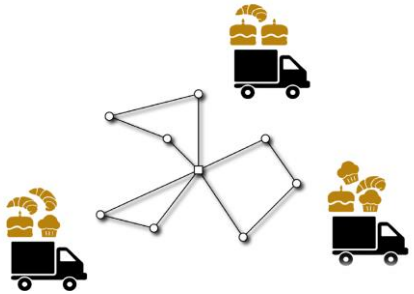
KU LEUVEN

Vehicle routing
A focus on heuristic design

schedulingseminar.com

Jan Christiaens

Greet Vanden Berghe



TRANSPORTATION SCIENCE
Vol. 43, No. 4, November 2009, pp. 408–416
ISSN 0041-1655 | EISSN 1526-5447 | 09 | 4304 | 0408

informs

DOI 10.1287/trsc.1090.0301

Fifty Years of Vehicle Routing

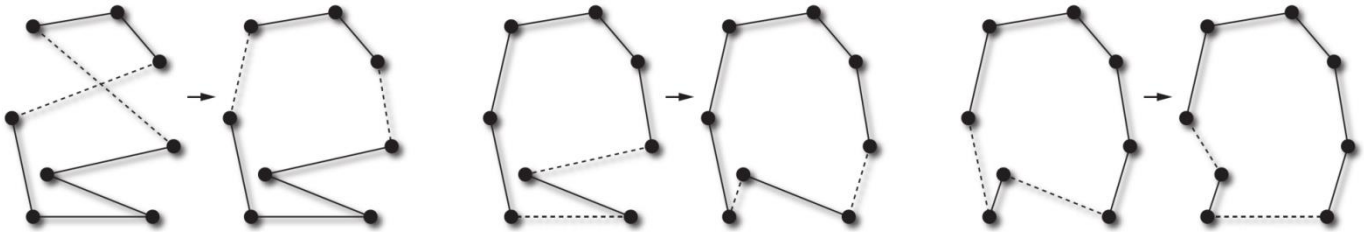
Gilbert Laporte
CIRRELT, Distribution Management, HEC Montréal, 3000, Montréal, Québec H3T 2A7, Canada
gilbert@crt.umontreal.ca

The *Vehicle Routing Problem* (VRP) was introduced 50 years ago by Dantzig and Ramser under the title "The Truck Dispatching Problem." The study of the VRP has given rise to major developments in the fields of exact algorithms and heuristics. In particular, highly sophisticated exact mathematical programming decomposition algorithms and powerful metaheuristics for the VRP have been put forward in recent years. The purpose of this article is to provide a brief account of this development.

Key words: vehicle routing problem; traveling salesman problem; exact algorithms; heuristics; metaheuristics; survey
History: Received: August 2009; revision received: September 2009; accepted: September 2009. Published online in *Articles in Advance* October 21, 2009.

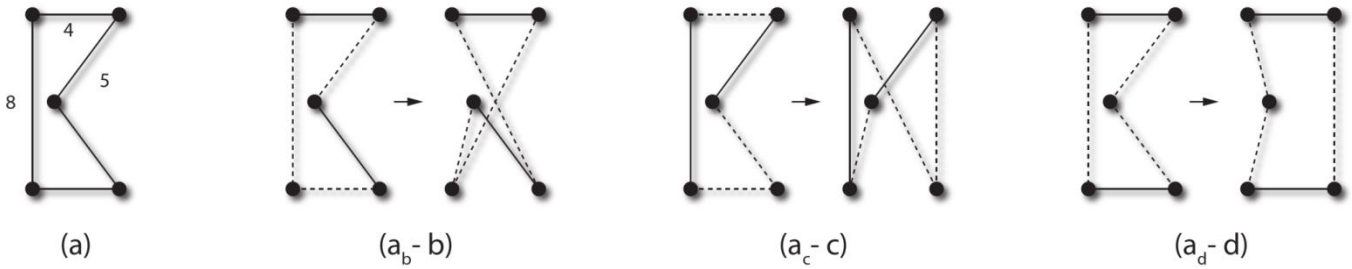
Heuristics

TSP – *2-opt*



Croes, 1958

TSP – *3-opt*

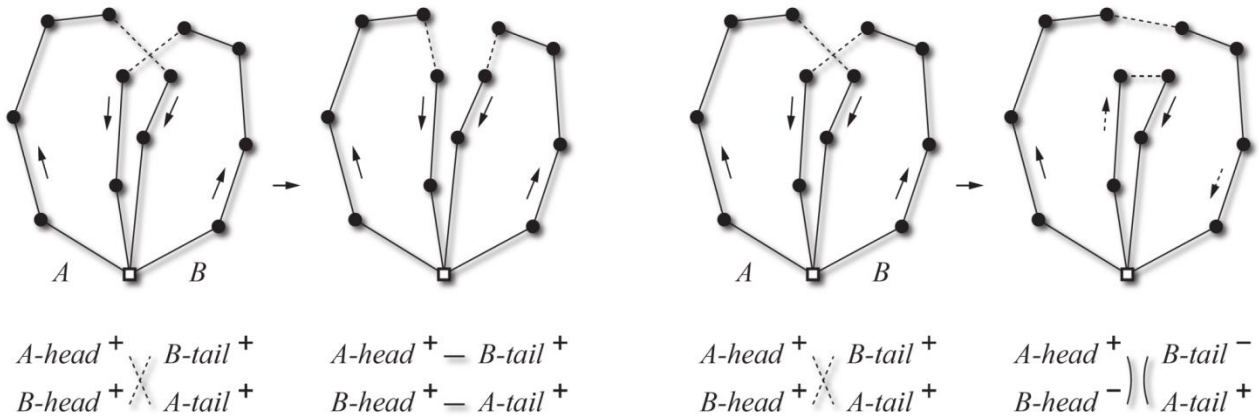


Lin, 1965

TSP – *More heuristics*

- ***k-opt*** (Lin and Kernighan, 1973)
  - Remove k edges, reconnect the possibly reversed strings
- ***or-opt*** (Or, 1976)
  - Relocate a string with maximum cardinality 3

VRP – *Inter-route 2-opt*

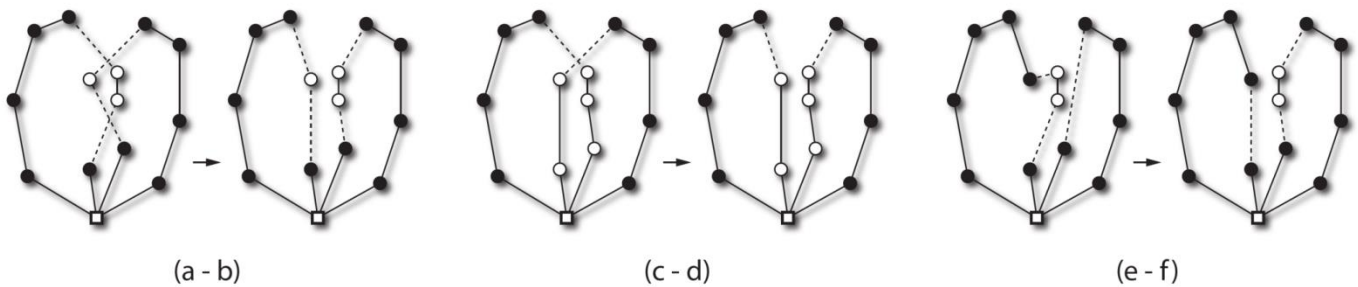


(a - b) (a - c)

VRP – *More neighborhoods*

- ***Or-opt*** (Or, 1976)
  - Relocate 1 string with maximum cardinality 3, possibly to a different route
- ***Single, double, pair+single, double pair*** (Waters, 1987)
  - Relocate 2 strings with maximum cardinality 2, possibly to different routes
- ***Chain-exchange*** (Fahrion and Wrede, 1990)
  - Relocate 2 strings with maximum cardinality $\overline{|t|} / 2$

VRPTW – *Exchange, Cross, Relocate*



Savelsbergh, 1988

VRPTW – *More heuristics*

- ***2-opt**** (Potvin and Rousseau, 1995)
  - inter-route 2-opt without string reversal, or Cross
- ***CROSS-exchange*** (Taillard et al., 1997)
  - Savelsbergh's *Exchange, Cross, Relocate*
  - (+) Relocate may operate as intra-route operator

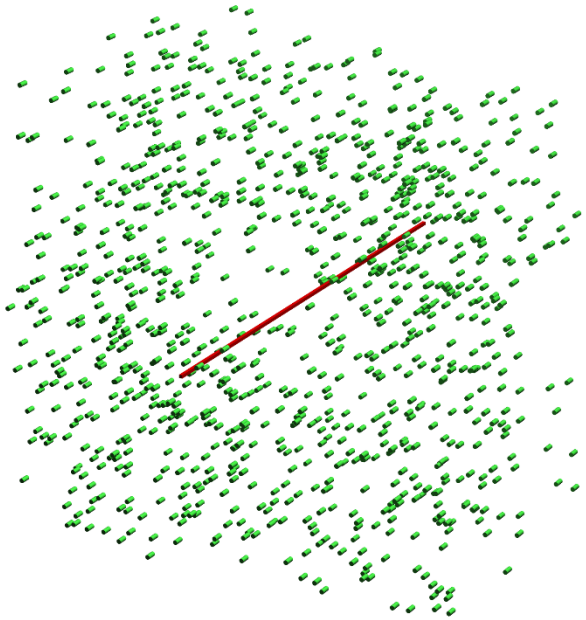
Ruin & recreate (R&R)

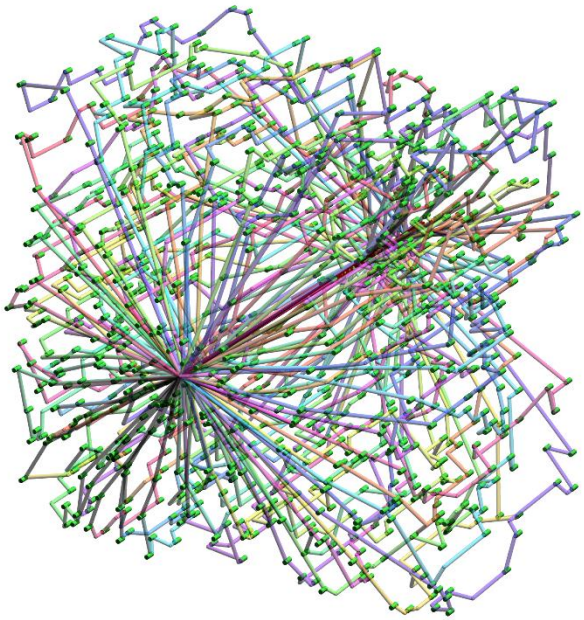
- **Term** introduced by Schrimpf et al.
  - Record Breaking Optimization Results Using the Ruin and Recreate Principle (**2000**)
  - Optimization with Ruin Recreate (**2002**) ***(United States Patent, IBM)***
  - # ruins, 1 recreate

- **Concept** established in older publications
  - Performance of Interconnection **Rip-Up and Reroute** Strategies (**1981**)

VRPTW heuristics

VRPTW heuristics	Reference
Exchange, cross and relocate	Savelsbergh 1988
2-opt*	Potvin and Rousseau 1995
CROSS-exchange	Taillard et al. 1997

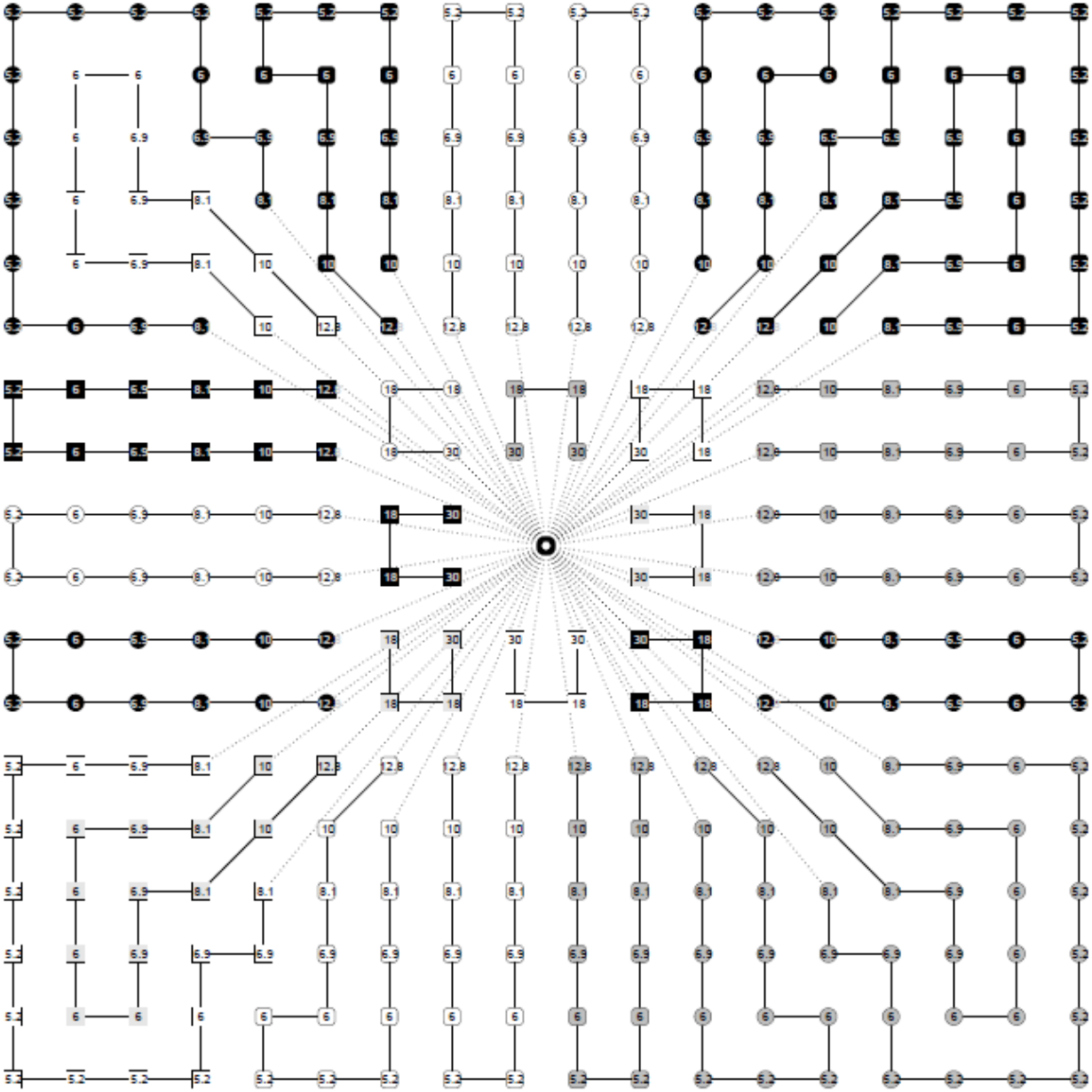




Ruin & recreate heuristics

	Reference	Ruin	Recreate
LNS	Shaw 1998	related removal	Branch & Bound
R&R	Schrimpf et al. 2000	random, radial or string removal	greedy insertion
ALNS	Pisinger and Røpke 2007	random, worst, related, cluster, time-oriented or historical removal	greedy or regret insertion with(out) noise function
SISRs	THIS TALK	adjacent string removal	greedy insertion with blinks

Data



CVRPLIB

Capacitated Vehicle Routing Problem Library

Galgos DI Puc-Rio TPP UI

All Instances | Plots | New Instances | CVRP Challenge | Updates | Posts | Links | About

You are here: Home

Benchmark	Instance	*n*	*K*	*Q*	UB	Opt	Features
Set A (Augerat, 1995)							
Set B (Augerat, 1995)							
Set E (Christofides and Eilon, 1969)							
Set F (Fisher, 1994)							
Set M (Christofides, Mingozzi and Toth, 1979)							
Set P (Augerat, 1995)							
Christofides, Mingozzi and Toth (1979)							
Rochat and Taillard (1995)							
Golden et al. (1998)							
Li et al. (2005)							
Uchoa et al. (2014)							


200

- **February, 23rd 2018-** Improved BKSs reported by Keld Helsgaun using LKH-3: X-n524-k153 … 819-k171 (158265), X-n837-k142 (193810), X-n856-k95 (89002), X-n936-k151 (132926), X-n957-k87 (85478) an…

- **August, 14th 2017 -** Improved BKSs reported (Toffolo, Vidal, Wauters (2017), Heuristics for … appear as a working paper): X-n294-k50 (47161), X-n322-k28 (29834), X-n327-k20 (27532), X-n344-k43 (42056), …

- **November, 23rd 2016 -** Improved BKSs reported by Jan Christiaens obtained using ASB-RR … 6187), X-n449-k29 (55269), X-n459-k26 (24173), X-n491-k59 (66510), X-n502-k39 (69230), X-n536-k96 (94988), X-n561-k42 (42722), X-n599-k92 (108490), X-n627-k43 (62210), X-n670-k130 (146451), X-n685-k75 (68261), X-n701-k44 (81934), X-n716-k35 (43414), X-n749-k98 (77365), X-n766-k71 (114525), X-n783-k48 (72445), X-n801-k40 (73331), X-n837-k142 (193813), X-n876-k59 (99331), X-n895-k37 (53946), X-n916-k207 (329247), X-n936-k151 (132926) and X-n957-k87 (85482).

- **November, 6th 2016 -** Improved BKS to X-n256-k16 (18839) reported by Túlio Toffolo and Thibaut Vidal. This contradicts a previous claim that a solution with value 18880 was optimal. After investigation, it was found that a typo in a script made the BCP method (Pecin et al., 2014) to be run with a minimum of 17 routes. The improving solution has 16 routes. The status of that instance is also corrected to "open".

- **April, 25th 2016 -** Proven optimal solutions by the BCP method (Pecin et al., 2014): X-n214-k11 (10856) and X-n233-k16 (19230).

- **April, 4th 2016 -**Improved and proven optimal solutions by the BCP method (Pecin et al., 2014): X-n331-k15 (31102) and X-n439-k37 (36391).

- **April, 1st 2016 -** Improved BKSs reported by Jan Christiaens: X-n322-k28 (29848), X-n336-k84 (139135), X-n344-k43 (42068), X-n351-k40 (25928), X-n384-k52 (65981), X-n401-k29 (66202), X-n449-k29 (55302), X-n459-k26 (24179), X-n480-k70 (89458), X-n491-k59 (66520), X-n502-k39 (69232), X-n536-k96 (94991), X-n548-k50 (86701), X-n573-k30 (50719), X-n586-k159 (190423), X-n599-k92 (108541), X-n613-k62 (59556), X-n627-k43 (62217), X-n641-k35 (63737), X-n670-k130 (146477), X-n685-k75 (68276), X-n701-k44 (81962), X-n716-k35 (43441), X-n733-k159 (136250), X-n749-k98 (77402), X-n766-k71 (114534), X-n783-k48 (72453), X-n801-k40 (73344), X-n819-k171 (158267), X-n837-k142 (193836), X-n856-k95 (89007), X-n876-k59 (99360), X-n895-k37 (53948), X-n916-k207 (329299), X-n957-k87 (85517), X-n979-k58 (119008) and X-n1001-k43 (72404).

- **June, 22nd 2015 -** Improved BKSs reported by Jan Christiaens: X-n322-k28 (29854), X-n336-k84 (139165), X-n344-k43 (42073), X-n351-k40 (25936), X-n384-k52 (66021), X-n401-k29 (66219), X-n480-k70 (89488), X-n491-k59 (66523), X-n536-k96 (95062), X-n561-k42 (42754), X-n573-k30 (50726), X-n586-k159 (190454), X-n599-k92 (108600), X-n627-k43 (62264), X-n641-k35 (63760), X-n670-k130 (146570), X-n685-k75 (68291), X-n701-k44 (81997), X-n716-k35 (43491), X-n733-k159 (136313), X-n749-k98 (77423), X-n766-k71 (114566), X-n783-k48 (72547), X-n801-k40 (73367), X-n819-k171 (158298), X-n837-k142 (193933), X-n856-k95 (89040), X-n876-k59 (99424), X-n895-k37 (54030), X-n916-k207 (329394), X-n936-k151 (132946), X-n957-k87 (85566), X-n979-k58 (119072) and X-n1001-k43 (72477).

SISRs

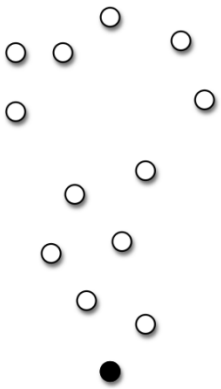
1. Ruin
2. Recreate
3. Fleet minimization

1. SISRs **ruin**

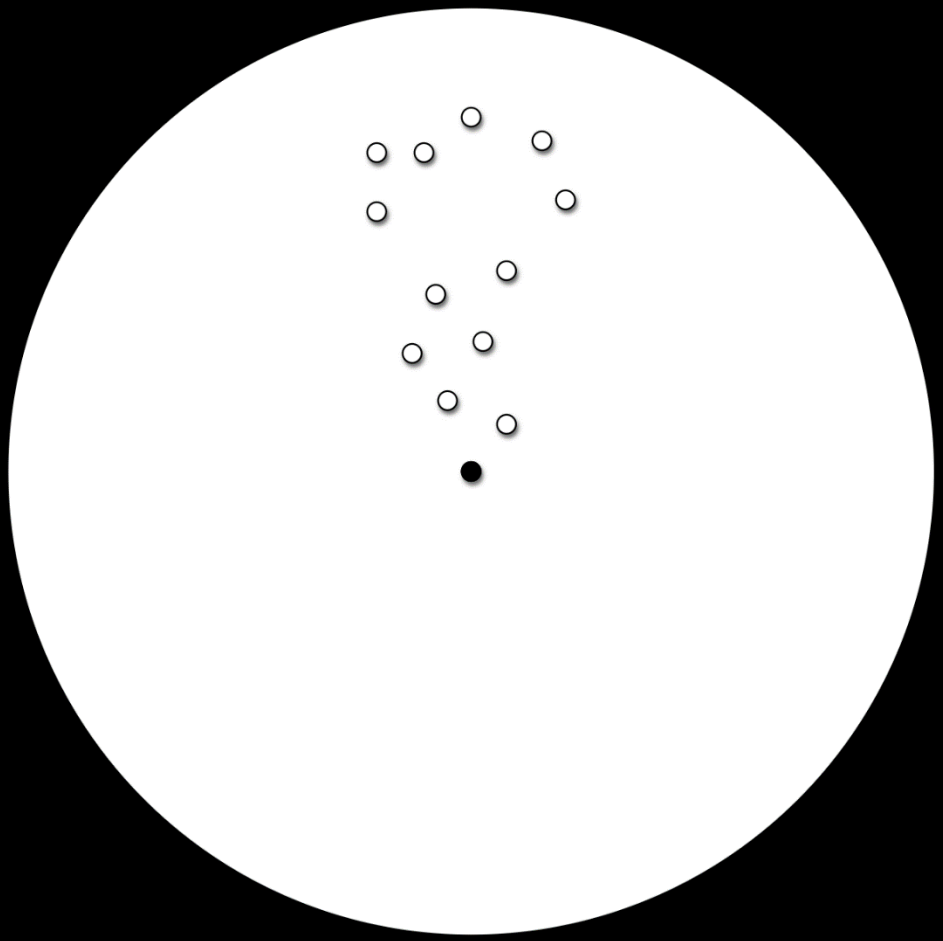
Remove customers – Adjacent string removal

- **capacity** slack - vehicles gain free capacity
- **spatial** slack - vehicles are detached from regions

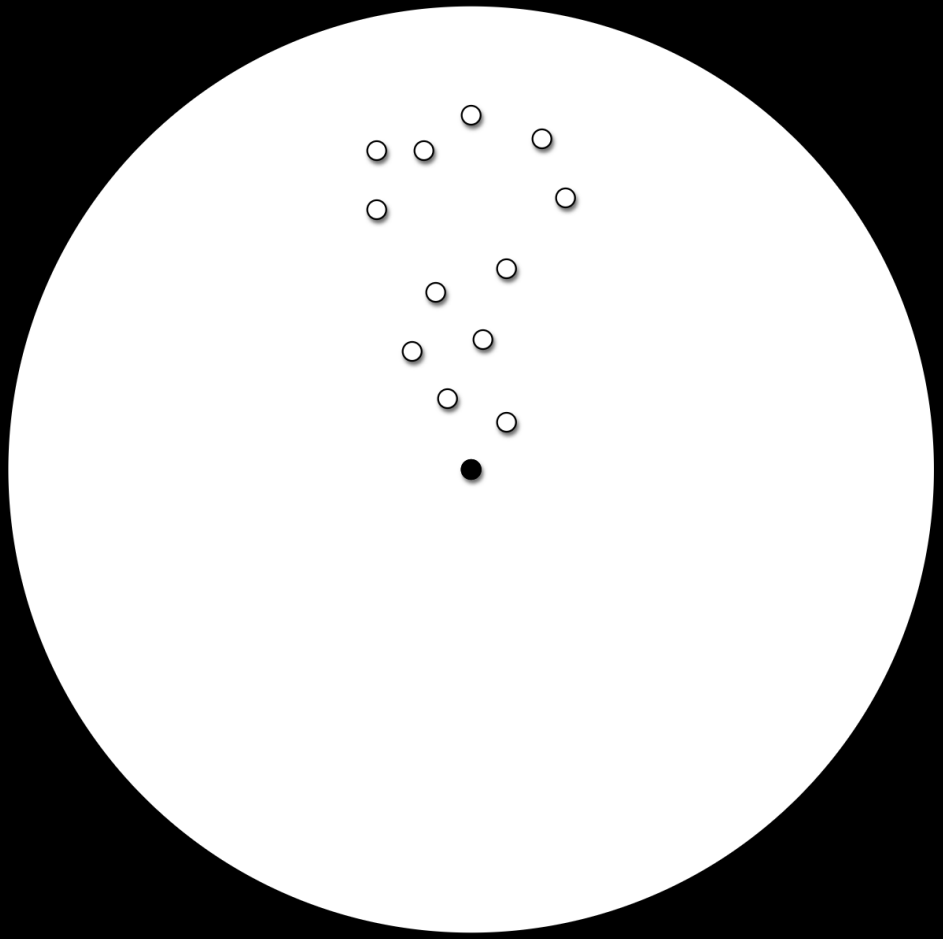
Spatial slack



Spatial slack – Maximum distance



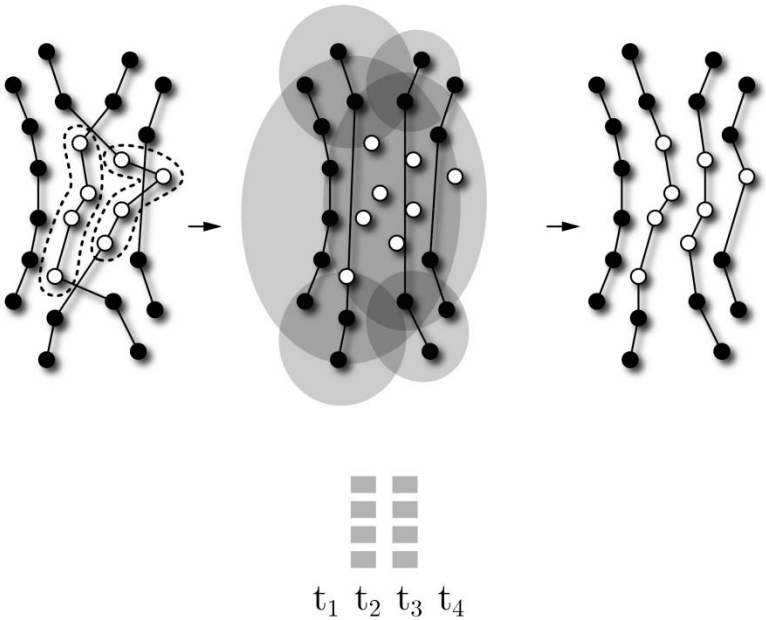
Spatial slack – Maximum distance



Spatial slack – Total distance

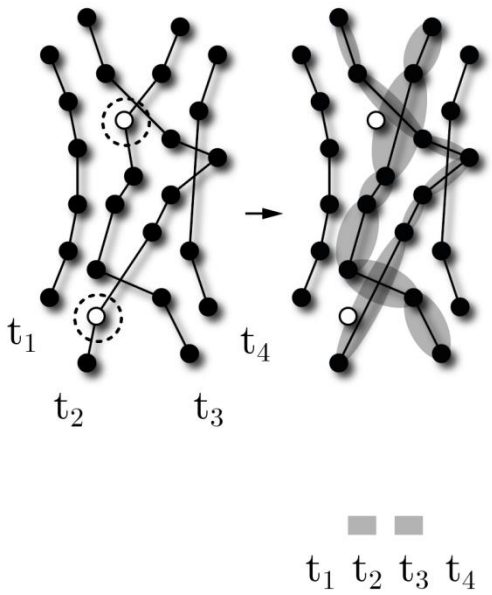
Spatial slack – Total distance

SISRs **ruin** – Adjacent string removal



Ruin methods - Alternatives

Small number

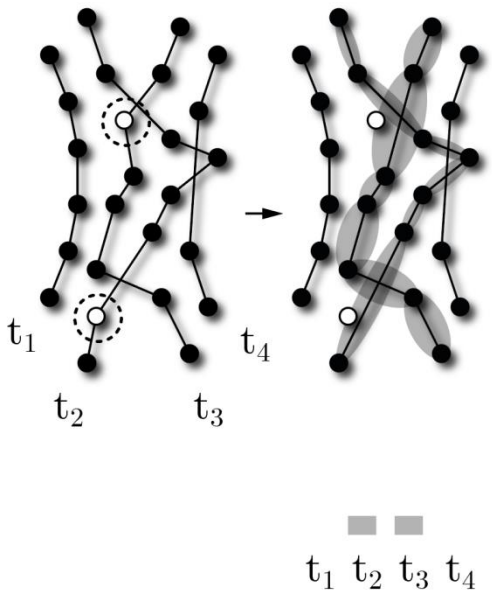


Random

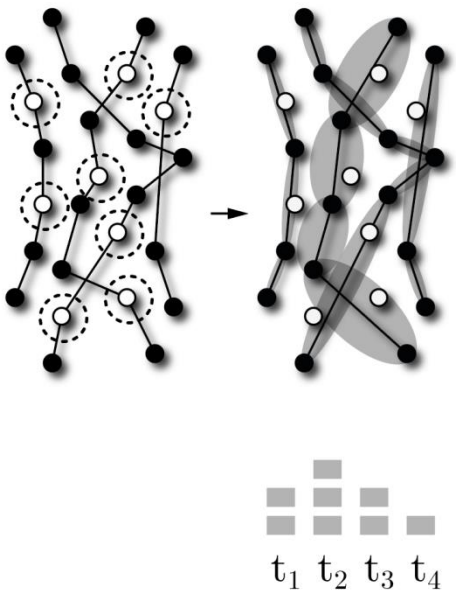
Radial

Ruin methods - Alternatives

Small number



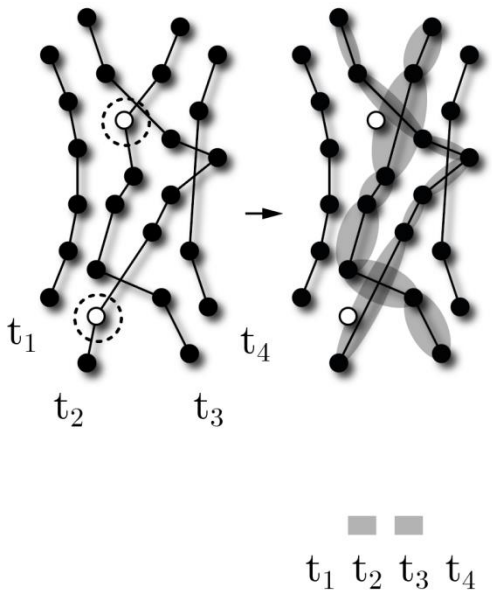
Random



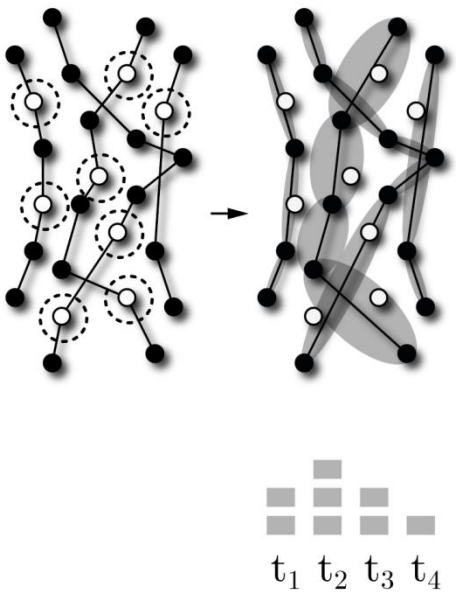
Radial

Ruin methods - Alternatives

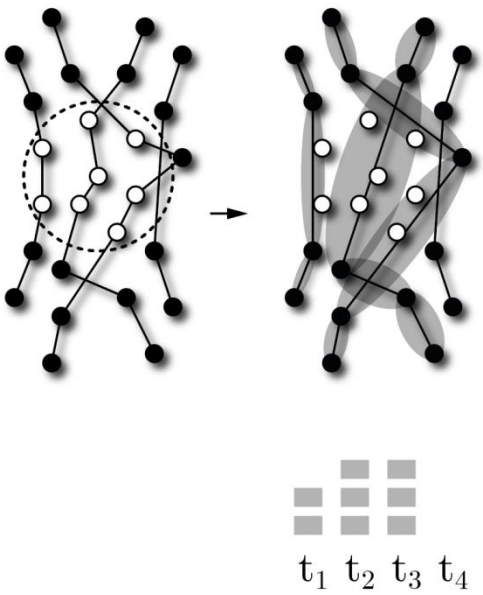
Small number



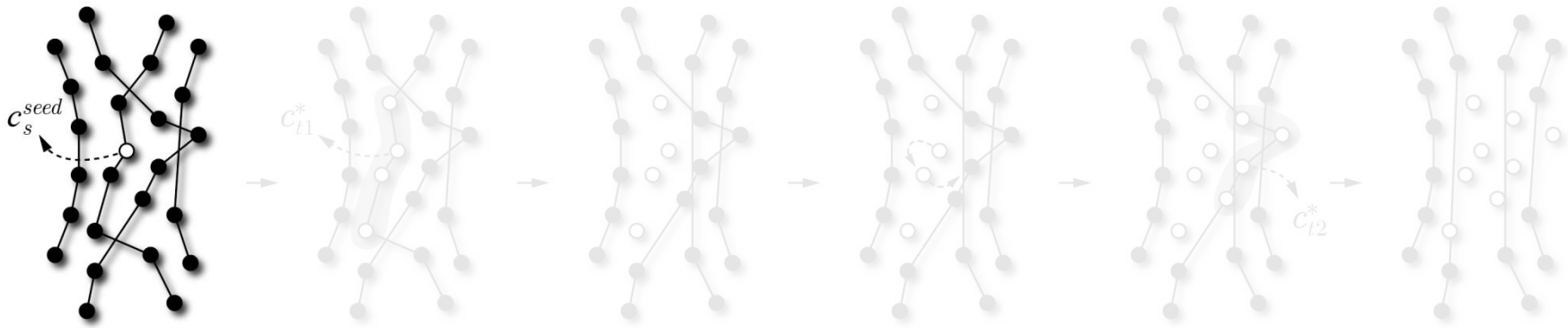
Random



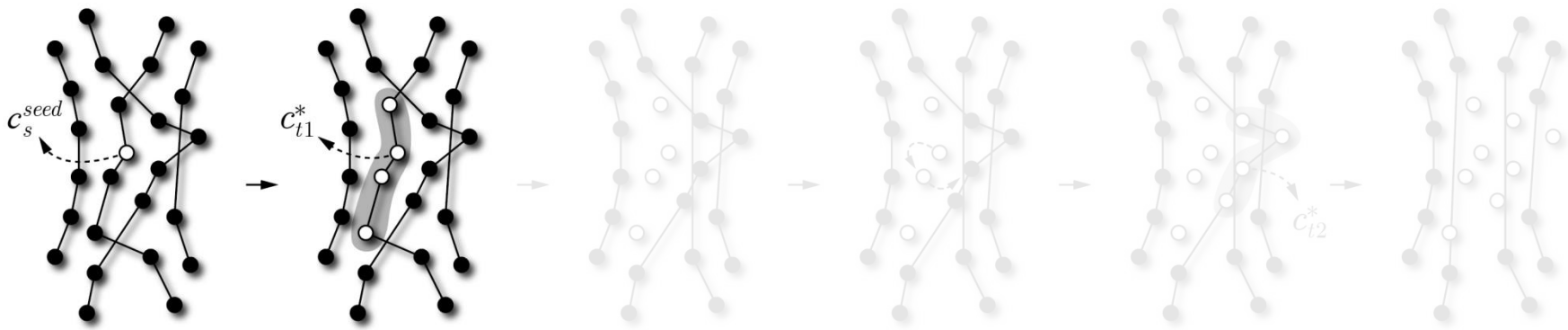
Radial



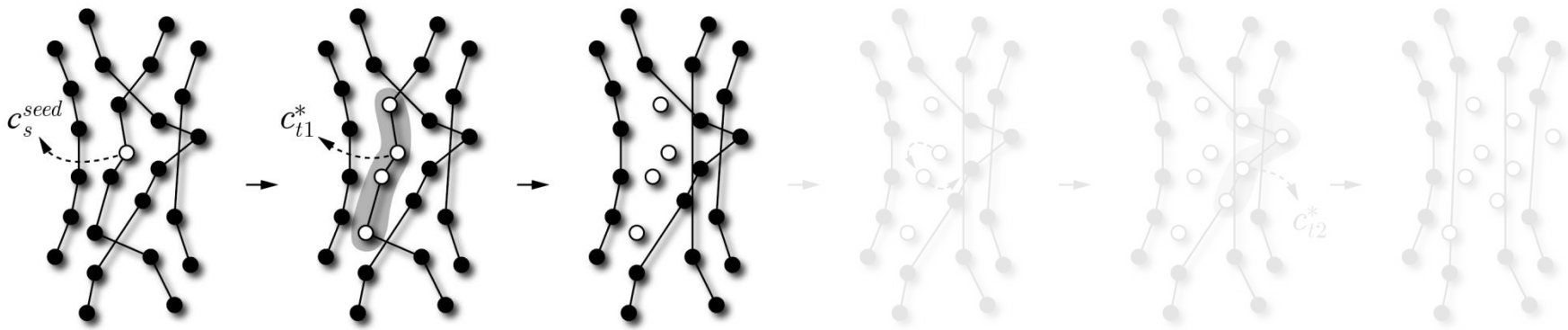
SISRs **ruin** – Adjacent string removal



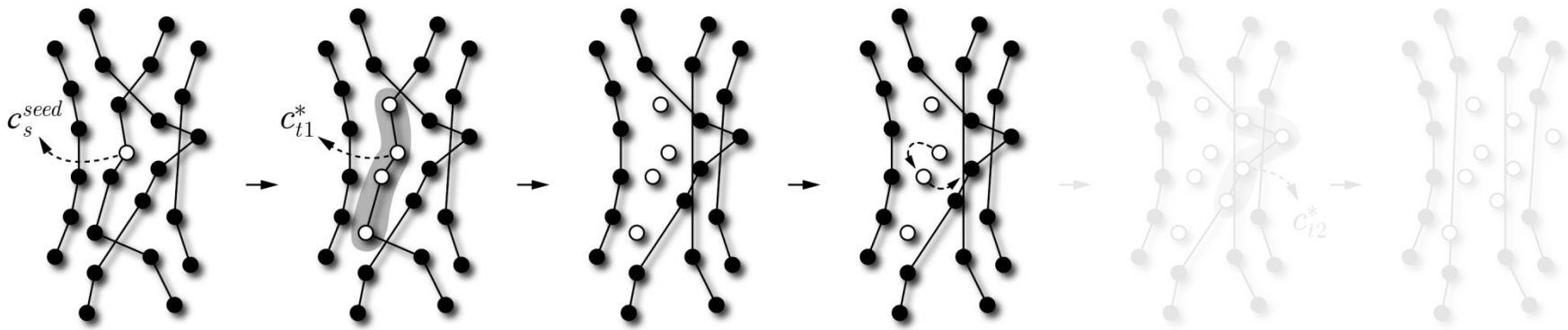
SISRs **ruin** – Adjacent string removal



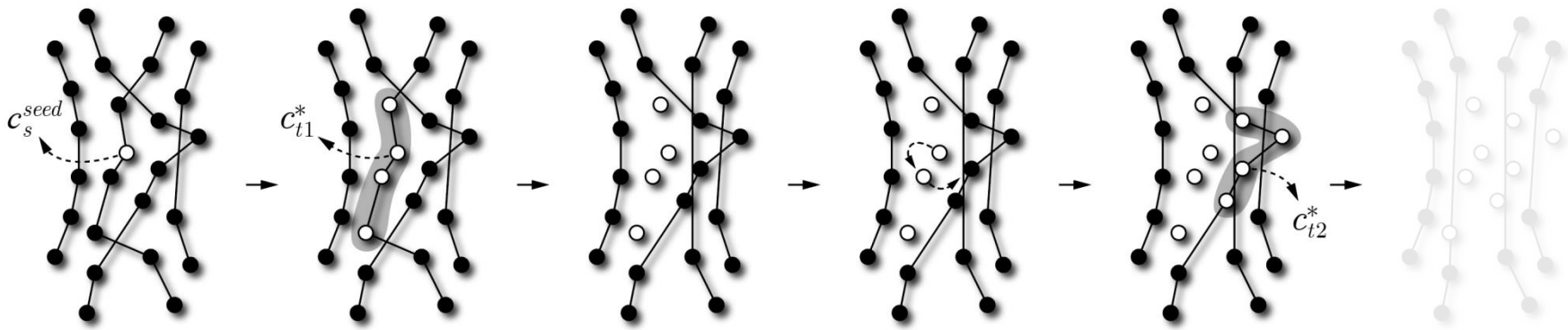
SISRs **ruin** – Adjacent string removal



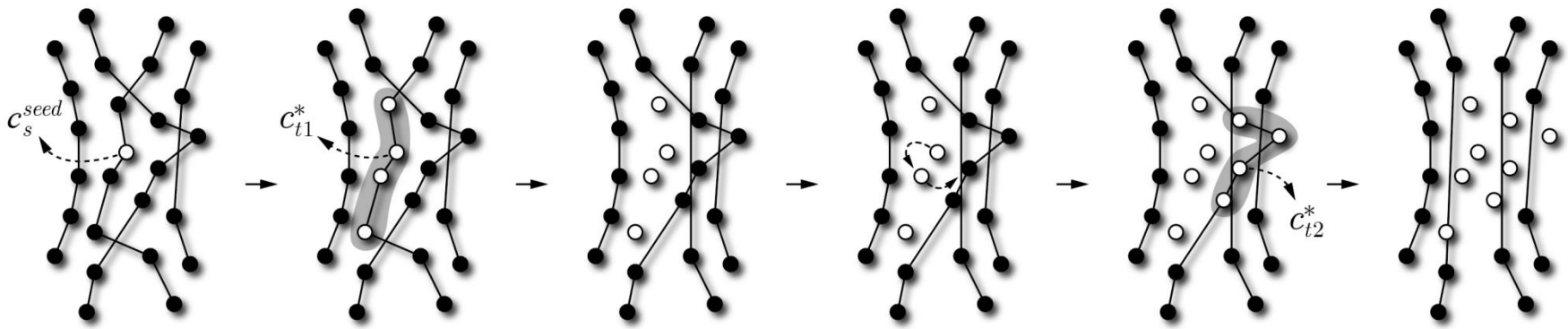
SISRs **ruin** – Adjacent string removal



SISRs **ruin** – Ajacent string removal



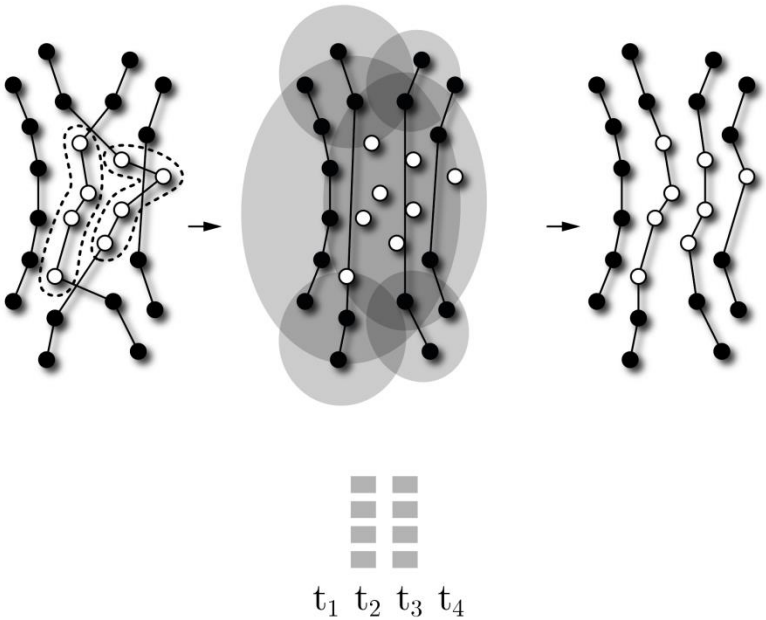
SISRs **ruin** – Ajacent string removal



2. SISRs **recreate**

- **insert customers** - greedy insertion with blinks
- **options enabled by** slack - strongly dependent on **ruin**

SISRs **recreate**



SISRs **recreate** – Greedy insertion with blinks

Greedy insertion:

- **always** at best position
- very greedy ☹

Greedy insertion with blinks:

Heuristic-Biased Stochastic Sampling

```
procedure RECREATE(s)
    sort(A)
    for c ∈ A do
        P ← NULL
        for t ∈ T (which can serve c) do
            for P_t in t do
                if P = NULL or costAt(P_t) < costAt(P) then
                    P ← P_t
        if P = NULL then
            T ← T ∪ new tour t
            P ← position in t
        insert c at P
        A ← A \ {c}
end procedure
```

SISRs **recreate** – Greedy insertion with blinks

Greedy insertion:

- **always** at best position
- very greedy ☹

Greedy insertion with blinks:

- **mostly** at best position
- less greedy ☺

Heuristic-Biased Stochastic Sampling

```
procedure RECREATE(s)
    sort(A)
    for c ∈ A do
        P ← NULL
        for t ∈ T (which can serve c) do
            for P_t in t do
                if U(0,1) < 1 − β then
                    if P = NULL or costAt(P_t) < costAt(P) then
                        P ← P_t
        if P = NULL then
            T ← T ∪ new tour t
            P ← position in t
        insert c at P
        A ← A \ {c}
end procedure
```

SISRs Recreate – Greedy insertion with blinks

Greedy insertion:

- **always** at best position
- very greedy ☹

Greedy insertion with blinks:

- **mostly** at best position
- less greedy ☺

Heuristic-Biased Stochastic Sampling

- HBSS (Bresina 1996)
- rank all options (best to worst)
- exponentially decreasing probability

```
procedure RECREATE(s)
    sort(A)
    for c ∈ A do
        P ← NULL
        for t ∈ T (which can serve c) do
            for P_t in t do
                if U(0,1) < 1 − β then
                    if P = NULL or costAt(P_t) < costAt(P) then
                        P ← P_t
        if P = NULL then
            T ← T ∪ new tour t
            P ← position in t
        insert c at P
        A ← A \ {c}
end procedure
```

SISRs **recreate** – Greedy insertion with blinks

Greedy insertion:

- **always** at best position
- very greedy ☹

Greedy insertion with blinks:

- **mostly** at best position
- less greedy ☺
- **rank-based** selection probabilities
- **without ranking** all options!! ☺

Heuristic-Biased Stochastic Sampling

- HBSS (Bresina 1996)
- rank all options (best to worst)
- exponentially decreasing probability

```
procedure RECREATE(s)
    sort(A)
    for c ∈ A do
        P ← NULL
        for t ∈ T (which can serve c) do
            for P_t in t do
                if U(0,1) < 1 − β then
                    if P = NULL or costAt(P_t) < costAt(P) then
                        P ← P_t
        if P = NULL then
            T ← T ∪ new tour t
            P ← position in t
        insert c at P
        A ← A \ {c}
end procedure
```

3. SISRs **fleet minimization**

- **Fleet minimization** - absences-based acceptance criterion


FedEx
Express
FedEx
Express
FedEx
FedEx
Express
FedEx
Express

SISRs **fleet minimization**

- VRPTW, PDPTW …

- Not strictly related to distance → two-stage approach
    1. **Fleet minimization**
    2. **Distance minimization**

- Nagata, Bräysy, and Dullaert (2010) VRPTW

  Nagata and Kobayashi (2010) PDPTW
    1. Remove vehicle
    2. “Squeeze” until all customers are served
    3. Repeat

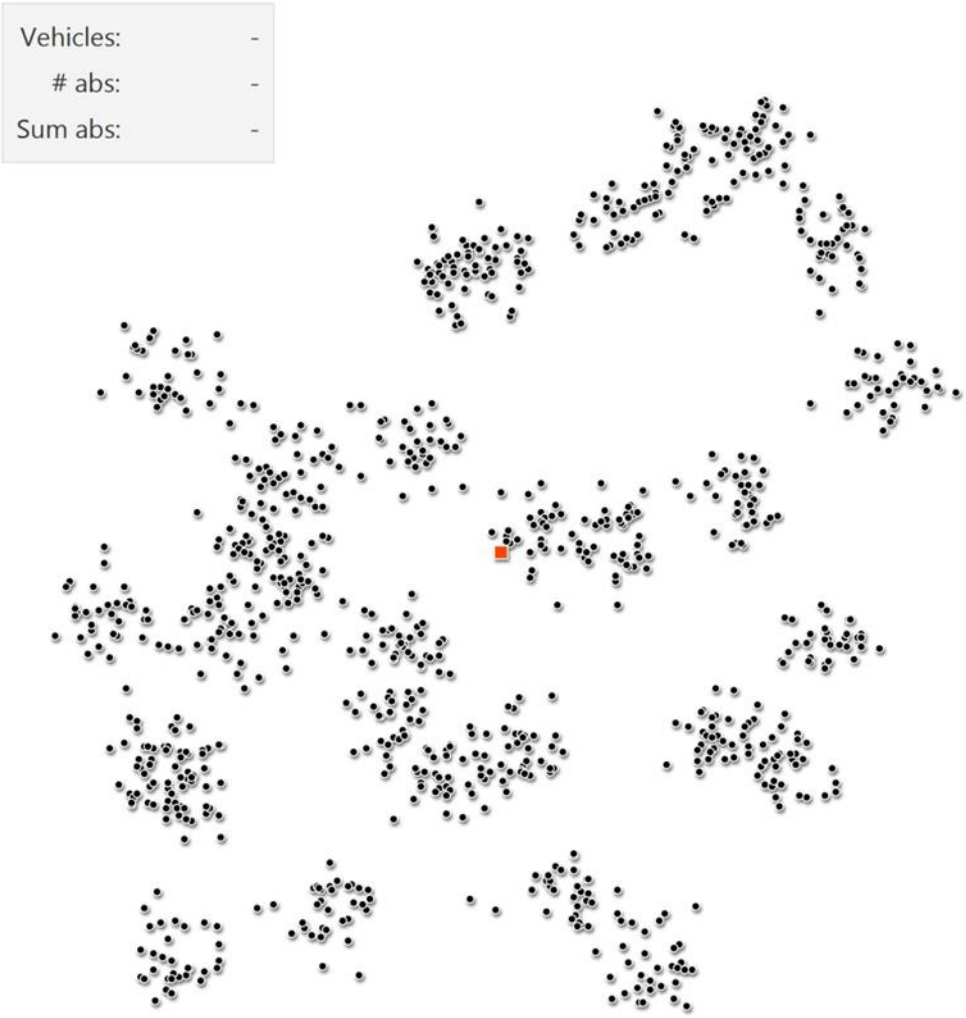
Fleet minimization – 1st attempt

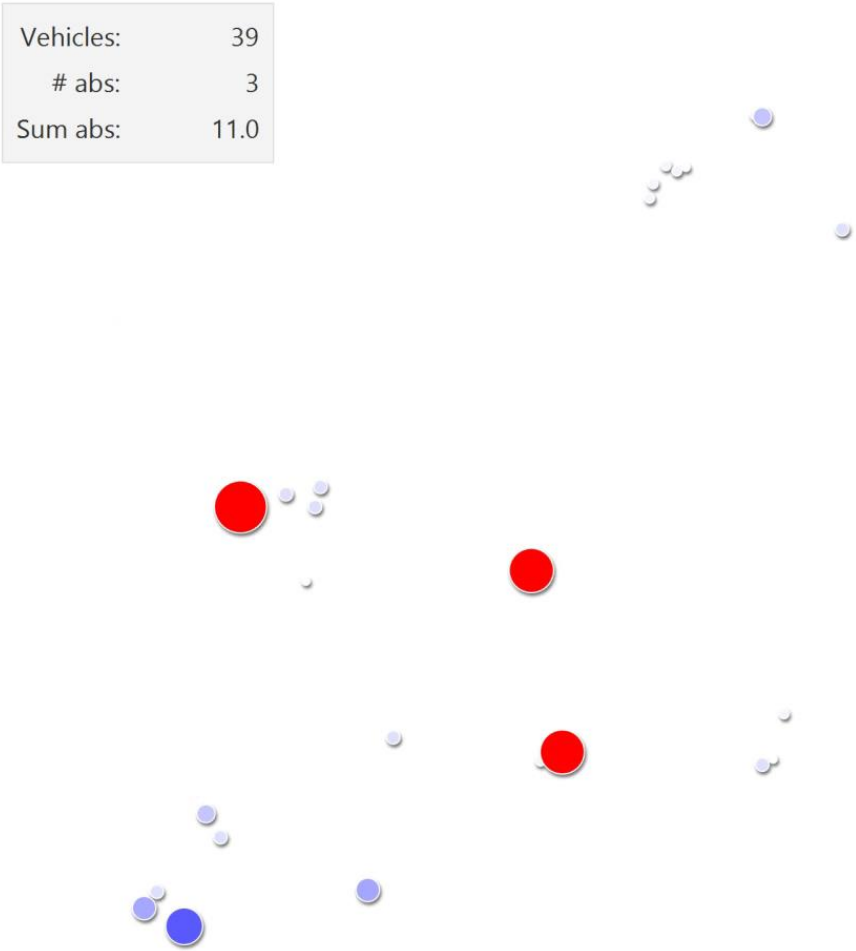
- Consider partial solutions
- Let A = set of absent customers

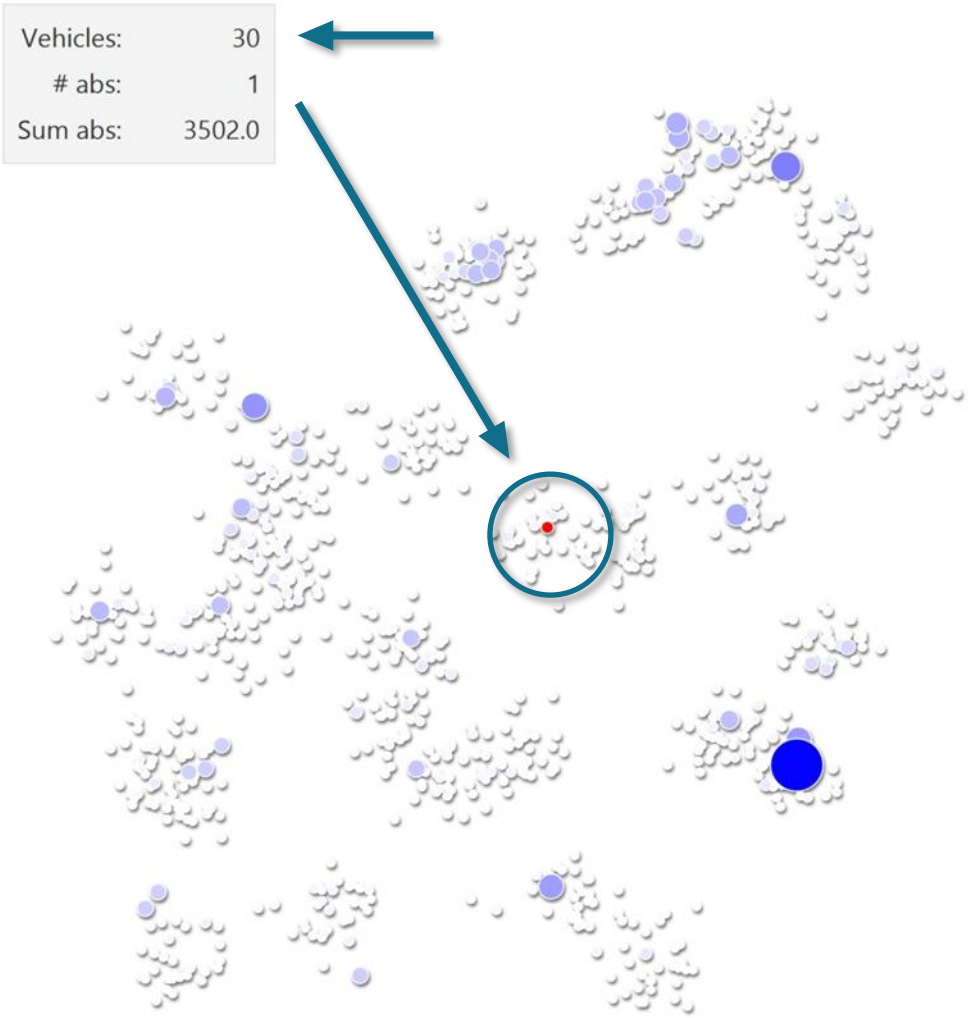
1. Apply SISRs
2. Accept s* if

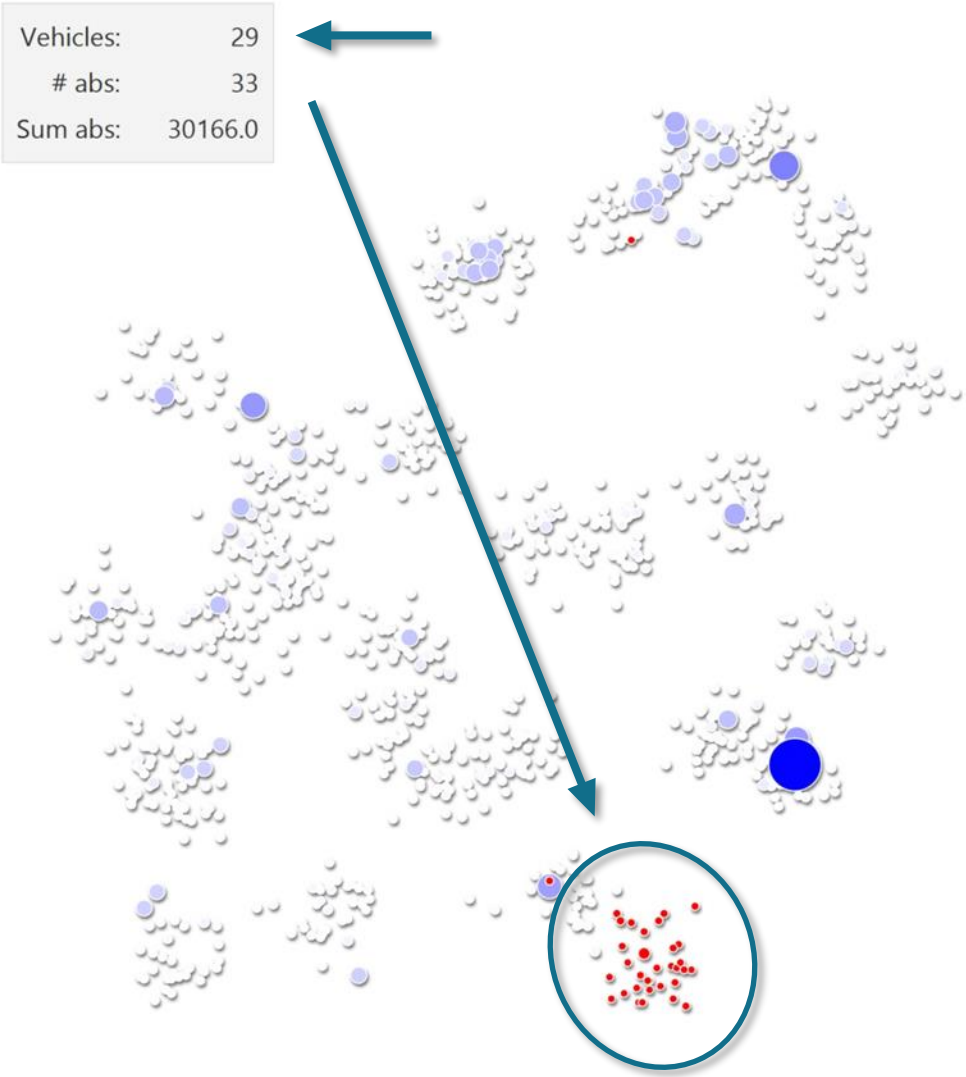
$$|A| \geq |A^*|$$

3. Repeat

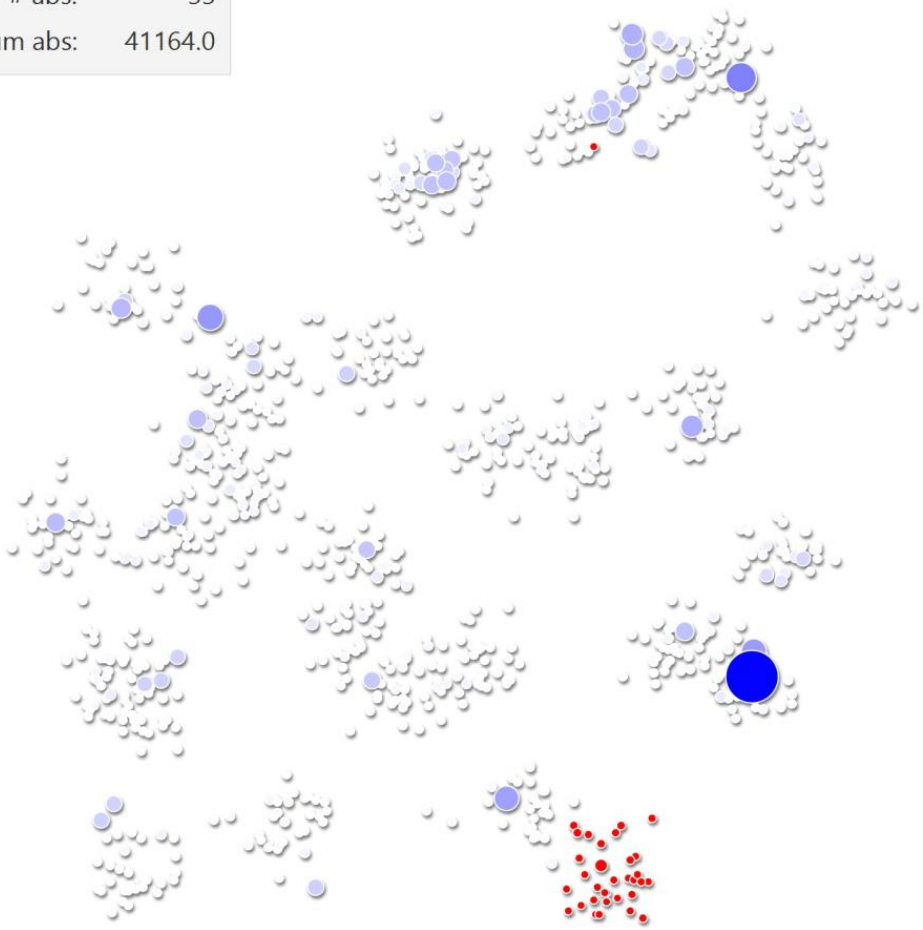

Vehicles: -
abs: -
Sum abs: -


Vehicles: 39
abs: 3
Sum abs: 11.0

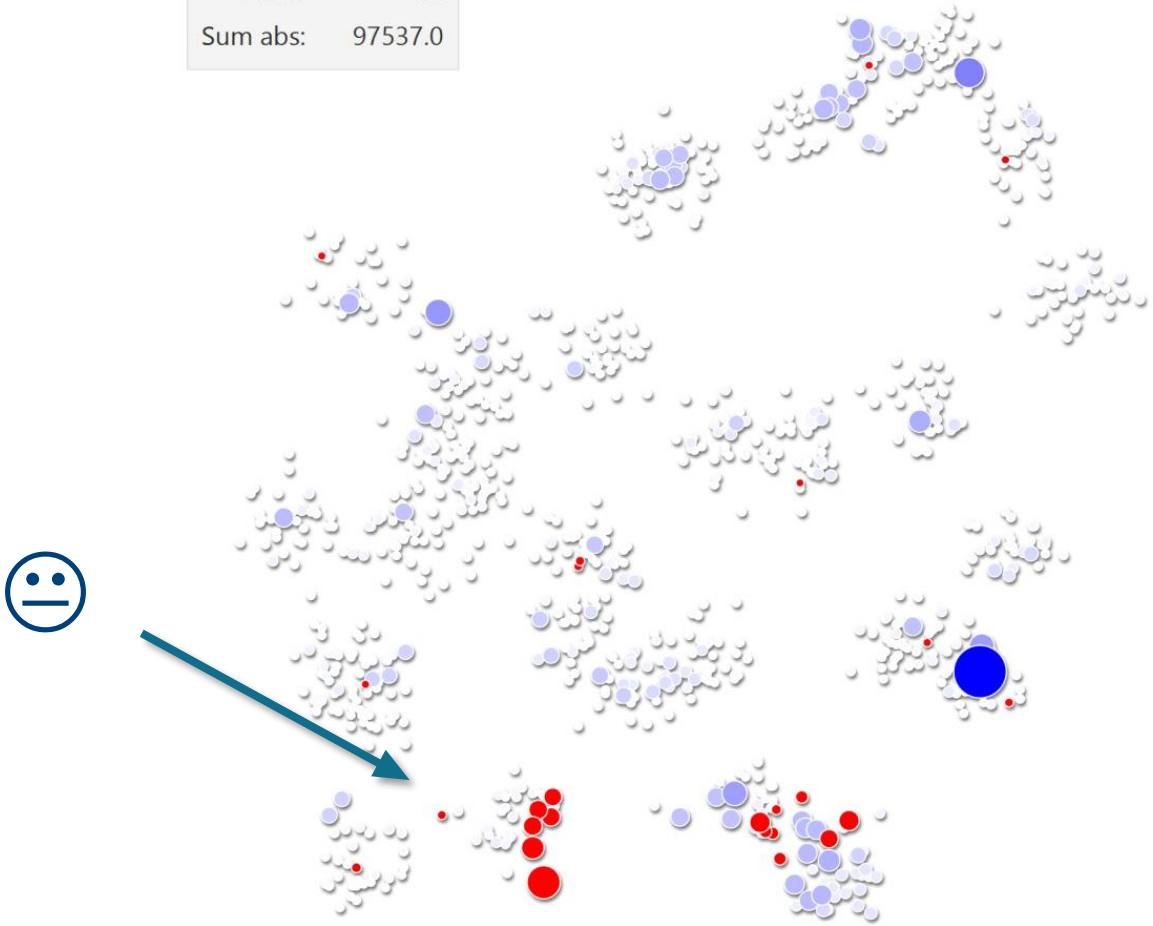

Vehicles: 30
abs: 1
Sum abs: 3502.0


Vehicles: 29
abs: 33
Sum abs: 30166.0

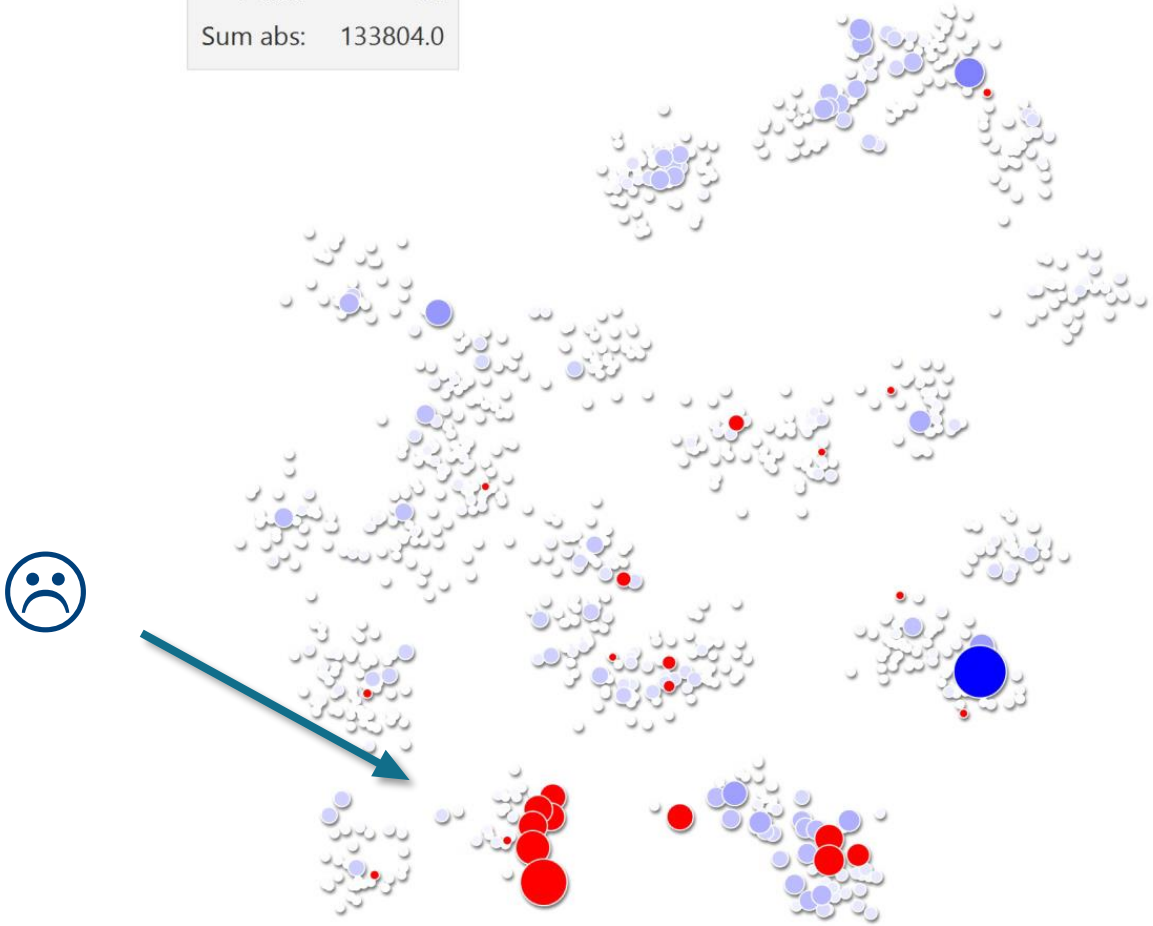
Vehicles:	29
# abs:	33
Sum abs:	41164.0



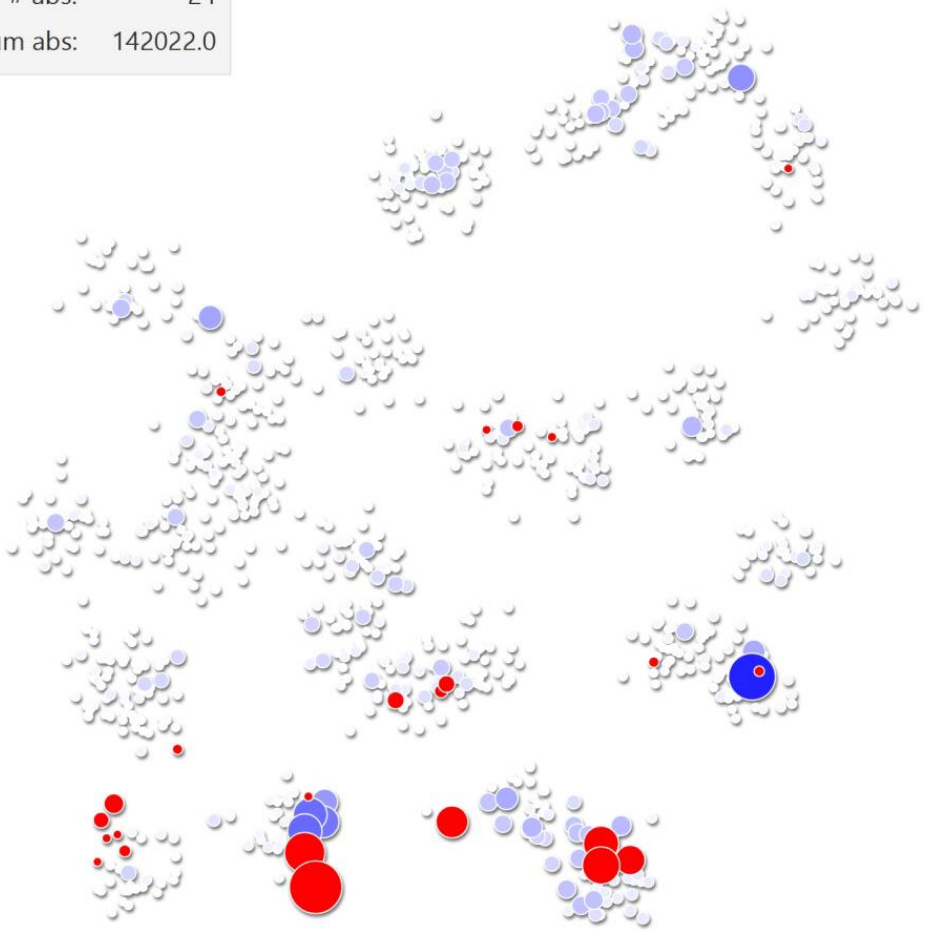
Vehicles:	29
# abs:	25
Sum abs:	97537.0



Vehicles:	29
# abs:	24
Sum abs:	133804.0

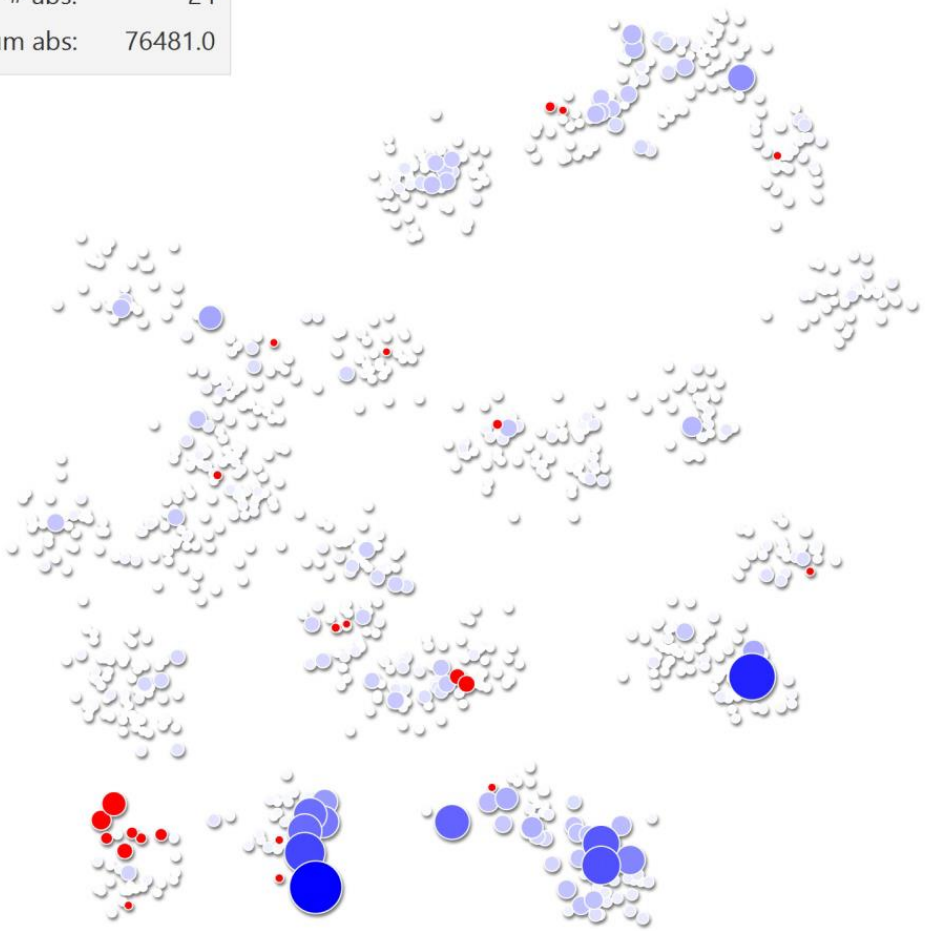


Vehicles:	29
# abs:	24
Sum abs:	142022.0

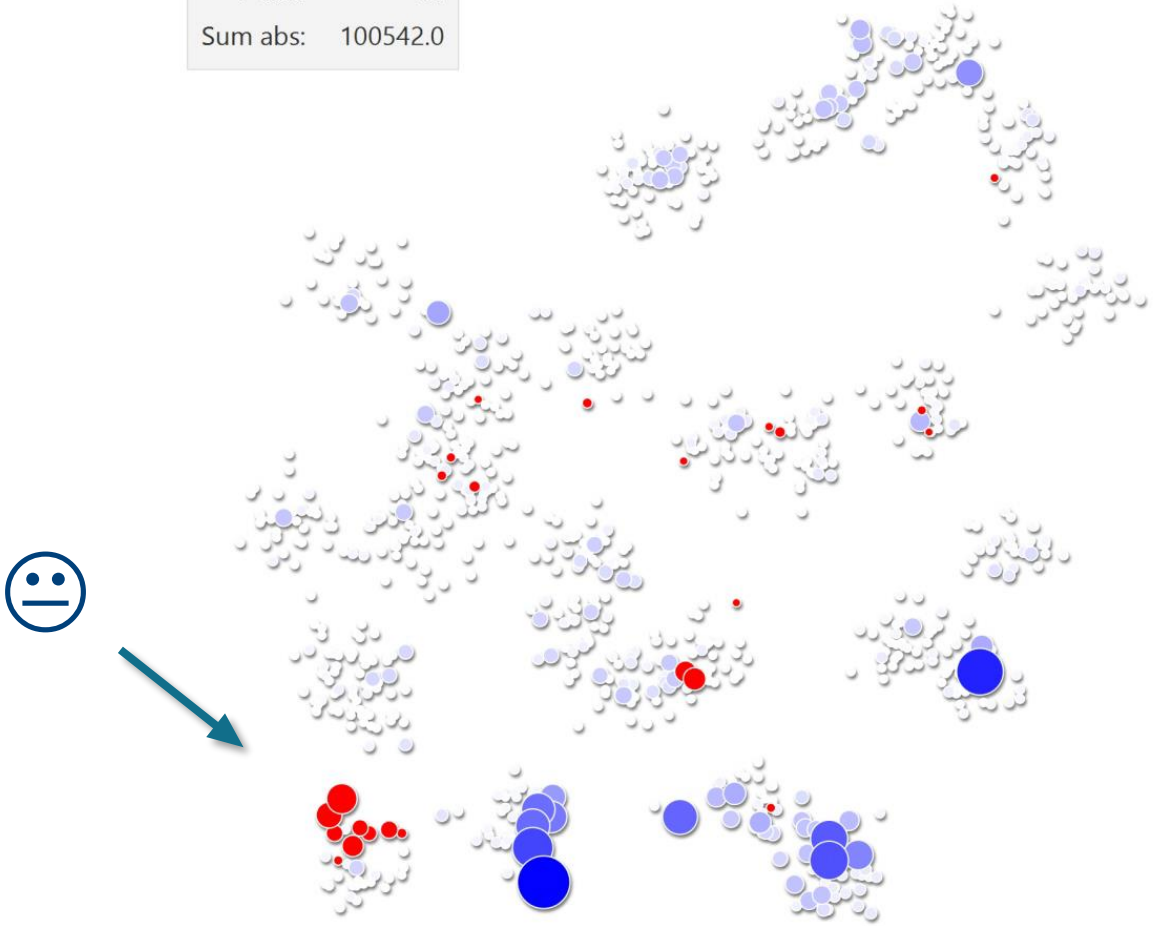


☺

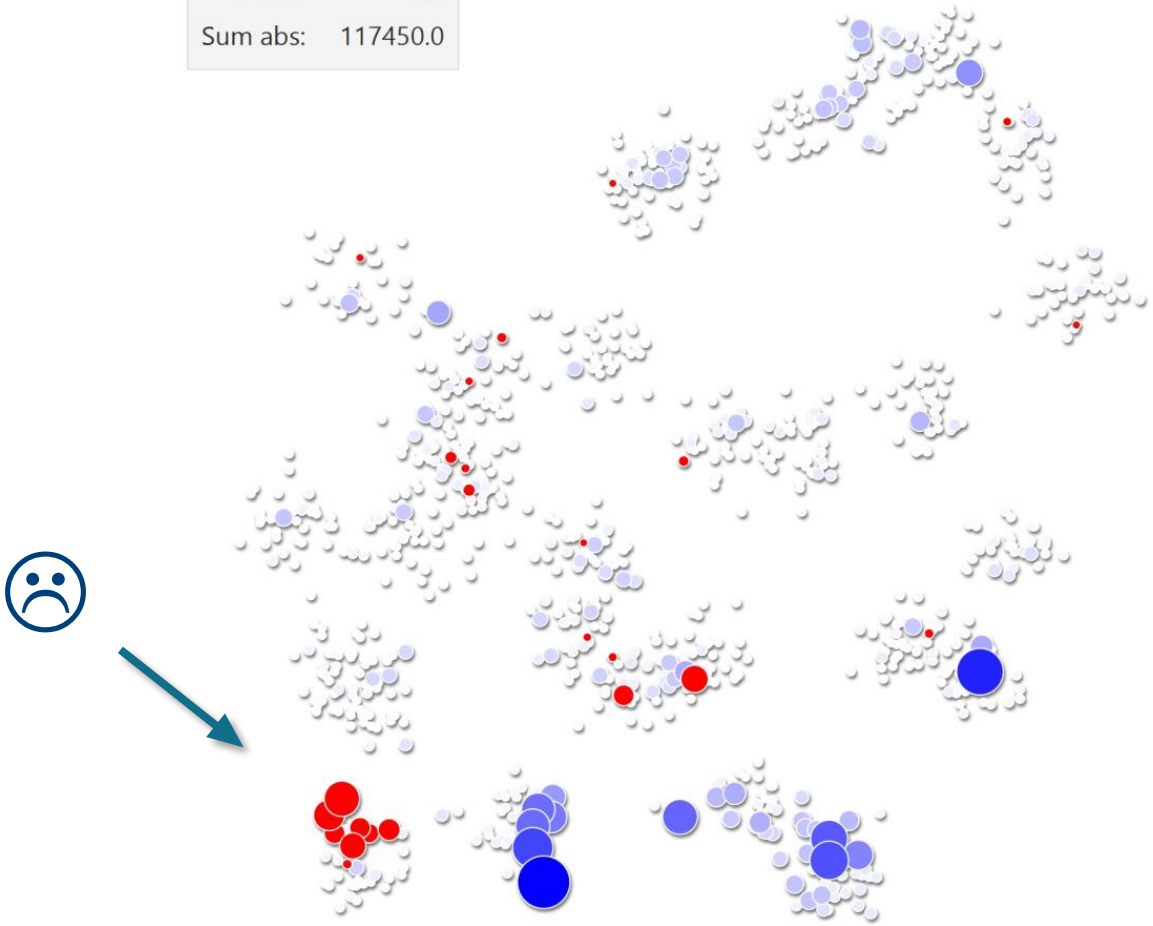
Vehicles:	29
# abs:	24
Sum abs:	76481.0



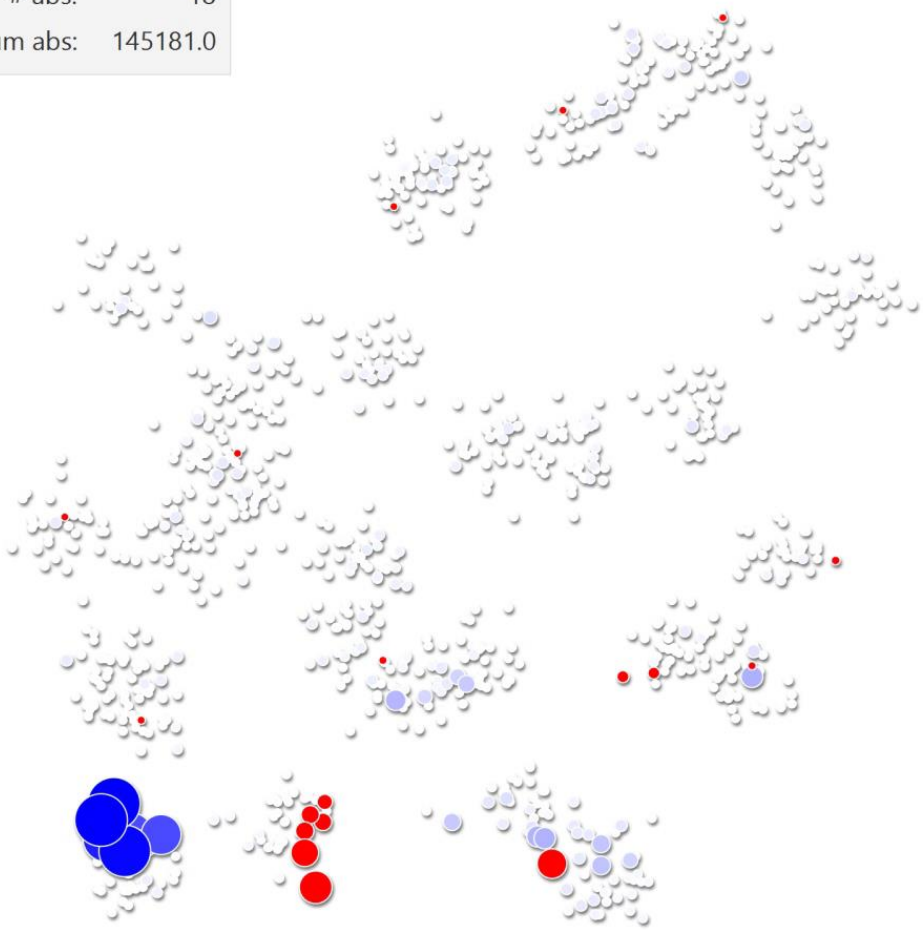
Vehicles:	29
# abs:	24
Sum abs:	100542.0



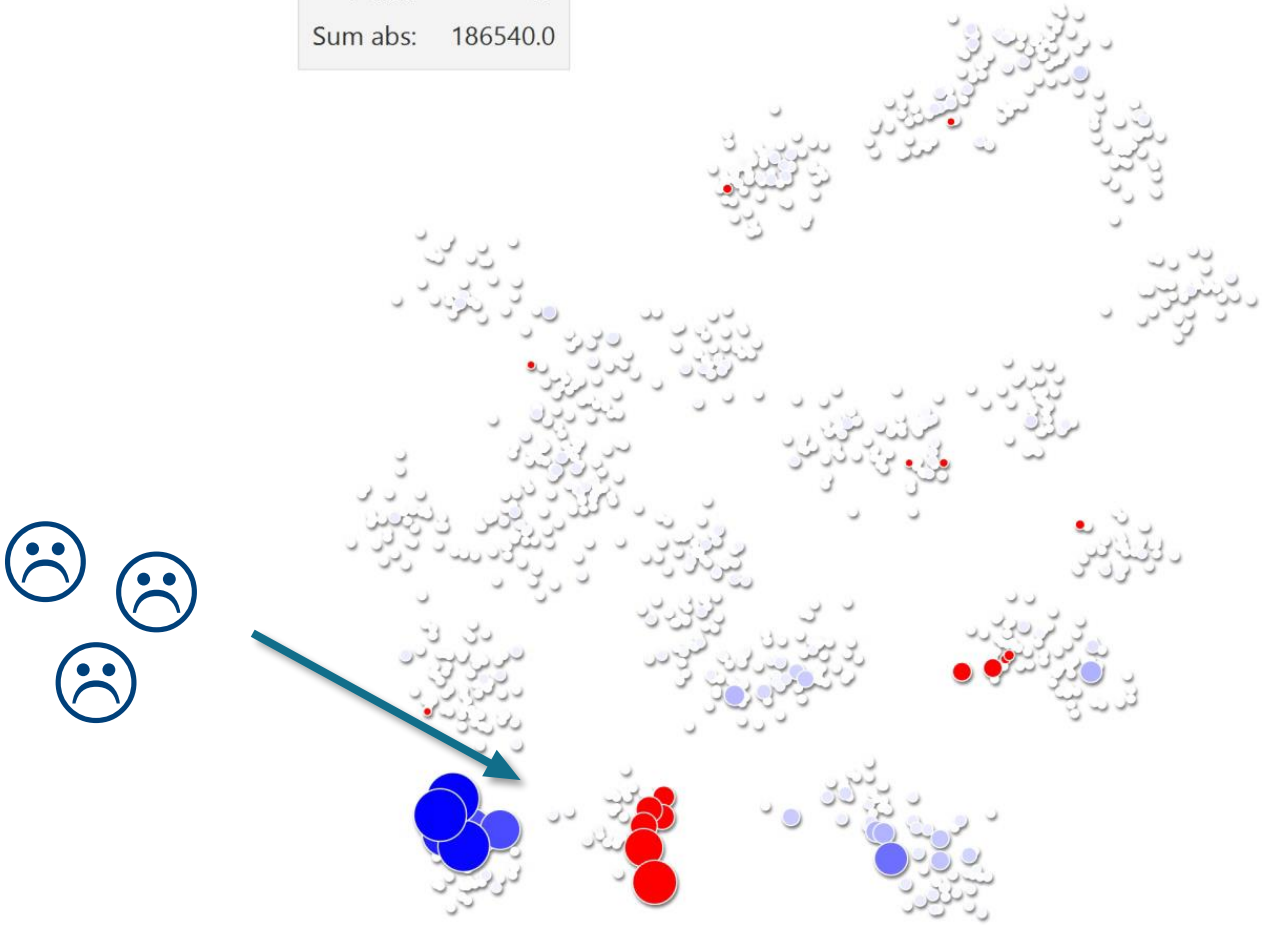
Vehicles:	29
# abs:	24
Sum abs:	117450.0



Vehicles:	29
# abs:	18
Sum abs:	145181.0



Vehicles:	29
# abs:	17
Sum abs:	186540.0



Absences-based acceptance criterion

Accept s* if

Absences-based acceptance criterion

Accept s* if

$$|A| > |A^*|$$

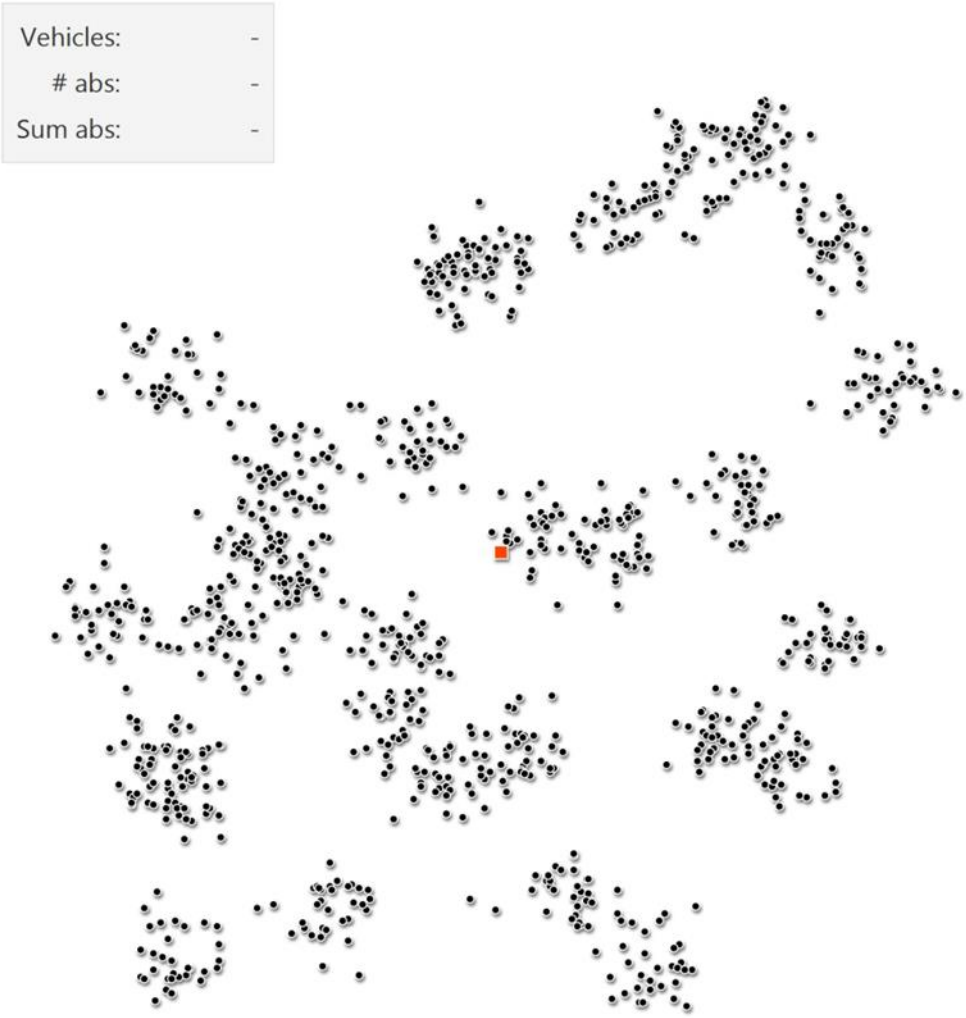
Absences-based acceptance criterion

Accept s* if

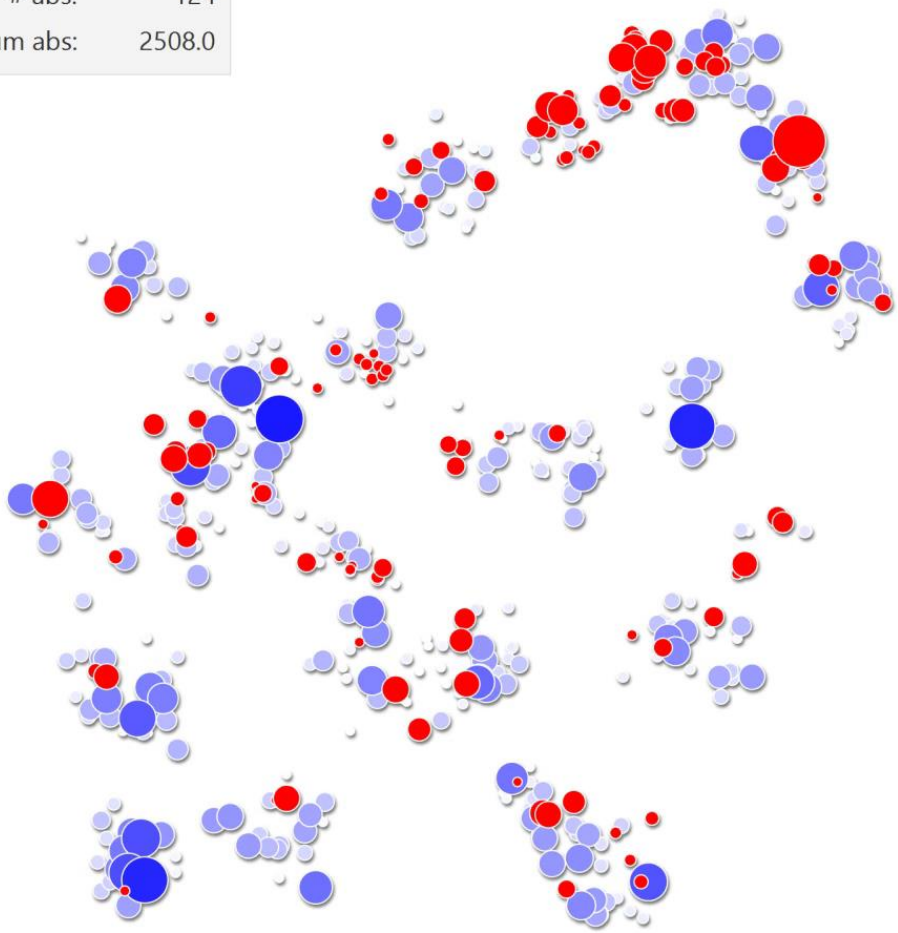
$$|A| > |A^*|$$

or if

$$\sum_{c \in A} abs_c > \sum_{c \in A^*} abs_c$$

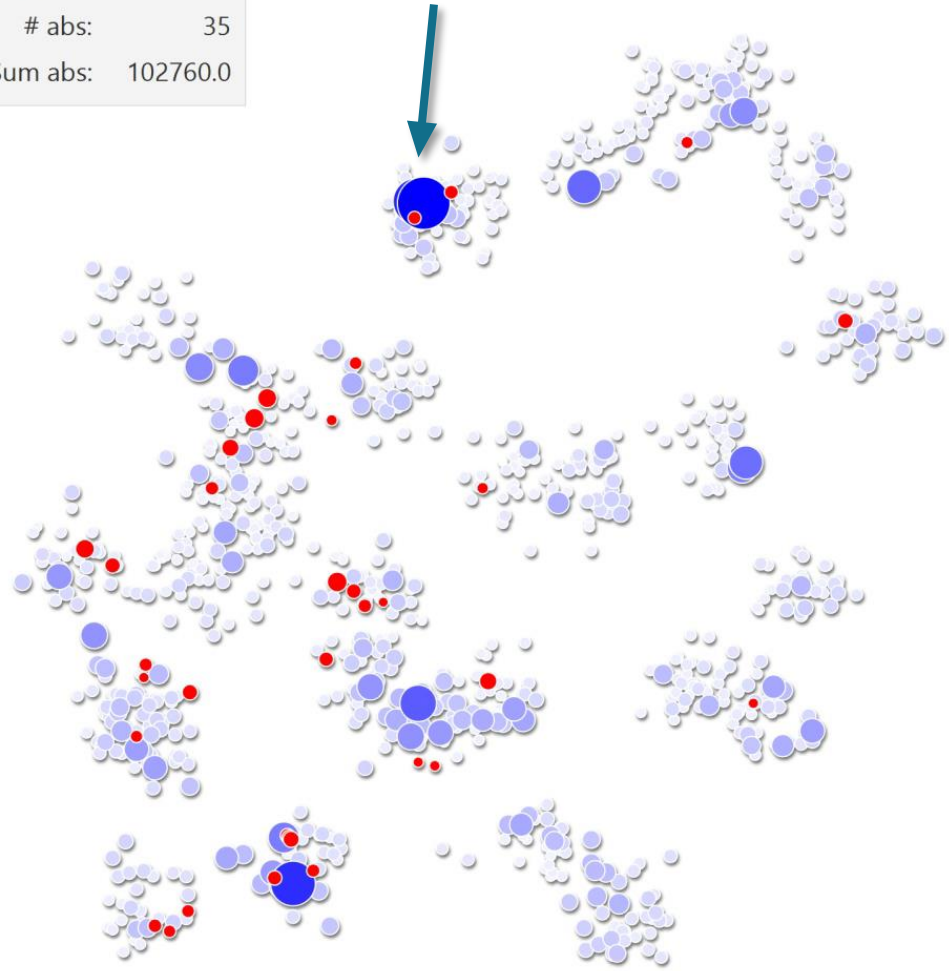

Vehicles: -
abs: -
Sum abs: -

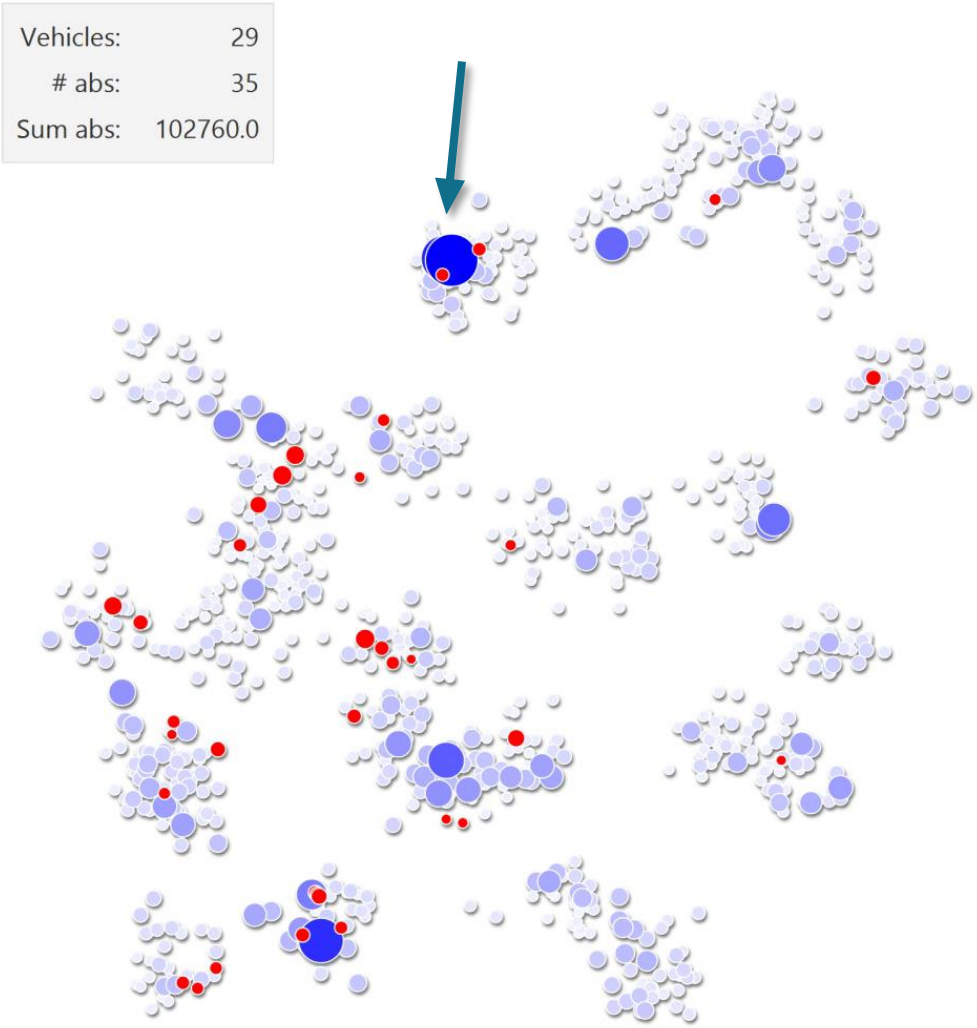
Vehicles:	39
# abs:	124
Sum abs:	2508.0



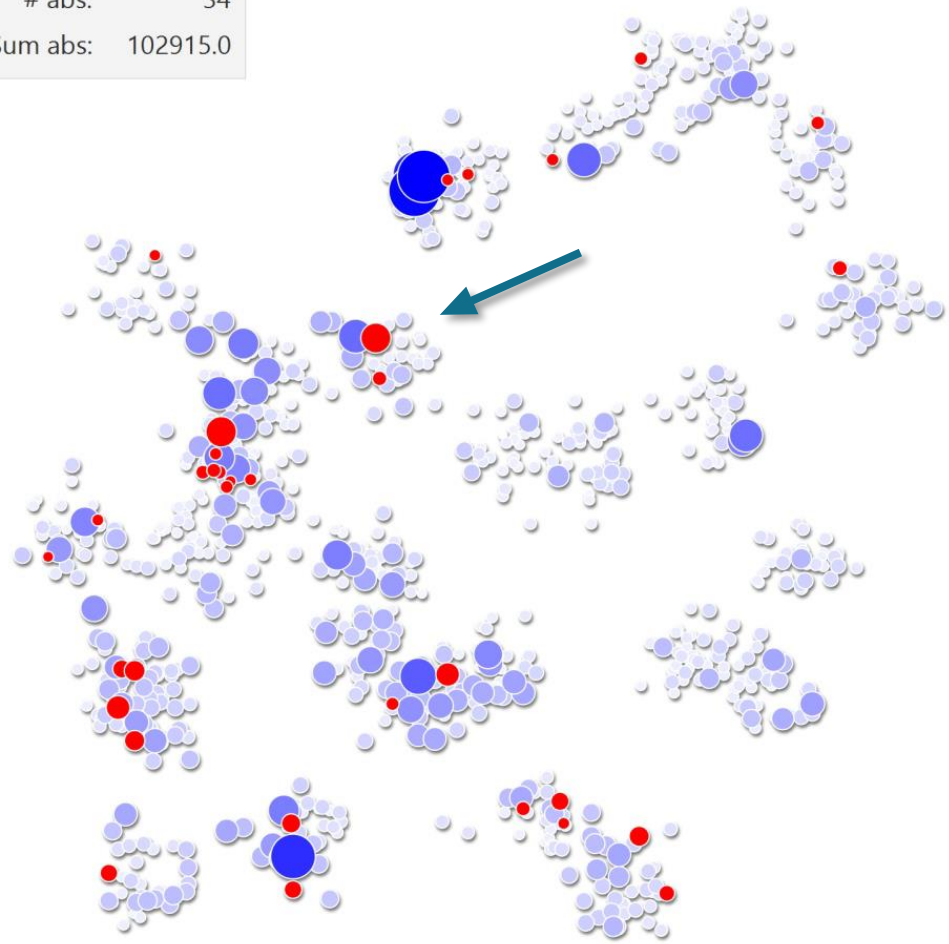
Skip to 29 vehicles

Vehicles:	29
# abs:	35
Sum abs:	102760.0

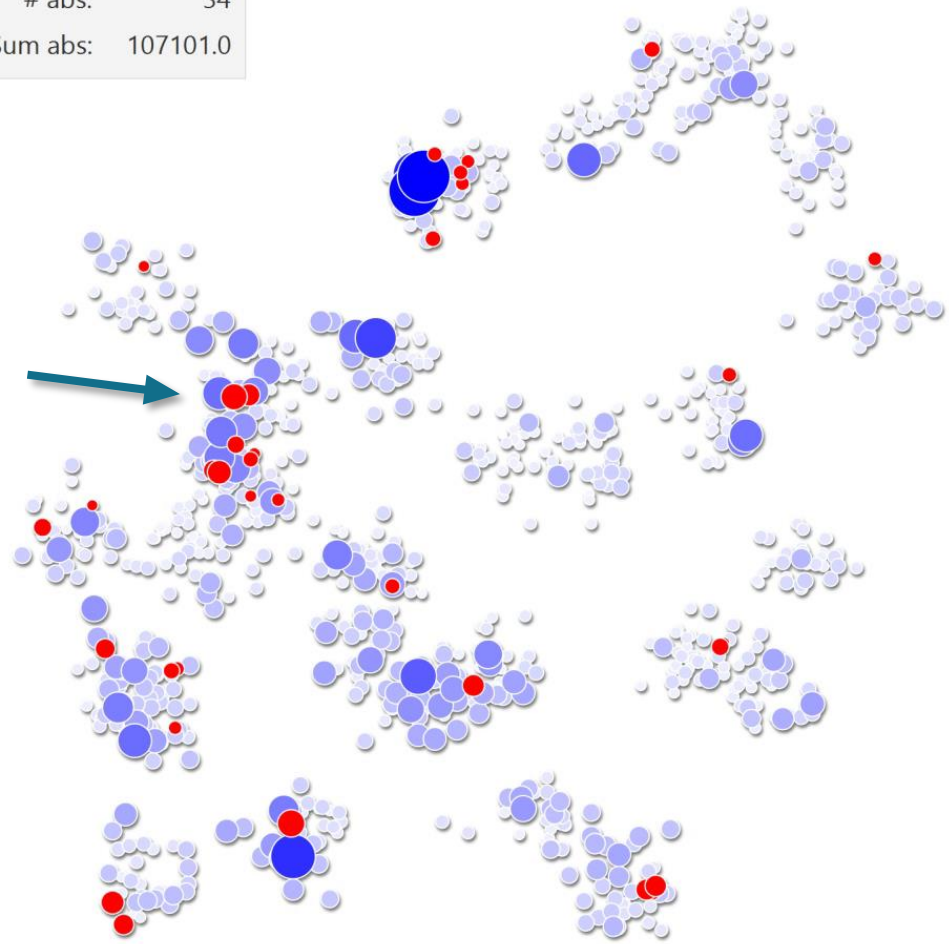



Vehicles: 29
abs: 35
Sum abs: 102760.0

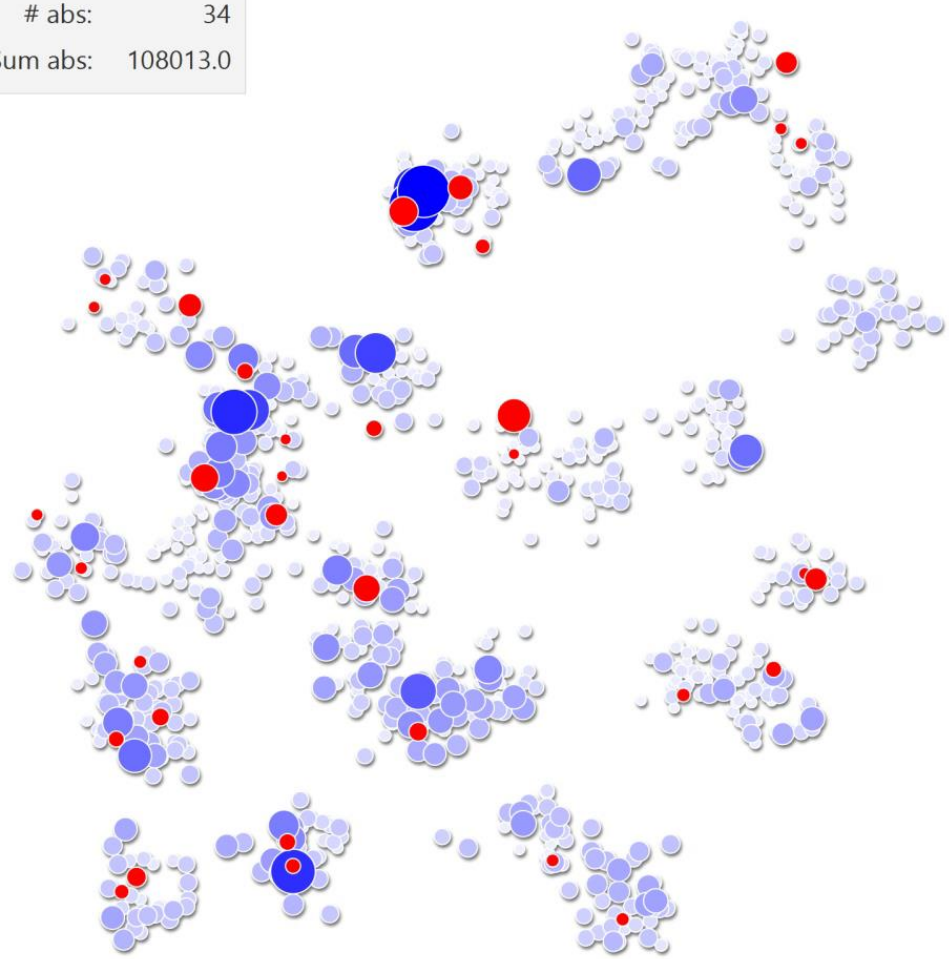
Vehicles:	29
# abs:	34
Sum abs:	102915.0



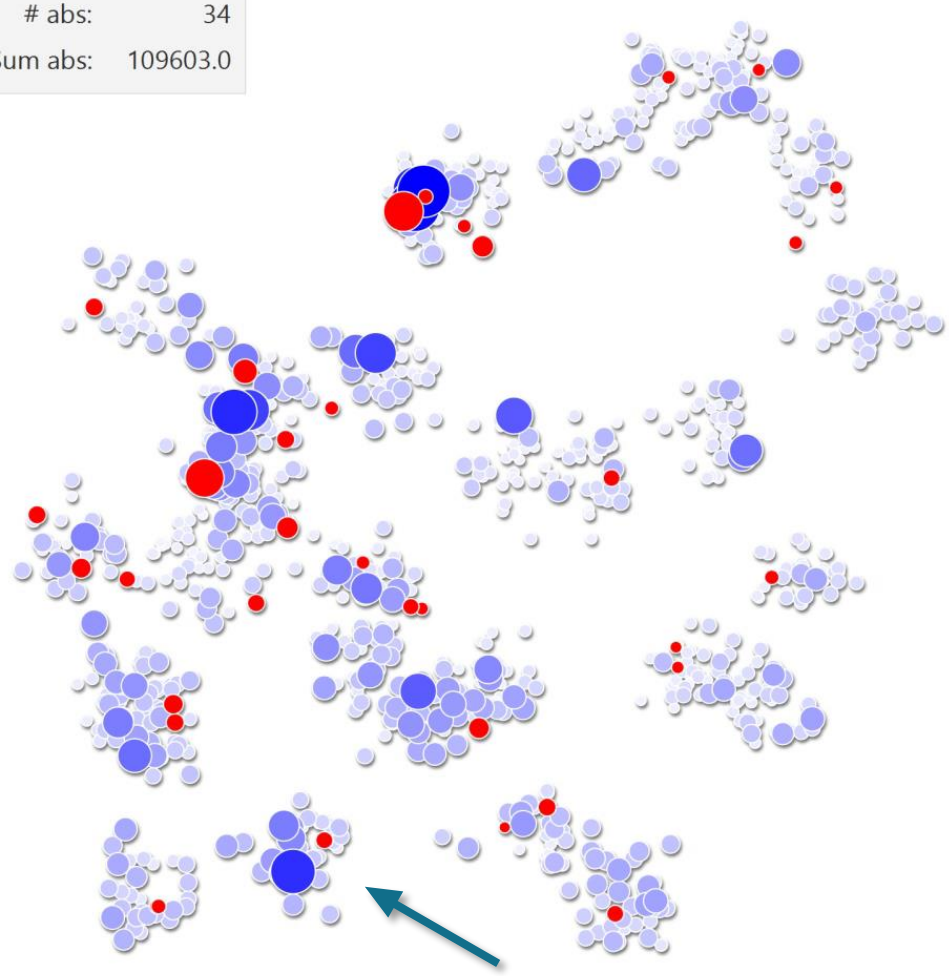
Vehicles:	29
# abs:	34
Sum abs:	107101.0



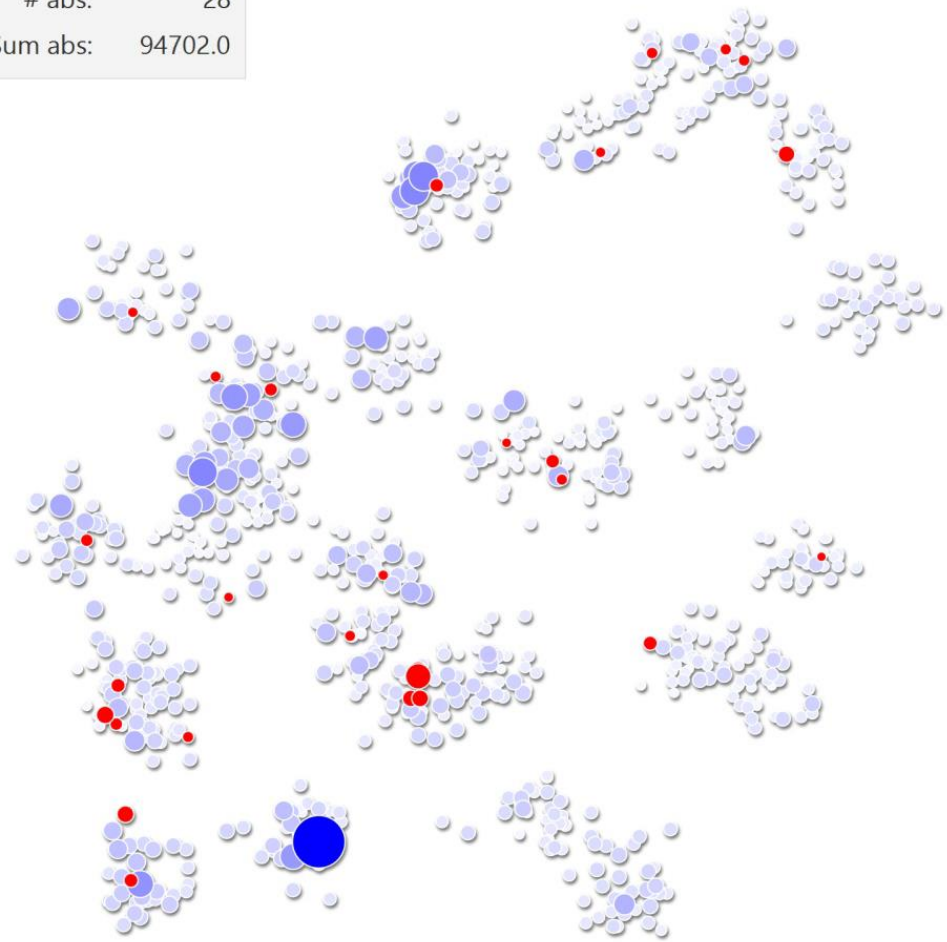
Vehicles:	29
# abs:	34
Sum abs:	108013.0



Vehicles:	29
# abs:	34
Sum abs:	109603.0

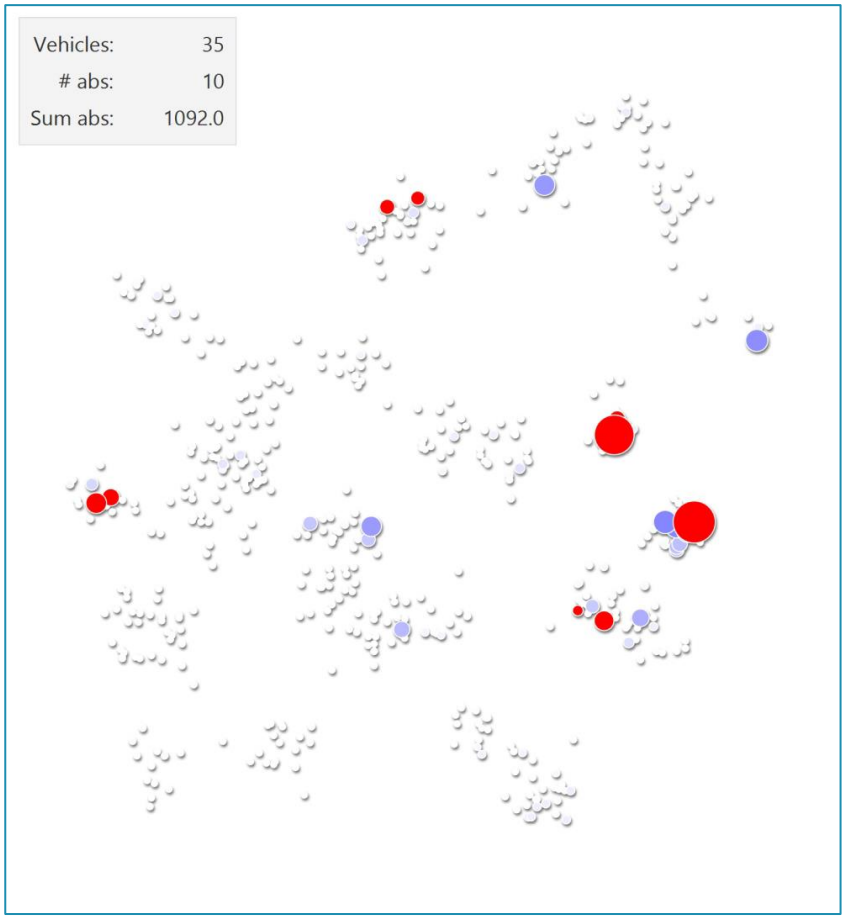


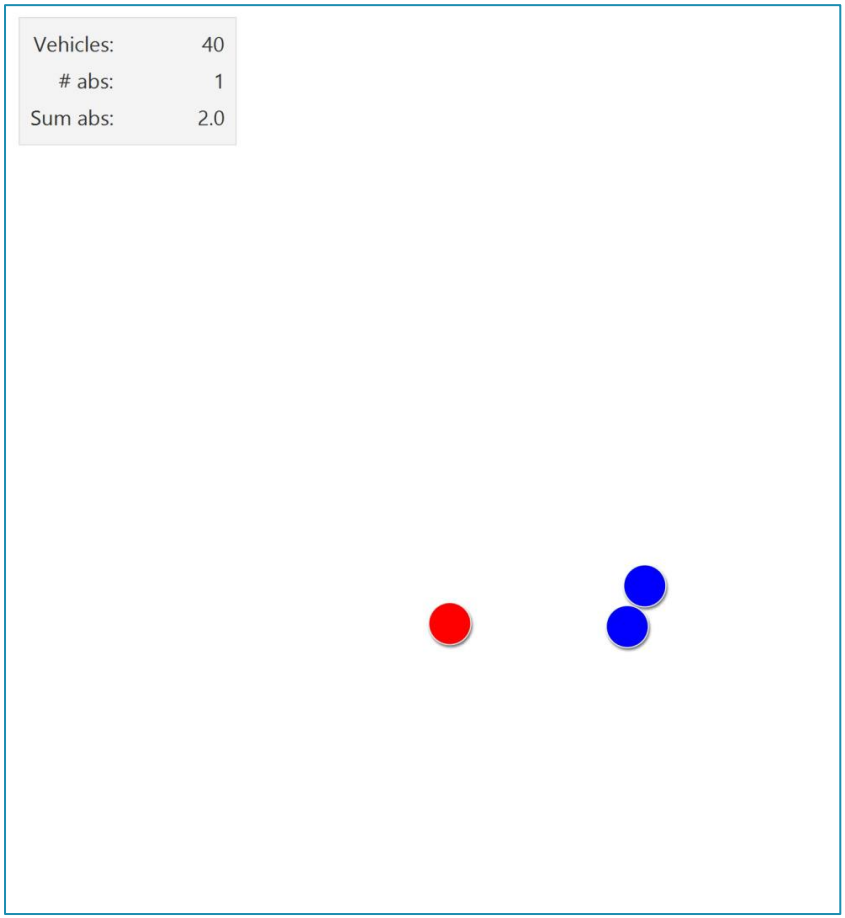
Vehicles:	29
# abs:	28
Sum abs:	94702.0



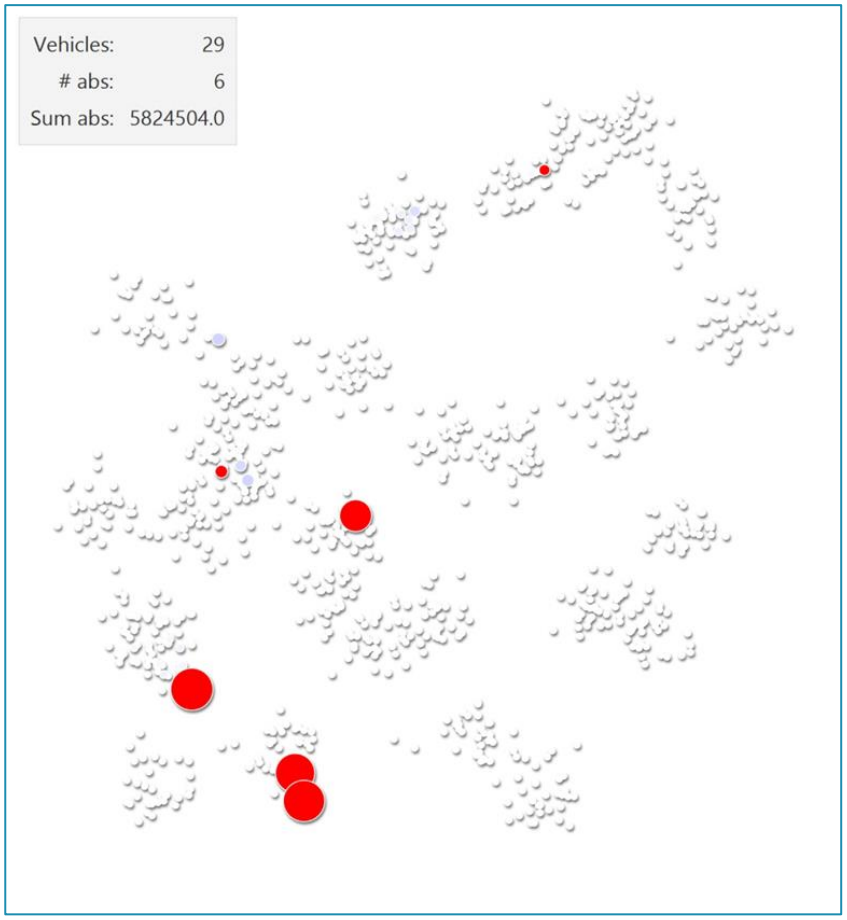
Acceptance criteria compared – Start

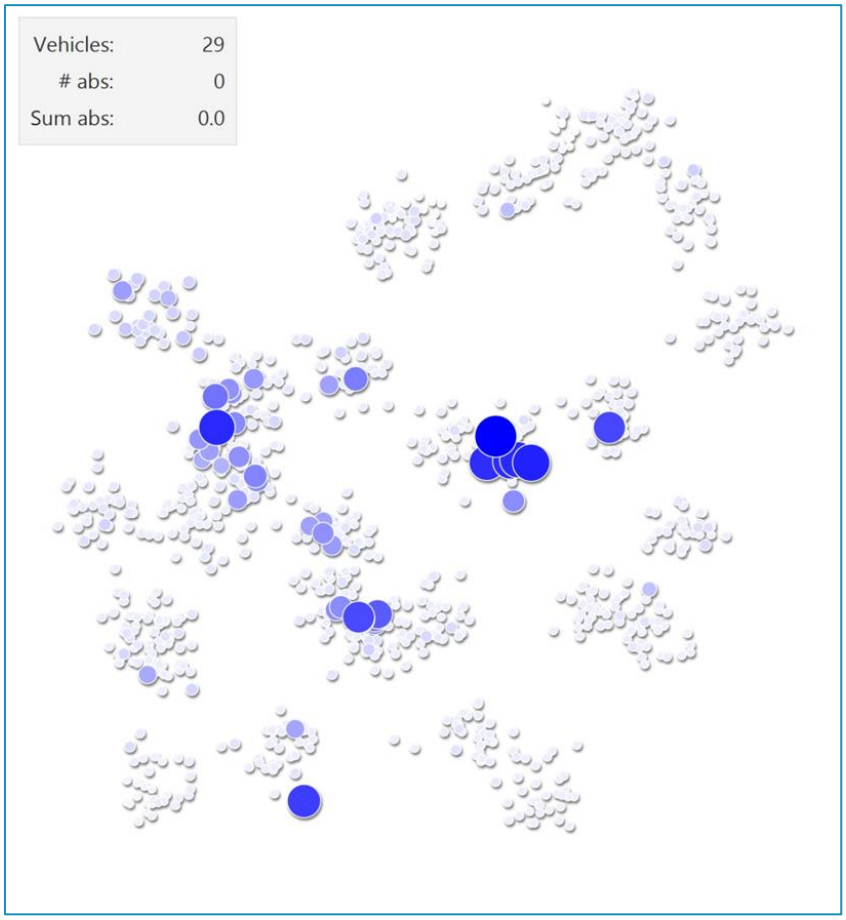
Acceptance criteria compared – Start

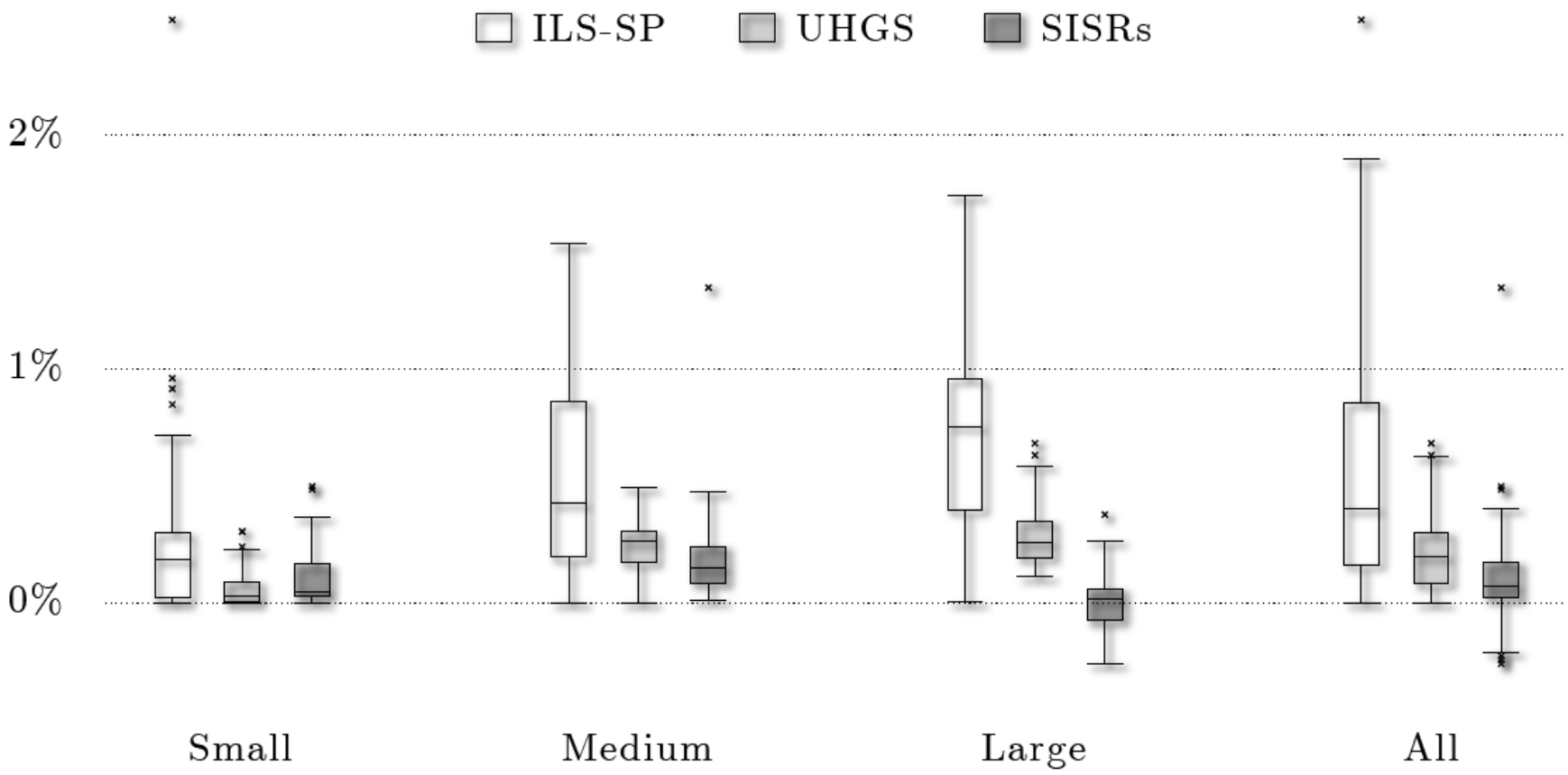




Acceptance criteria compared – Final solutions








DHL


ON STRIKE
FIRST STUDENT
TEAMSTERS LOCAL 174
UNFAIR
SCHOOL BUS
STOP

Jan Christiaens, Greet Vanden Berghe: **Slack Induction by String Removals for Vehicle Routing Problems,** Transportation Science, 54 (2), 299-564, 2020

KU LEUVEN