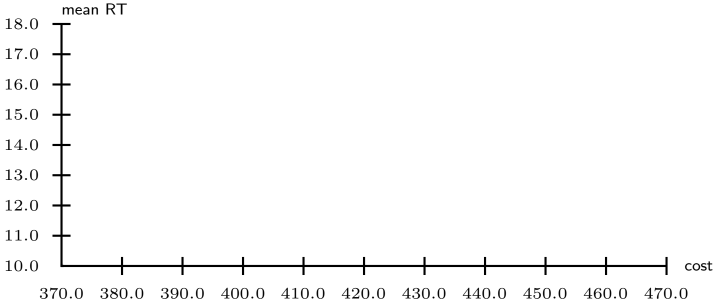


Figure: Instance a4-24

An Illustrative Example

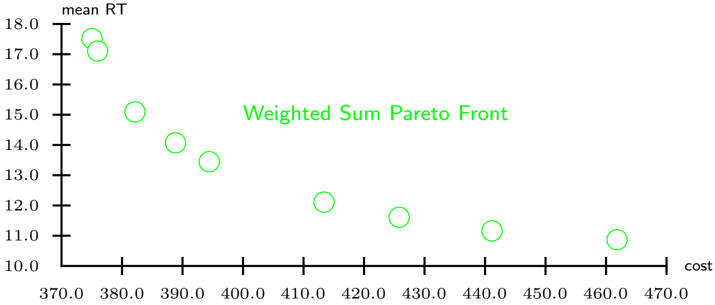


Figure: Instance a4-24

An Illustrative Example

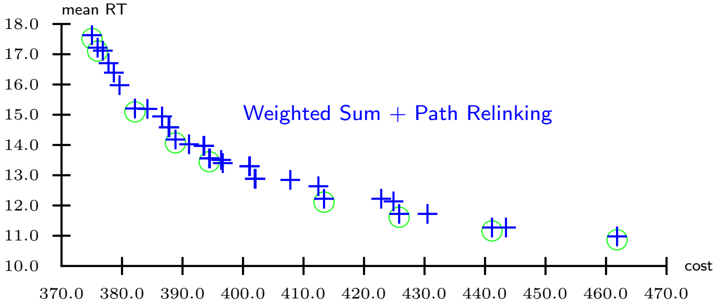


Figure: Instance a4-24

An Illustrative Example

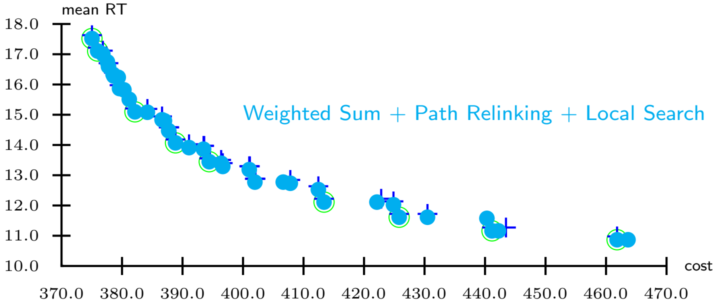


Figure: Instance a4-24

An Illustrative Example

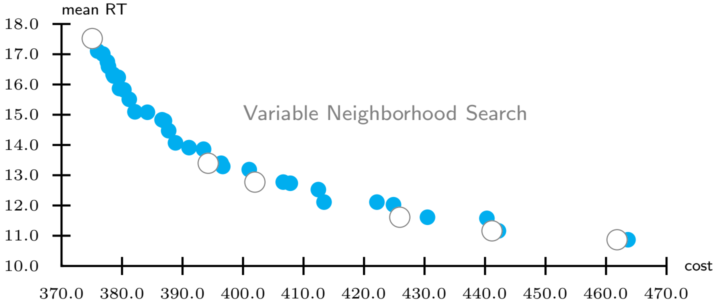


Figure: Instance a4-24

An Illustrative Example

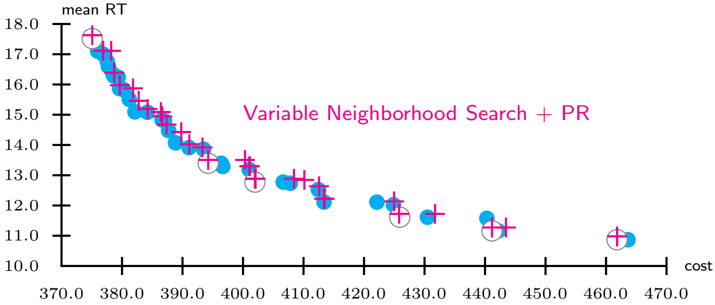


Figure: Instance a4-24

An Illustrative Example

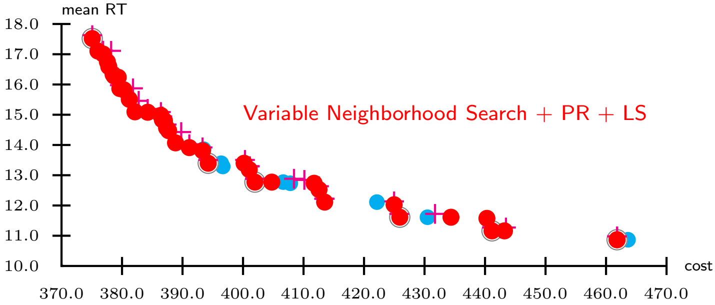


Figure: Instance a4-24

An Illustrative Example

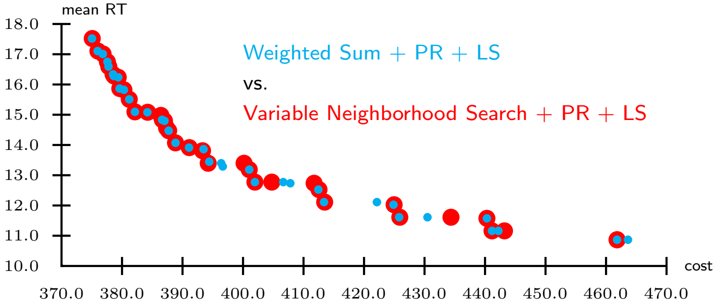


Figure: Instance a4-24

Run Times

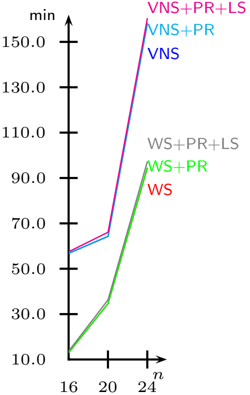


Figure: 2 vehicles

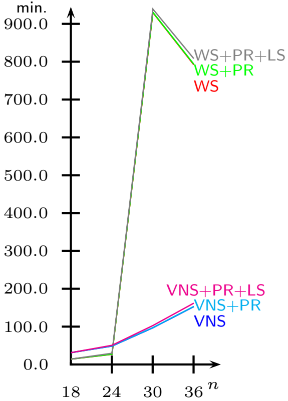


Figure: 3 vehicles

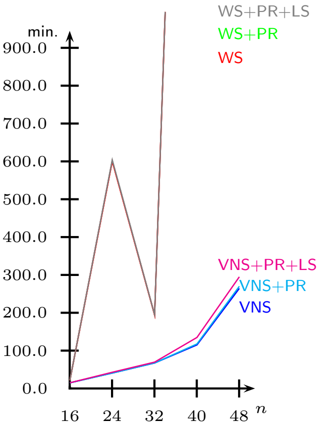
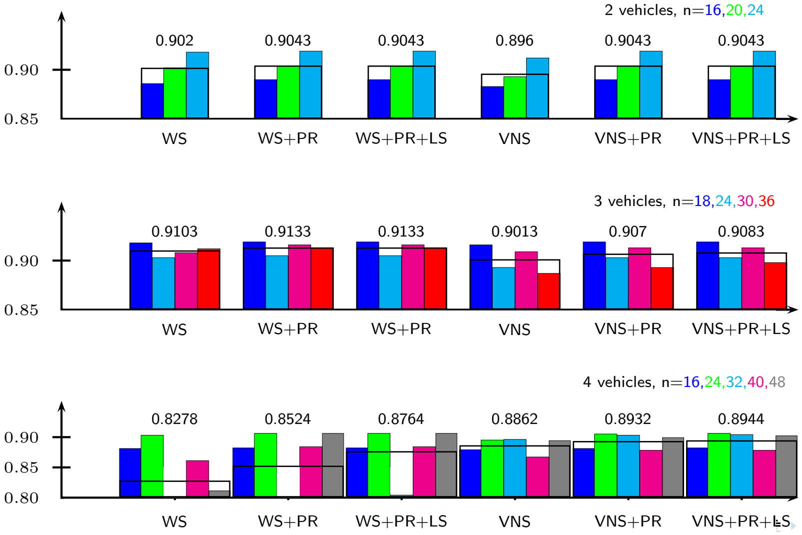
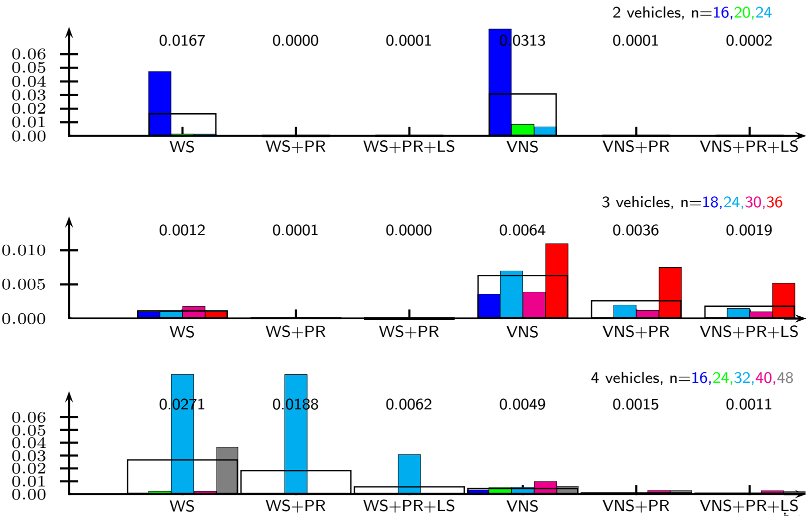


Figure: 4 vehicles

Hypervolume Indicator



R3 Indicator



Conclusion

- An efficient method to generate tradeoff solutions for the MODARP could be developed.
- For small and medium-sized instances PR and PR+LS work effectively.
- The performance metrics indicate that the proposed method can be applied to larger instances with unknown Pareto optimal frontiers.
- Application to real world problem situations is reasonable (Cooperation with Austrian Red Cross).

Outlook

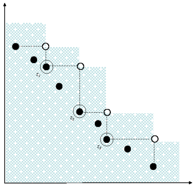


Figure: Nadir Points

- Introduce nadir points and proximity related selection mechanism to choose initial and guiding solutions
- Use other user invconvenience related objectives (maximum user ride time, mean user waiting time, etc.)
- Investigate the 3 and 4 objective case

Thank you for your attention!
Questions?

Performance Measures

2. unary epsilon indicator I_ϵ

- $I_\epsilon(A)$ is the minimum factor ϵ such that if every point in reference set R is multiplied by ϵ, then the resulting approximation set is weakly dominated by A
- $$I_\epsilon(A) = I_\epsilon(A, R) = \inf_{\epsilon \in \mathbb{R}} \left\{ \forall z^2 \in R \exists z^1 \in A : z^1 \preceq_\epsilon z^2 \right\} \quad (4)$$

  where the ϵ-dominance relation is defined as
  $z^1 \preceq_\epsilon z^2 \Leftrightarrow \forall i \in 1, 2, \ldots, n : z_i^1 \leq \epsilon \cdot z_i^2$
- Lower I_ϵ is preferable

Mathematical formulation by Ropke et al. (2007)

$$(1)\ \min \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} \tag{5}$$

$$(2)\ \min \frac{1}{n} \sum_{i \in P} L_i \tag{6}$$

$$\sum_{i \in N} x_{ij} = 1 \qquad \forall j \in P \cup D \tag{7}$$

$$\sum_{j \in N} x_{ij} = 1 \qquad \forall i \in P \cup D \tag{8}$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 2 \qquad \forall S \in \mathcal{S} \tag{9}$$

$$B_j \geq (B_i + d_i + t_{ij}) x_{ij} \qquad \forall i \in N, j \in N \tag{10}$$

$$Q_j \geq (Q_i + q_i) x_{ij} \qquad \forall i \in N, j \in N \tag{11}$$

$$e_i \leq B_i \leq l_i \qquad \forall i \in N \tag{12}$$

$$L_i = B_{n+i} - (B_i + d_i) \qquad \forall in \in P \tag{13}$$

$$t_{i,n+i} \leq L_i \leq L \qquad \forall in \in P \tag{14}$$

$$\max\{0, q_i\} \leq Q_i \leq \min\{Q, Q + q_i\} \qquad \forall i \in N \tag{15}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in N, j \in N \tag{16}$$