# New Benchmark Instances for the Capacitated Vehicle Routing Problem


Eduardo Uchoa - UFF

Diego Pecin – PUC-Rio

Artur Pessoa - UFF

Marcus Poggi – PUC-Rio

Anand Subramanian - UFPB

Thibaut Vidal – MIT, PUC-Rio

# Vehicle Routing

One of the most widely studied families of problems in Combinatorial Optimization:

- Direct applications in the real world systems that distribute goods and services, vital to the modern economies.

Reflecting the variety of aspects present in those systems, the VRP literature is spread into dozens of variants.

# Vehicle Routing Variants

For examples, there are variants that consider:

- Time windows,
- Multiple depots,
- Mixed vehicle fleet,
- Split delivery, pickup and delivery,
- Loading constraints, etc.

“Rich” variants try to consider as many of those aspects as possible

# Capacitated Vehicle Routing Problem (CVRP)

VRP “poorest” variant:

- single depot
- homogeneous fleet,
- only deliveries (or only pickups)
- customer demands measured by a one-dimensional number (weight, volume, number of pax, etc)

Only route restriction: the sum of the demands must not exceed the vehicles capacity *Q*.

# Capacitated Vehicle Routing Problem (CVRP)

In the VRP landscape, CVRP occupies a central position:

- By historical reasons, it was defined in 1959 by Dantzig (one of our Founding Fathers) and Ramser
- By practical reasons, there is a significant number of real world systems that can be satisfactorily modeled as a CVRP

# Capacitated Vehicle Routing Problem (CVRP)

CVRP is a natural testbed for new ideas:

- Its simplicity allows cleaner descriptions and implementations, without the additional conceptual burden of more complex variants

# Capacitated Vehicle Routing Problem (CVRP)

CVRP is a natural testbed for new ideas:

- Several good ideas were first proposed for the CVRP and then successfully generalized for many other variants

Example: Clarke and Wright heuristic (1964)

# Our Claim

Recent advances in both exact and heuristic methods made existing CVRP benchmark instances obsolete or limited

# Our Contribution

A new set of 100 benchmark instances, ranging from 100 to 1000 customers:

- We tried to make this set as diversified as possible, covering a wide range of characteristics found in real applications.

# Existing benchmark instances

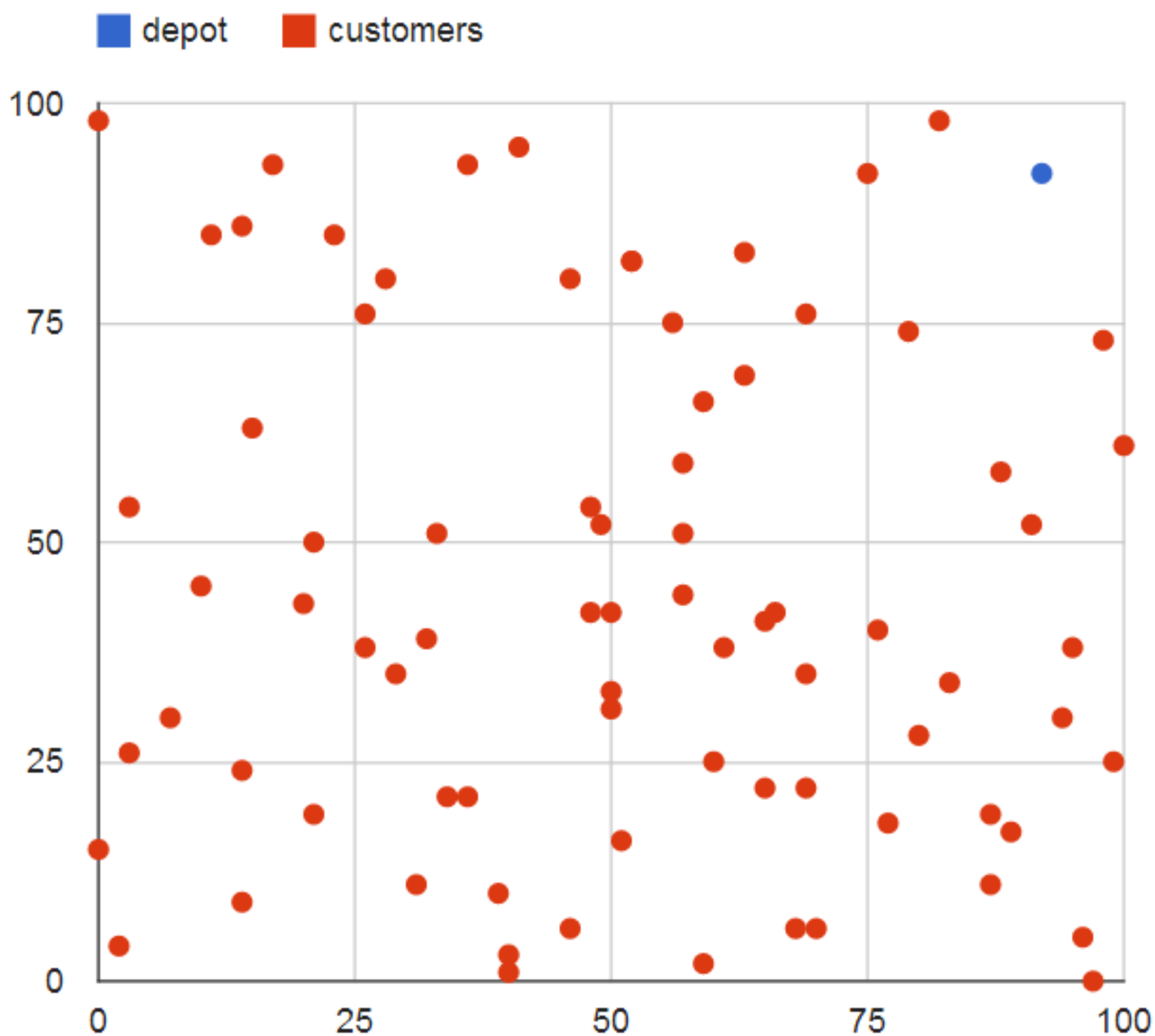
Since 1995, the **exact methods** have been tested mostly in benchmarks from series A, B, E, F, M and P.

# Series A (Augerat, 1995)

- 27 instances with *n* (number of customers) from 30 to 79, *Q*=100, demands from the U[1,30], 10% of them are multiplied by 3
- Random depot and customer positioning in the grid [1,100]x[1,100], euclidean distances

**Very easy** for modern algorithms:

- Average solution time (BCP-PPPU14): 4.5s
- Good heuristics find all optima



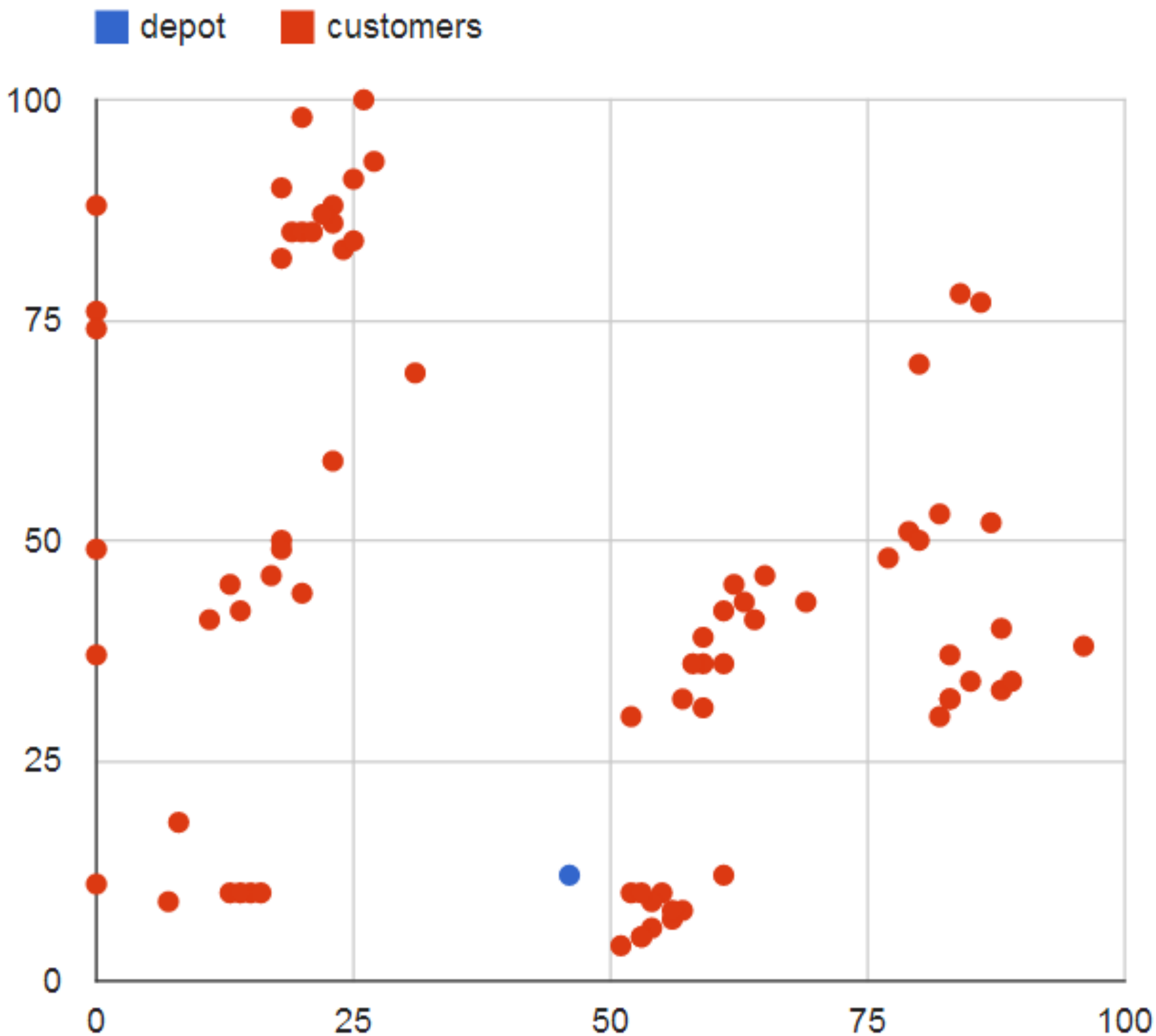

A-n80-k10
(n=79,
Q=100)

# Series B (Augerat, 1995)

- 23 instances, *n* from 30 to 77, *Q*=100, demands from U[1,30], 10% of them are multiplied by 3, random depot, clustered customers

**Very easy** for modern algorithms:

- Average solution time (BCP-PPPU14): 5.9s
- Good heuristics find all optima




B-n78-k10
(n=77,
Q=100)

# Series P (Augerat, 1995)

- 24 instances adapted from previous instances by other authors, $n$ from 15 to 100, $Q$ from 35 to 3000.

**Easy** for modern algorithms:

- Average solution time (BCP-PPPU14): 32.7s
- The best heuristics find all optima

# Series E (Christofides and Eilon, 1969, some modified by Gillet and Miller, 1974)

- The most classical benchmark set
- 13 instances with *n* between 12 and 100

**Easy** for modern algorithms:

- Average solution time (BCP-PPPU14): 29.2s
- The best heuristics find all optima

# Series F (Fisher, 1994)

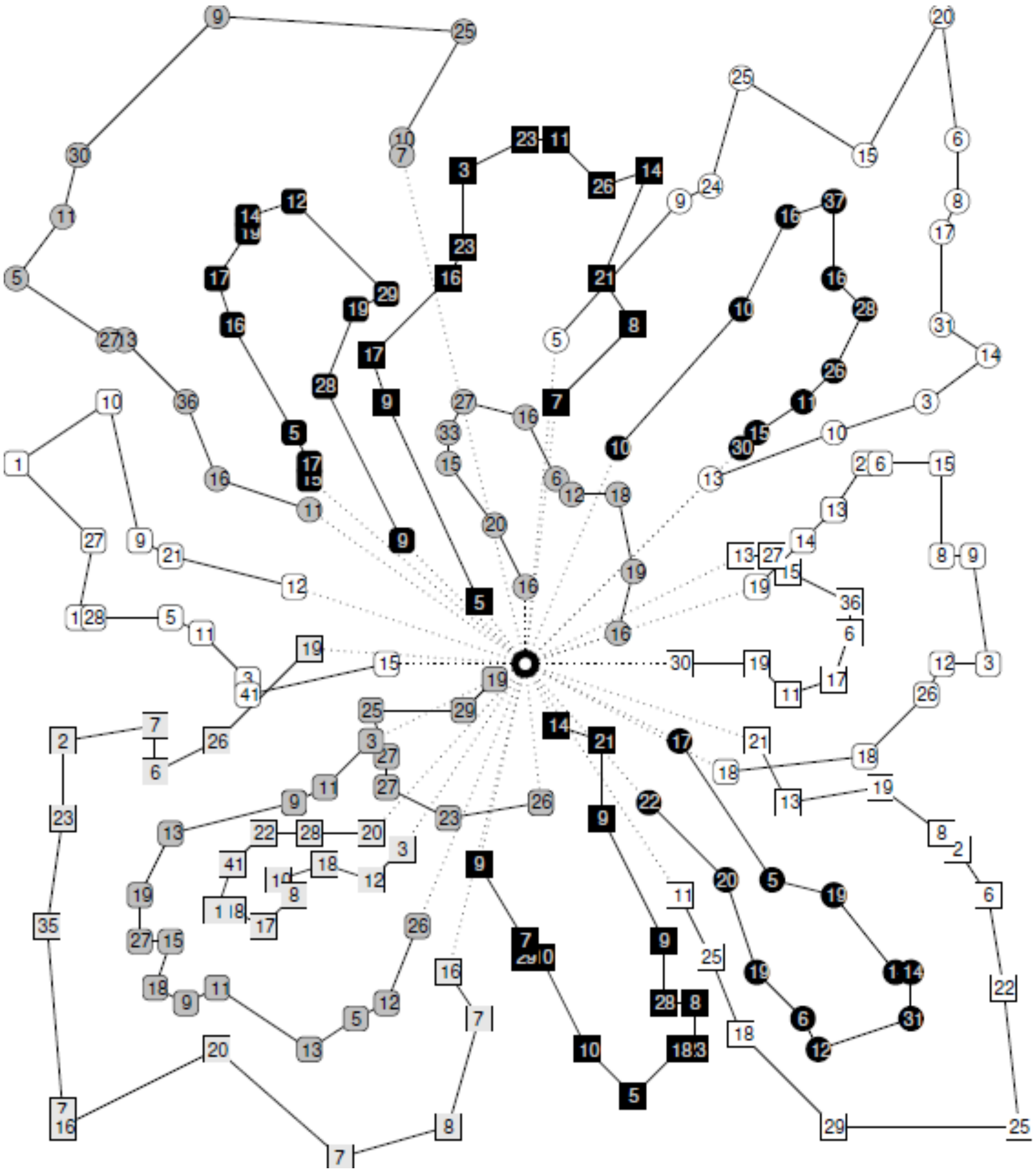
- Only 3 instances (F-n45-k4, F-n75-k4, F-n135-k7).
- Large Q, small demands -> very long routes -> BC performs better than BCP!

**Easy** for (some) modern algorithms:

- Avg. time BC-LLE04: 53.1s (modern machine)
- Avg. Time BCP-PPPU14: 3679s
- Good heuristics find all optima

# Series M (Christofides, Mingozzi and Toth, 1979)

- 5 instances with *n* between 100 and 199.
- Instances M-n151-k12, M-n200-k16 and M-n200-k17 remained very challenging until recently
- M-n151-k12 was solved in 2012 by Contardo and by Ropke
- The last two instances were solved last year (PPPU14)
- M-n200-k16 is very hard for heuristics



M-n200-k16
(n=199,
Q=200)

Optimal: 1274

# Existing benchmark instances

The **heuristic methods** have been tested mostly in other benchmarks.

# CMT instances

Very classical, they are the same 5 instances in series M plus 2 instances from series E

- However, the distances are not rounded
- As happens in other benchmark sets used by the heuristic community, it also contains 7 instances with distance/duration constraints (DCVRP)

# CMT instances

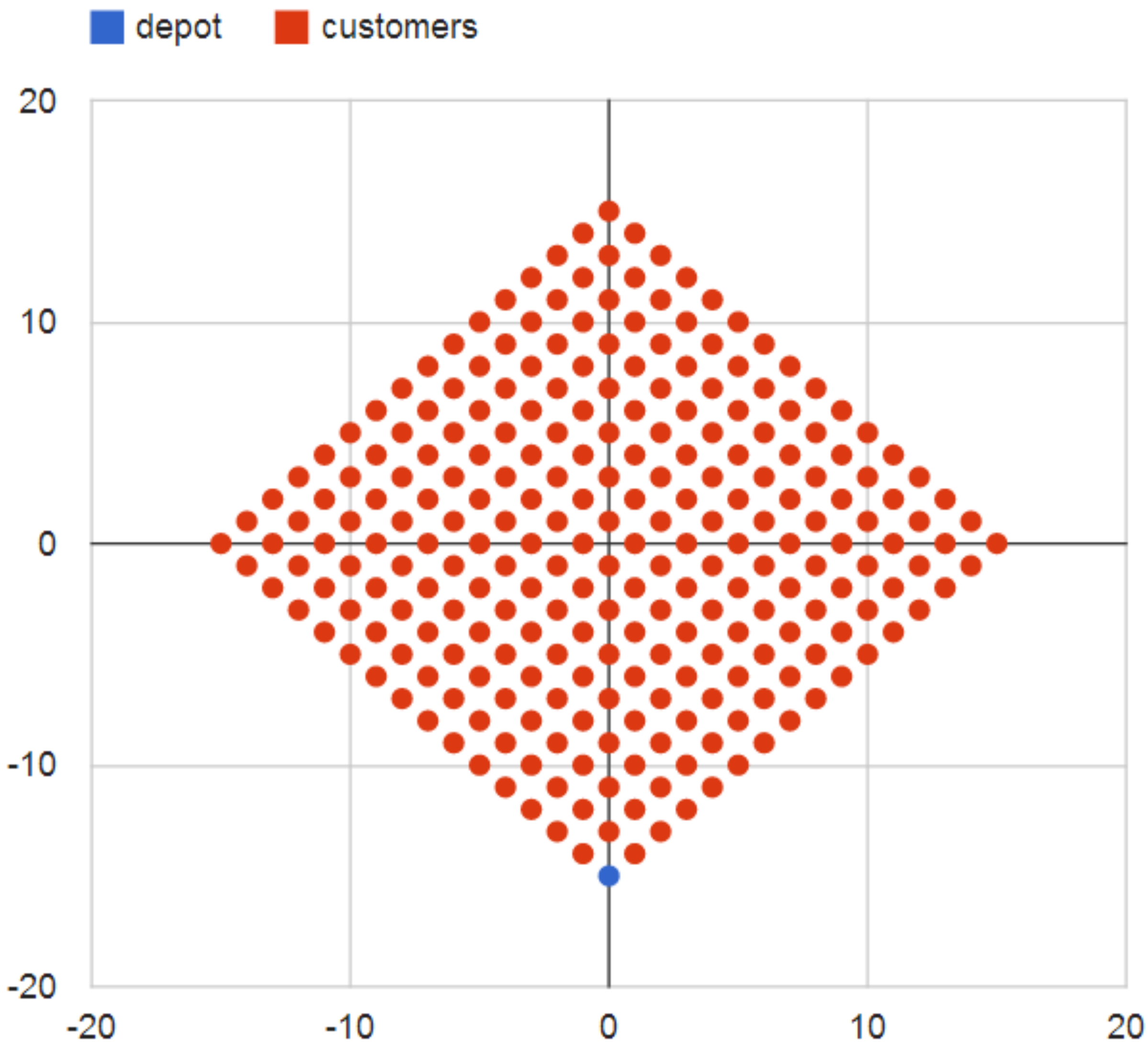
Until recently, no one took the trouble of running a modern exact method on them!

- Avg. solution time (BCP-PPPU14): 2816s
- 6 solutions, now known to be optimal, were already found in Taillard (1995). Modern heuristics find them easily
- Instance CMT5 (akin to M-n200) is very hard to heuristics. Only Mester and Bräysy (2007) had already published the optimal value of 1291.29

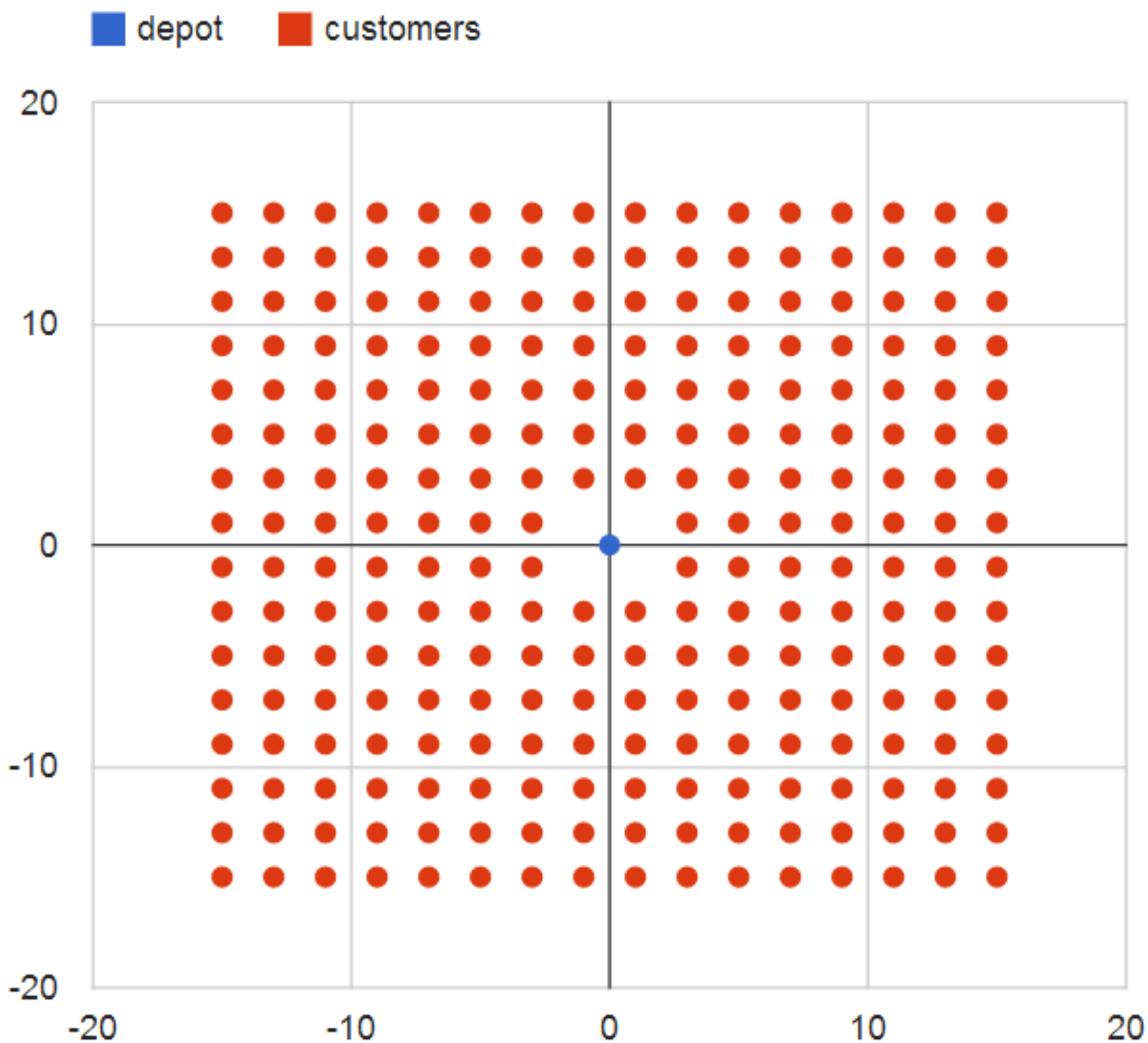
# Golden, Wasil, Kelly and Chao Instances (1998)

From 240 to 483 customers, appear frequently in recent literature

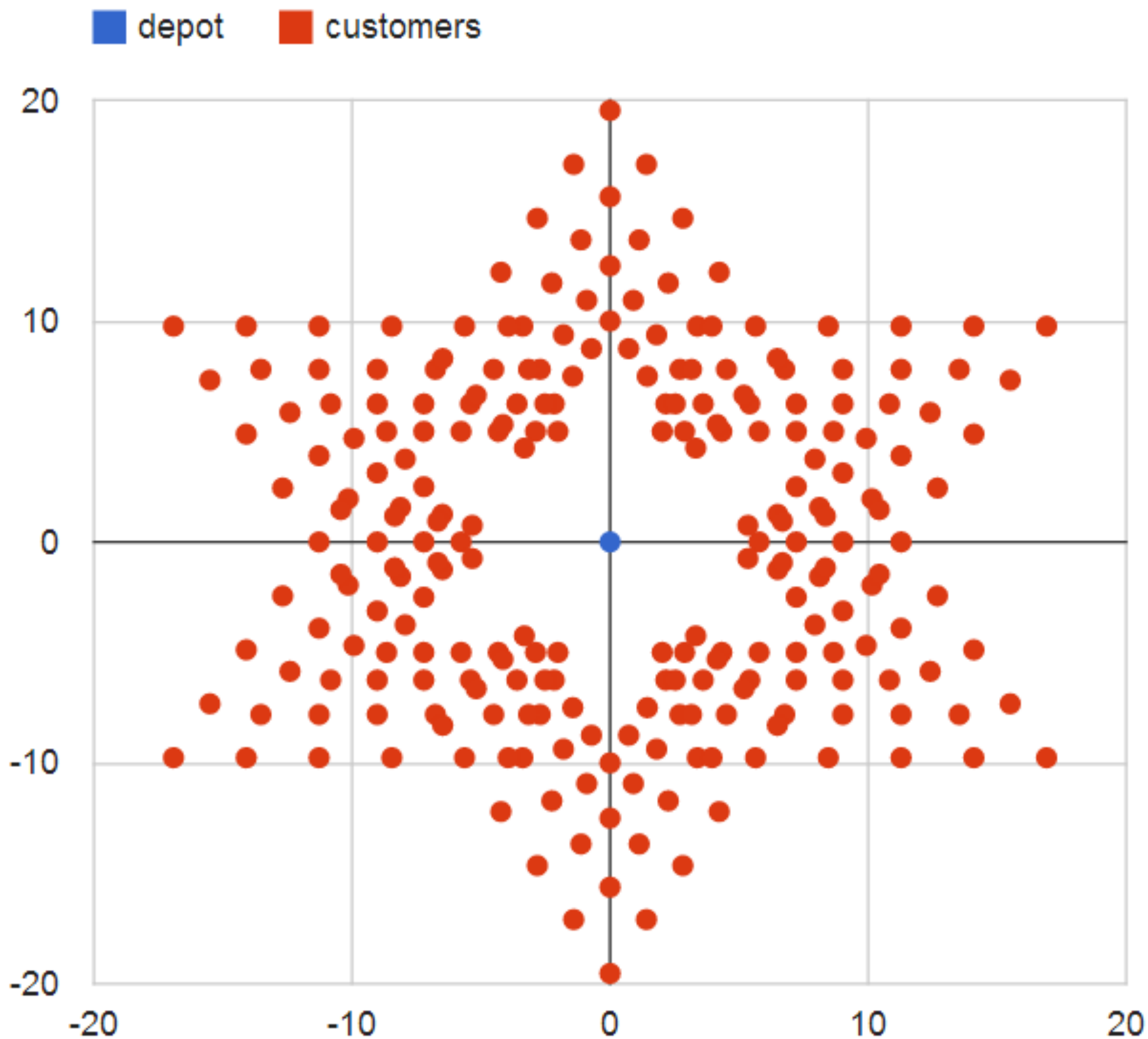
- 12 instances:
  - Concentric squares, depot in the corner
  - Concentric squares, depot in the center
  - Concentric 6 pointed stars, depot in the center
- There are also 8 DCVRP instances:
  - concentric circles, depot in the center




Golden_9
(n=255,
Q=1000)




Golden_13
(n=255,
Q=1000)

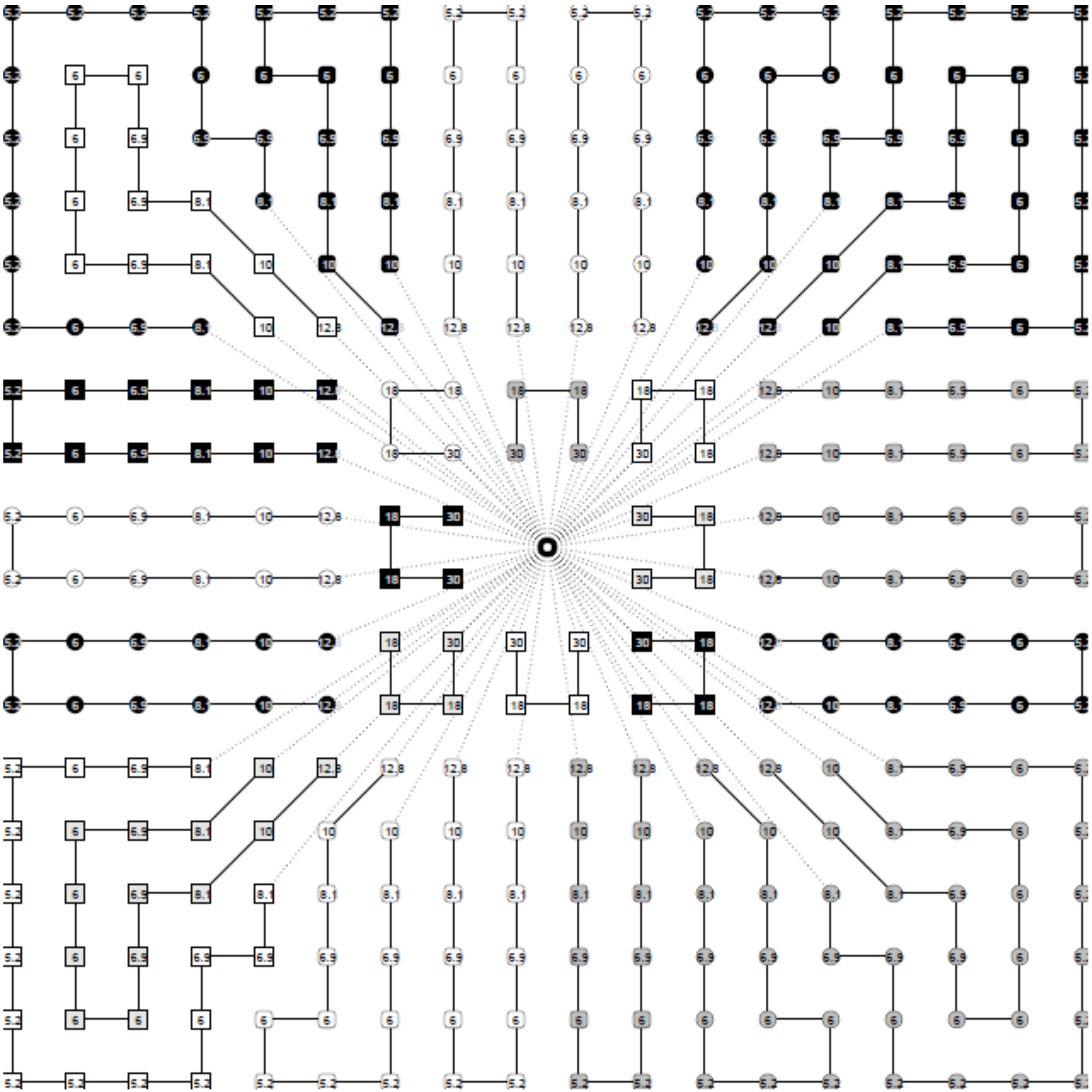



Golden_17
(n=240,
Q=200)

# Golden, Wasil, Kelly and Chao Instances (1998)

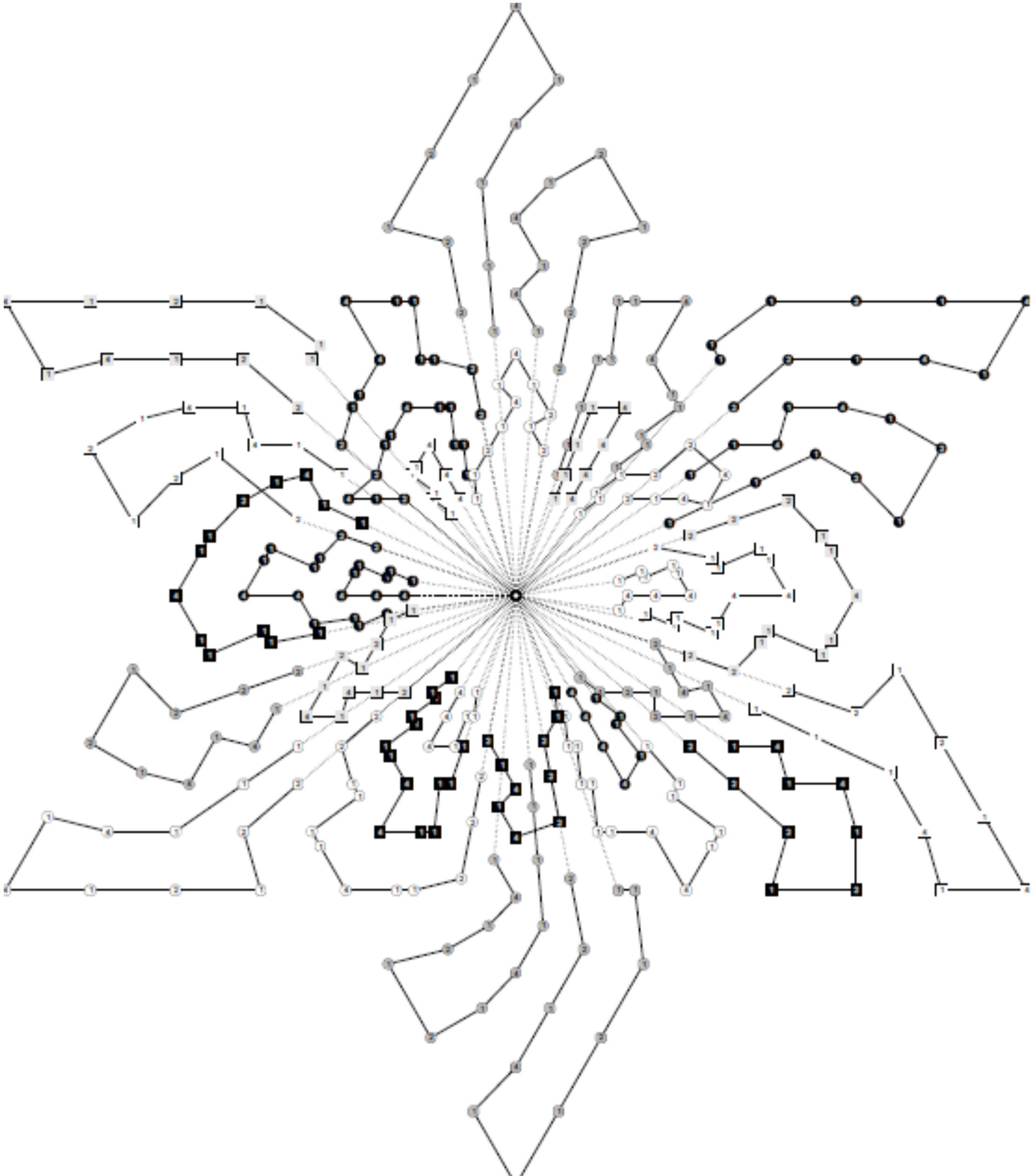
Hard for most heuristics in the literature

- However, Nagata and Bräysy (2009) found 11 BKS, and Vidal et at. (2012) found all 12 BKS.
- Recently, PPPU14 proved that 4 of those BKS are optimal
  - The difficulty of solving those instances comes from their high degree of symmetry



Golden_14
(n = 320,
Q=1000)

Optimal:
1080.55



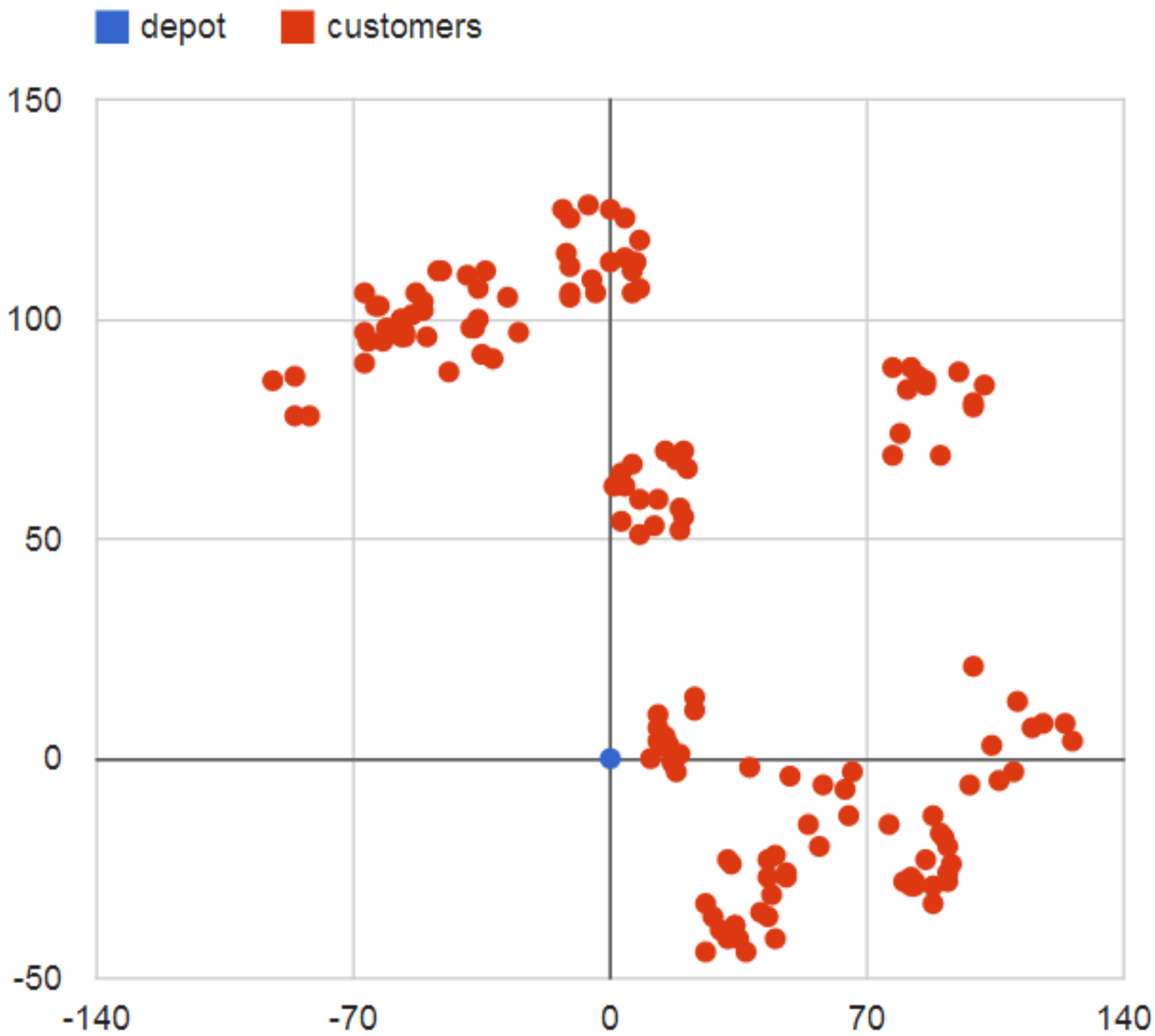
Golden_19
(n = 360,
Q = 200)

Optimal:
1365.60

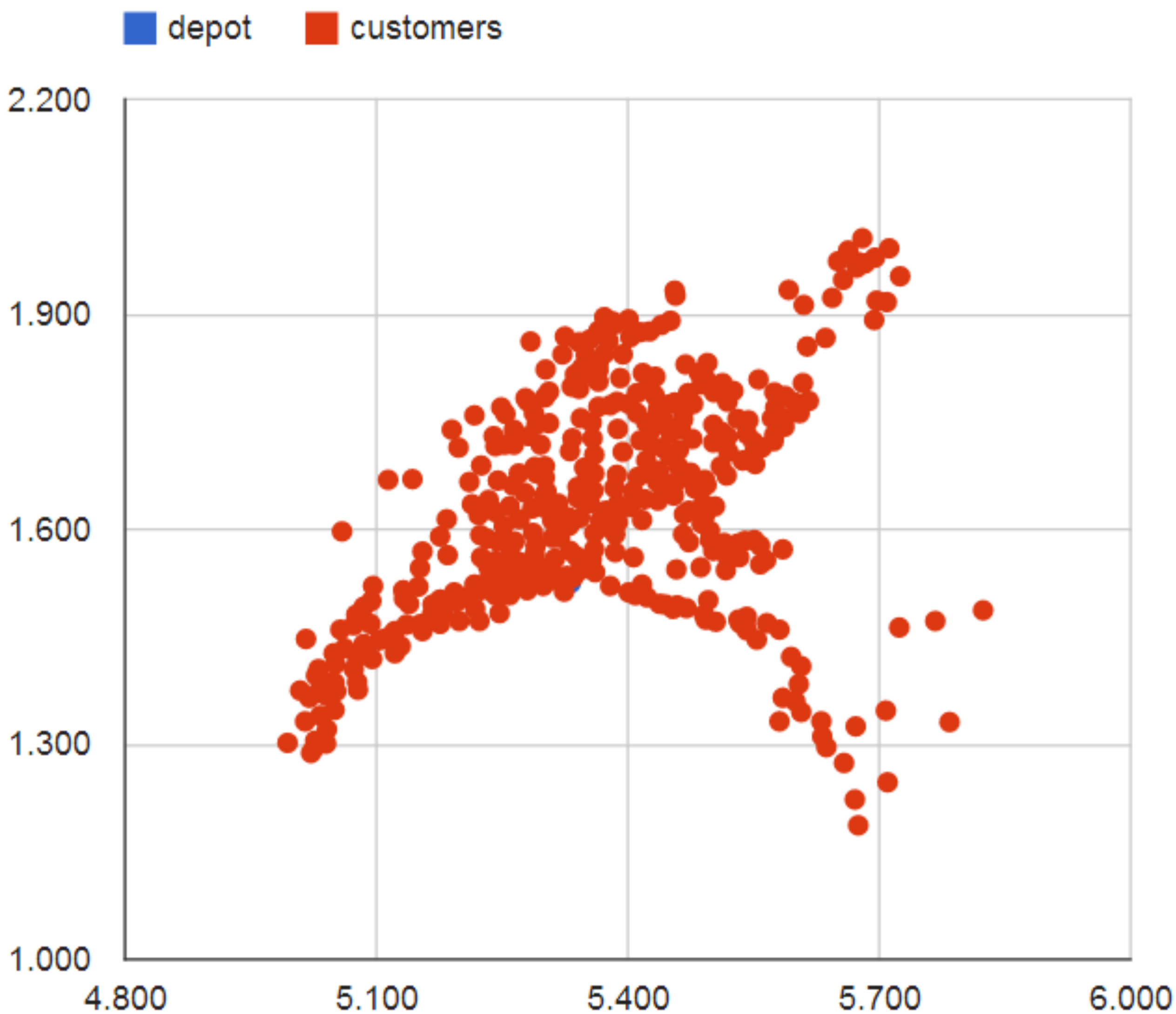
# Taillard Instances (1993)

A little less popular

- 4 instances for each n in {75,100,150}. All were already solved to optimality
- There is also a single real world instance with n=385, it remains open




tai150a
(n=150,
Q=1544)




tai385
(n=385,
Q=65)

# Existing benchmark instances

In our view, the current CVRP benchmarks are nearly exhausted

- For example, there is essencially a single instance (already solved) in the wide interval from 151 to 239 customers

# Newly Proposed Instances

- Depot and customers are positioned in a grid [0,1000]x[0,1000]
- 50 instances between 100 and 330 customers
- 50 instances between 335 and 1000 customers

# Depot Positioning

There are 3 possibilities:

- Central – (500,500)
- Eccentric – corner (0,0)
- Random

# Customer Positioning

3 possibilities:

- Random
- Clustered
- Random-Clustered

  (half-random, half-clustered)

# Clustering Method

- The number of seeds ($S$) is taken from U(3,8)
- The seeds "attract" customers with an exponential decay
- The probability that a position $p$ in the grid receives a customer is proportional to:

$$\sum_{s=1}^{S} e^{-dist(p,s)/40}$$

# Demand Distribution

7 possibilities:

- Unitary
- Small demands, large variance: U[1,10]
- Small demands, small variance: U[5,10]
- Large demands, large variance: U[1,100]
- Large demands, small variance: U[50,100]

# Demand Distribution

- Depending on position:
  - U[1,50] if customer is in an odd quadrant (with respect to (500,500)),
  - U[51,100] otherwise
- Many small (U[1,10]) and a few large (U(50,100))

# Average Route Size (n/K)

One of the most important characteristics of an instance.

$K = \lceil \sum_{i=1}^{n} d_i / Q \rceil$ is a LB (almost always tight) on the minimum possible number of routes

- The desired value of n/K is taken from a triangular distribution T(3,10,25)
- A capacity Q that better approximates this value is chosen

# Newly Proposed Instances

Of course, it is not possible to generate the “Cartesian product” of all those characteristics

- Instead, the generated 100 instances correspond to a sampling

# New Instances 1-20

| # | Name | Short | n | Dep | Cust | S | Dem | Q | n/K |
|---|---|---|---|---|---|---|---|---|---|
| 1 | X-n101-k25 | X101 | 100 | R | RC | 7 | 1-100 | 206 | 4,0 |
| 2 | X-n106-k14 | X106 | 105 | E | C | 3 | 50-100 | 600 | 7,5 |
| 3 | X-n110-k13 | X110 | 109 | C | R | - | 5-10 | 66 | 8,4 |
| 4 | X-n115-k10 | X115 | 114 | C | R | - | SL | 169 | 11,4 |
| 5 | X-n120-k6 | X120 | 119 | E | RC | 8 | U | 21 | 19,8 |
| 6 | X-n125-k30 | X125 | 124 | R | C | 5 | Q | 188 | 4,1 |
| 7 | X-n129-k18 | X129 | 128 | E | RC | 8 | 1-10 | 39 | 7,1 |
| 8 | X-n134-k13 | X134 | 133 | R | C | 4 | Q | 643 | 10,2 |
| 9 | X-n139-k10 | X139 | 138 | C | R | - | 5-10 | 106 | 13,8 |
| 10 | X-n143-k7 | X143 | 142 | E | R | - | 1-100 | 1190 | 20,3 |
| 11 | X-n148-k46 | X148 | 147 | R | RC | 7 | 1-10 | 18 | 3,2 |
| 12 | X-n153-k22 | X153 | 152 | C | C | 3 | SL | 144 | 6,9 |
| 13 | X-n157-k13 | X157 | 156 | R | C | 3 | U | 12 | 12,0 |
| 14 | X-n162-k11 | X162 | 161 | C | RC | 8 | 50-100 | 1174 | 14,6 |
| 15 | X-n167-k10 | X167 | 166 | E | R | - | 5-10 | 133 | 16,6 |
| 16 | X-n172-k51 | X172 | 171 | C | RC | 5 | Q | 161 | 3,4 |
| 17 | X-n176-k26 | X176 | 175 | E | R | - | SL | 142 | 6,7 |
| 18 | X-n181-k23 | X181 | 180 | R | C | 6 | U | 8 | 7,8 |
| 19 | X-n186-k15 | X186 | 185 | R | R | - | 50-100 | 974 | 12,3 |
| 20 | X-n190-k8 | X190 | 189 | E | C | 3 | 1-10 | 138 | 23,6 |

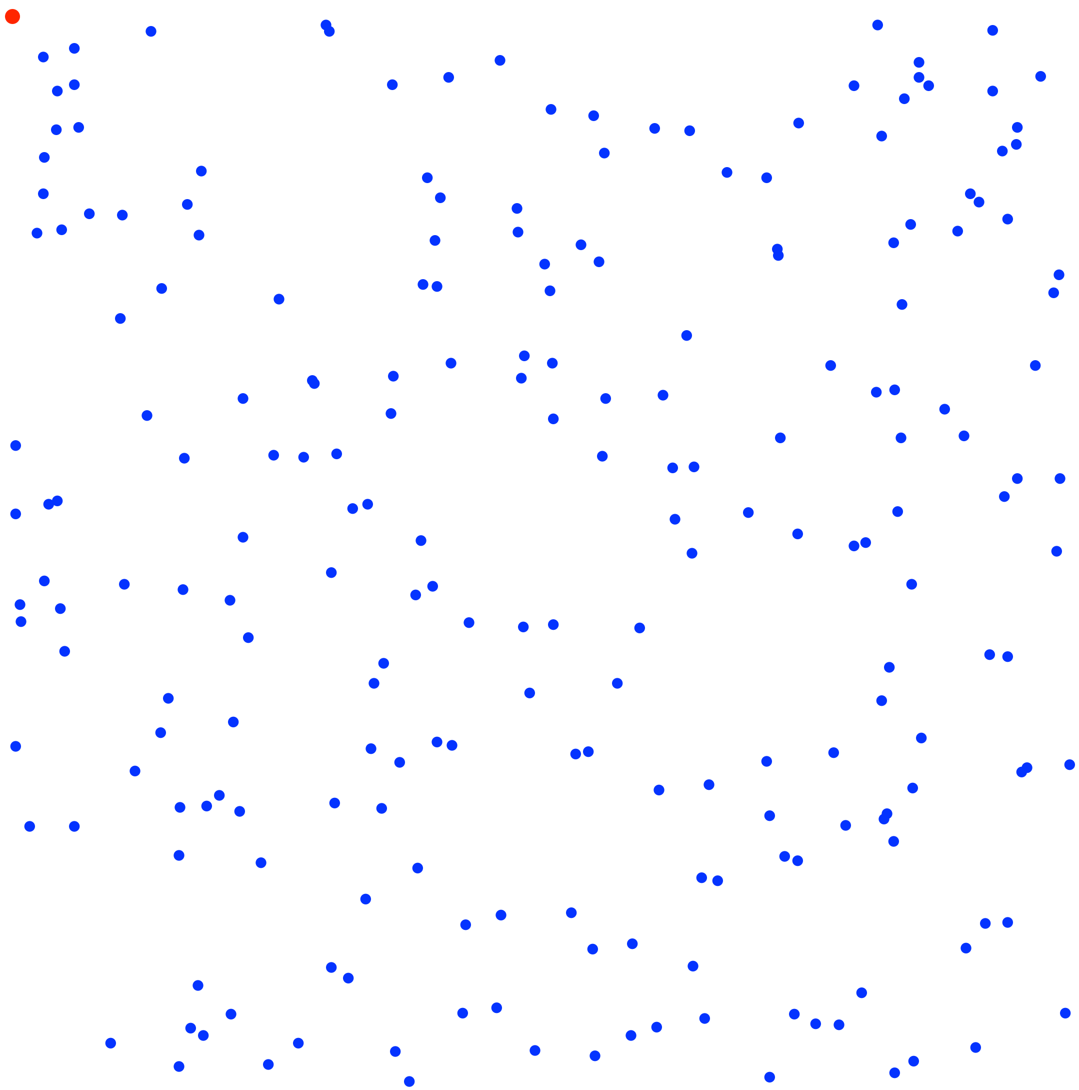
# New Instances 21-40

| # | Name | Short | n | Dep | Cust | S | Dem | Q | n/K |
|---|---|---|---|---|---|---|---|---|---|
| 21 | X-n195-k51 | X195 | 194 | C | RC | 5 | 1-100 | 181 | 3,8 |
| 22 | X-n200-k36 | X200 | 199 | R | C | 8 | Q | 402 | 5,5 |
| 23 | X-n204-k19 | X204 | 203 | C | RC | 6 | 50-100 | 836 | 10,7 |
| 24 | X-n209-k16 | X209 | 208 | E | R | - | 5-10 | 101 | 13,0 |
| 25 | X-n214-k11 | X214 | 213 | C | C | 4 | 1-100 | 944 | 19,4 |
| 26 | X-n219-k73 | X219 | 218 | E | R | - | U | 3 | 3,0 |
| 27 | X-n223-k34 | X223 | 222 | R | RC | 5 | 1-10 | 37 | 6,5 |
| 28 | X-n228-k23 | X228 | 227 | R | C | 8 | SL | 154 | 9,9 |
| 29 | X-n233-k16 | X233 | 232 | C | RC | 7 | Q | 631 | 14,5 |
| 30 | X-n237-k14 | X237 | 236 | E | R | - | U | 18 | 16,9 |
| 31 | X-n242-k48 | X242 | 241 | E | R | - | 1-10 | 28 | 5,0 |
| 32 | X-n247-k47 | X247 | 246 | C | C | 4 | SL | 134 | 5,2 |
| 33 | X-n251-k28 | X251 | 250 | R | RC | 3 | 5-10 | 69 | 8,9 |
| 34 | X-n256-k16 | X256 | 255 | C | C | 8 | 50-100 | 1225 | 15,9 |
| 35 | X-n261-k13 | X261 | 260 | E | R | - | 1-100 | 1081 | 20,0 |
| 36 | X-n266-k58 | X266 | 265 | R | RC | 6 | 5-10 | 35 | 4,6 |
| 37 | X-n270-k35 | X270 | 269 | C | RC | 5 | 50-100 | 585 | 7,7 |
| 38 | X-n275-k28 | X275 | 274 | R | C | 3 | U | 10 | 9,8 |
| 39 | X-n280-k17 | X280 | 279 | E | R | - | SL | 192 | 16,4 |
| 40 | X-n284-k15 | X284 | 283 | R | C | 8 | 1-10 | 109 | 18,9 |

# New Instances 81-100

| # | Name | Short | n | Dep | Cust | S | Dem | Q | n/K |
|---|---|---|---|---|---|---|---|---|---|
| 81 | X-n655-k131 | X655 | 654 | C | C | 4 | U | 5 | 5,0 |
| 82 | X-n670-k126 | X670 | 669 | R | R | - | SL | 129 | 5,3 |
| 83 | X-n685-k75 | X685 | 684 | C | RC | 6 | Q | 408 | 9,1 |
| 84 | X-n701-k44 | X701 | 700 | E | RC | 7 | 1-10 | 87 | 15,9 |
| 85 | X-n716-k35 | X716 | 715 | R | C | 3 | 1-100 | 1007 | 20,4 |
| 86 | X-n733-k159 | X733 | 732 | C | R | - | 1-10 | 25 | 4,6 |
| 87 | X-n749-k98 | X749 | 748 | R | C | 8 | 1-100 | 396 | 7,6 |
| 88 | X-n766-k71 | X766 | 765 | E | RC | 7 | SL | 166 | 10,8 |
| 89 | X-n783-k48 | X783 | 782 | R | R | - | Q | 832 | 16,3 |
| 90 | X-n801-k40 | X801 | 800 | E | R | - | U | 20 | 20,0 |
| 91 | X-n819-k171 | X819 | 818 | C | C | 6 | 50-100 | 358 | 4,8 |
| 92 | X-n837-k142 | X837 | 836 | R | RC | 7 | 5-10 | 44 | 5,9 |
| 93 | X-n856-k95 | X856 | 855 | C | RC | 3 | U | 9 | 9,0 |
| 94 | X-n876-k59 | X876 | 875 | E | C | 5 | 1-100 | 764 | 14,8 |
| 95 | X-n895-k37 | X895 | 894 | R | R | - | 50-100 | 1816 | 24,2 |
| 96 | X-n916-k207 | X916 | 915 | E | RC | 6 | 5-10 | 33 | 4,4 |
| 97 | X-n936-k151 | X936 | 935 | C | R | - | SL | 138 | 6,2 |
| 98 | X-n957-k87 | X957 | 956 | R | RC | 4 | U | 11 | 11,0 |
| 99 | X-n979-k58 | X979 | 978 | E | C | 6 | Q | 998 | 16,9 |
| 100 | X-n1001-k43 | X1001 | 1000 | R | R | - | 1-10 | 131 | 23,3 |

# X-n219-k73 or X219



n: 218

depot positioning: eccentric

customer positioning: random

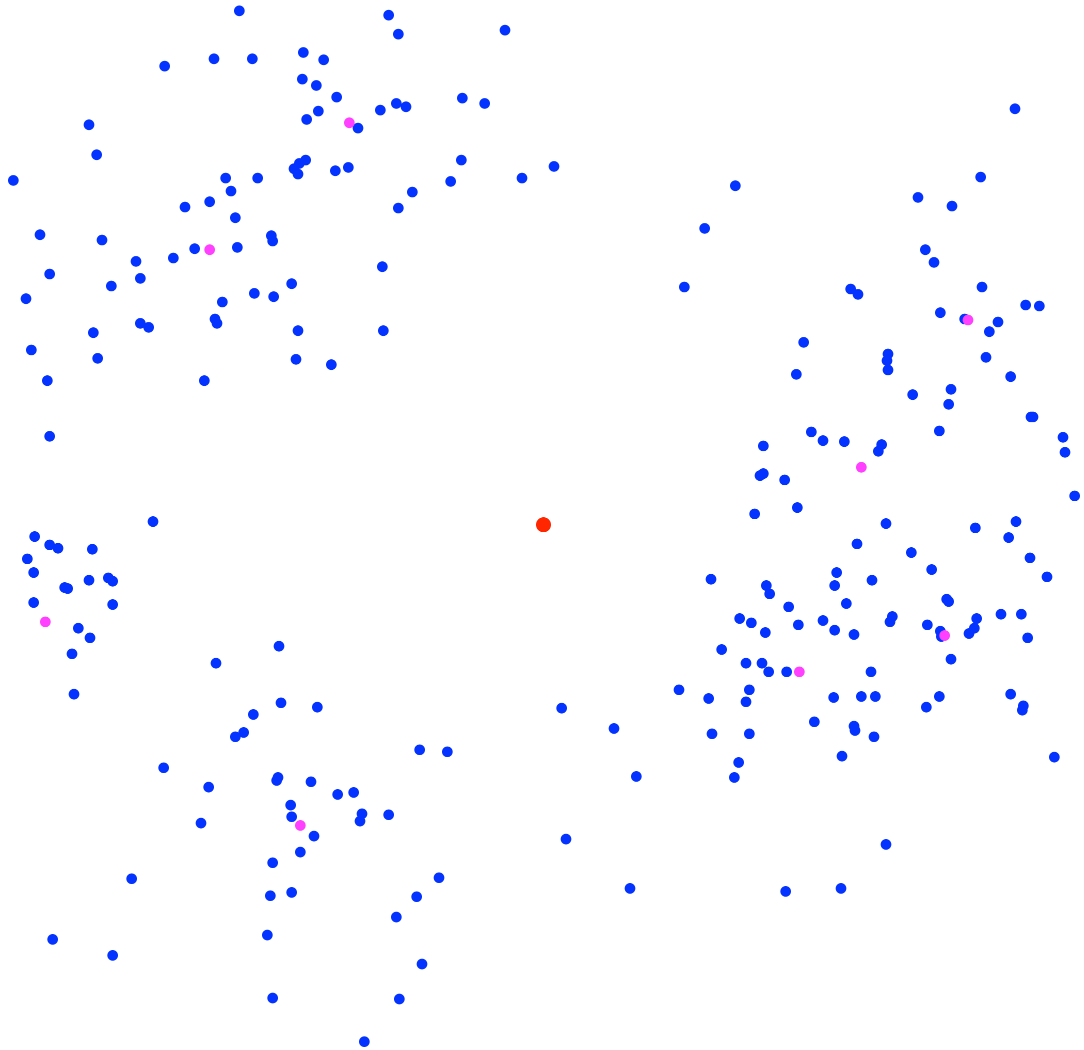
demands: unitary

Q: 3

K: 73

n/K: 2,98

# X-n256-k16 or X256



n: 255

depot positioning: center

customer positioning: clustered (8 seeds)

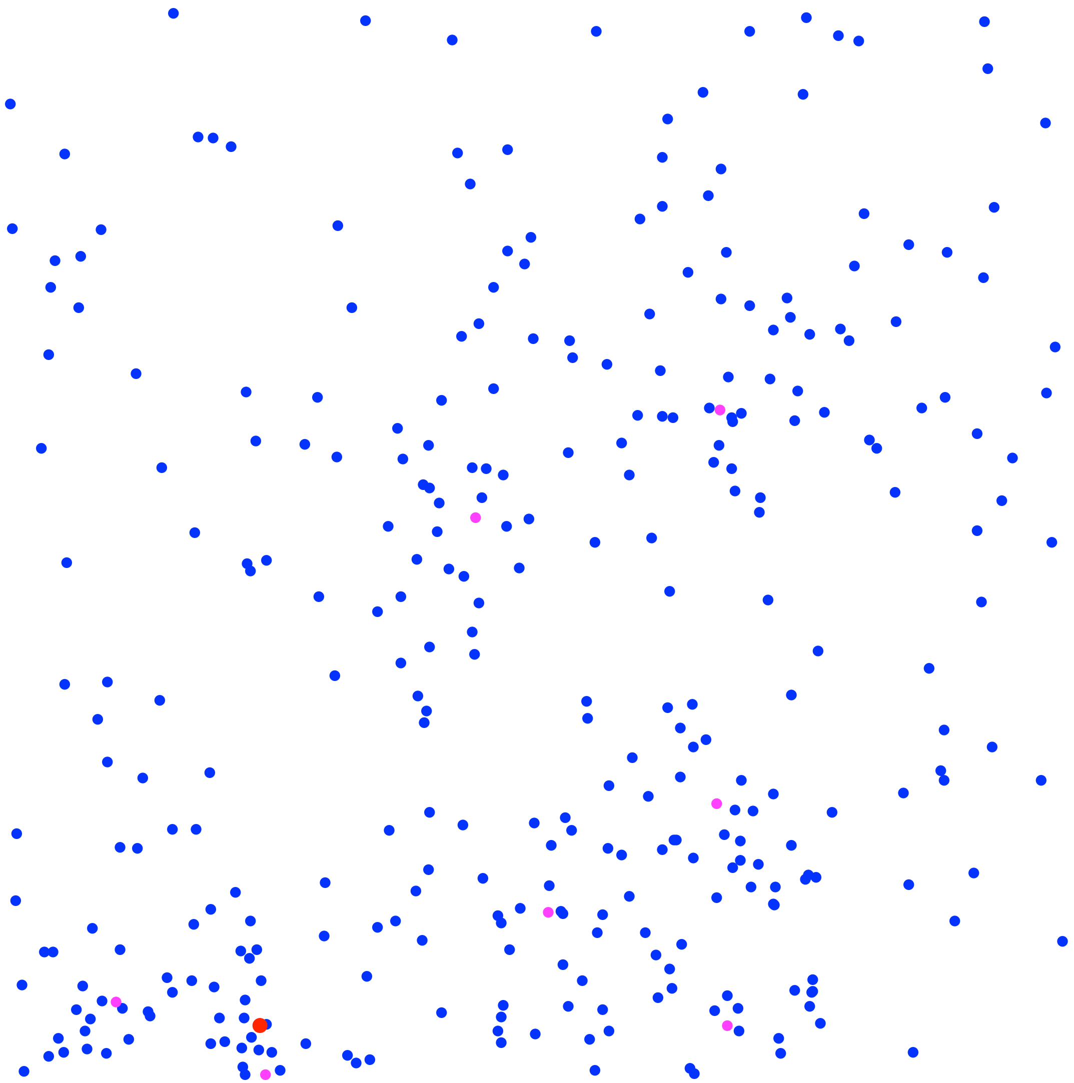
demands: [50-100]

Q: 1225

K: 16

n/K: 15.9

# X-n327-k20 or X327



n: 326

depot positioning: random

customer positioning: random-clustered (7 seeds)

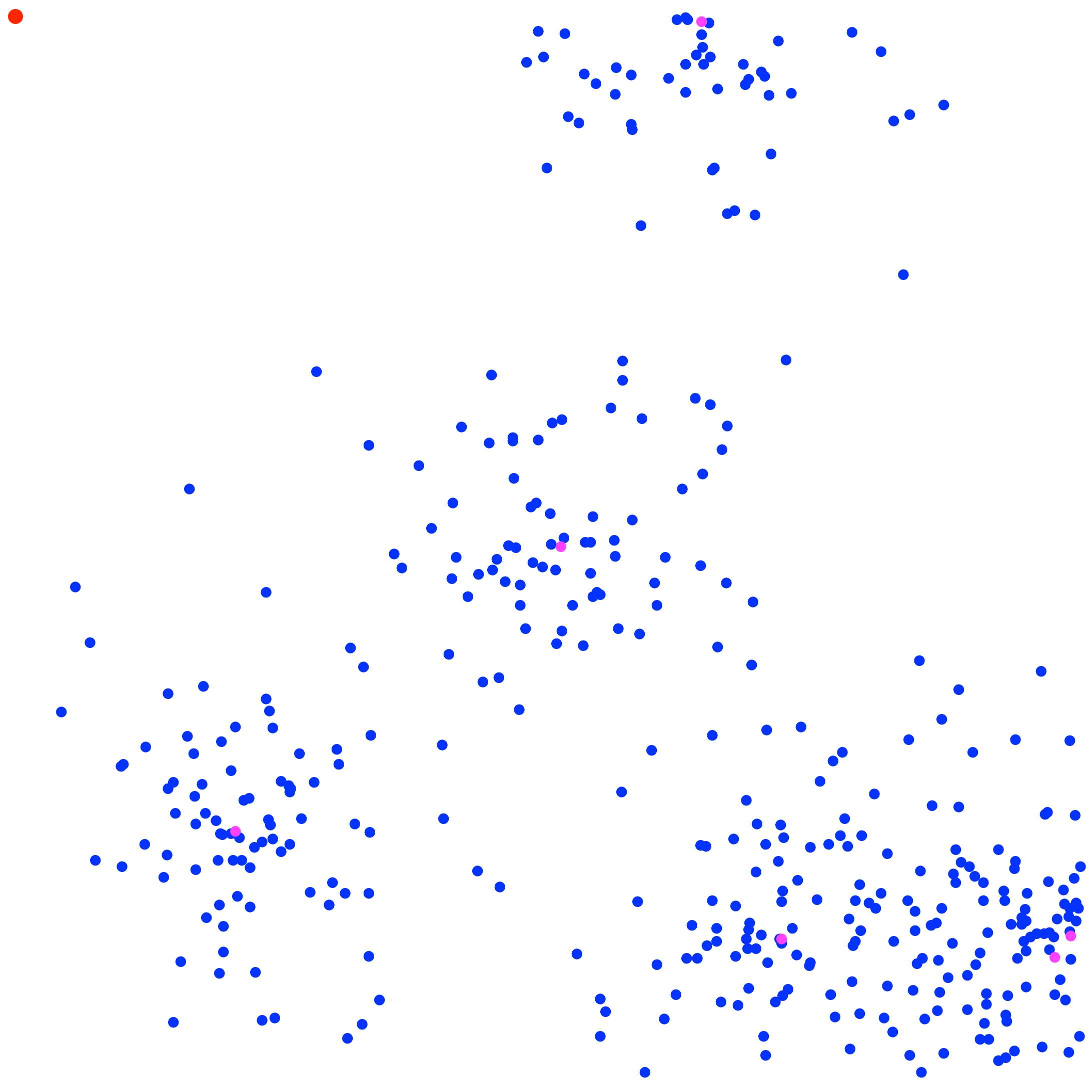
demands: [5-10]

Q: 128

K: 20

n/K: 16.3

# X-n401-k29 or X401



n: 400

depot positioning: eccentric

customer positioning: clustered (6 seeds)

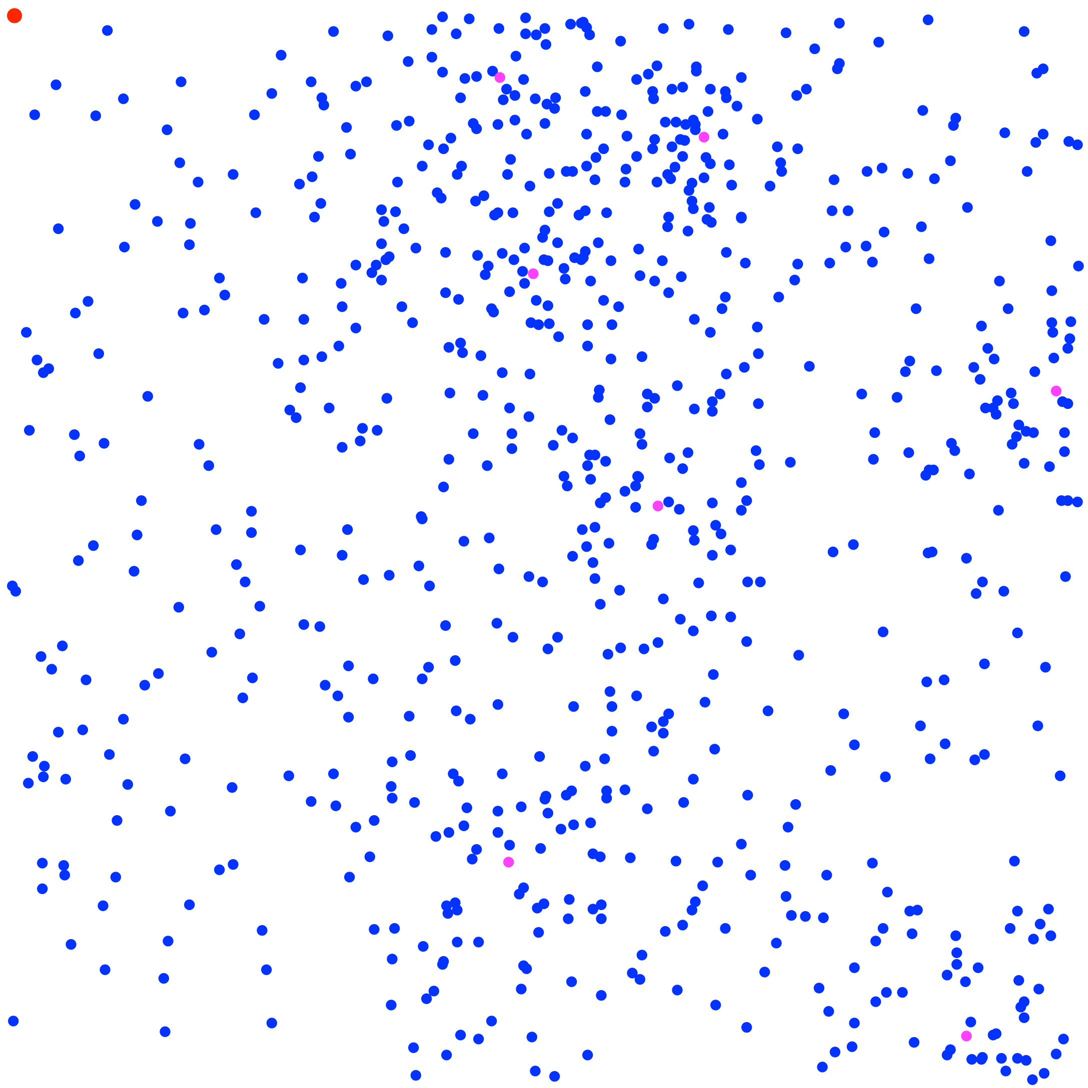
demands: Q [1-50] odd quadrant, [51-100] even quadrant

Q: 745

K: 29

n/K: 13.8

# X-n766-k71 or X766



n: 765

depot positioning: eccentric

customer positioning: random-clustered (7 seeds)

demands: SL 105 customers [1-50], 660 customers [1-10]

Q: 166

K: 71

n/K: 10.8

# Two Decisions

# To Round Or Not To Round?

Nearly all existing instances are euclidean, depot and customer coordinates are given:

- Following the TSPLIB convention, in the literature on exact methods the distances are rounded to the nearest integer.
- In the literature on heuristics, distances are seldom rounded.

# Advantages of Rounding

Most Mathematical Programming based algorithms (including standard MIP solvers) have a limited optimality precision:

- CPLEX default: only $10^{-4}$ (0.01%)!
- Can be increased to $10^{-6}$ or even $10^{-7}$
- More precision may require special software (more bits in the floating-point numbers or even exact rational arithmetic).

# Advantages of Rounding

Distance rounding usually makes the optimal values found by standard MP methods reliable

- The practice of not rounding, but only reporting two decimal places does not solve the problem:
  - 853.2350001 -> 853.24
  - 853.2349999 -> 853.23

# Advantages of Rounding?

Rounding allows exact methods “to cheat” a bit!

- If an UB of 1000 is known, a B&B node with LB 999.01 can be fathomed
- Most existing instances have optima around $10^3$ -> enough to make some algorithms to be more than twice as fast

# Disadvantages of Rounding

Rounded instances are more likely to have alternative optimal solutions, therefore:

- A benchmark of rounded instances has less power for comparing competing algorithms (specially heuristics), since there are more ties
- This effect is significant on instances with optima around $10^3$

# Decision: To Round

The new instances were created to have optima around $10^5$:

- Large enough to avoid significant “B&B cheating” and many alternative optimal solutions
- Small enough to avoid precision problems

# To Fix Or Not To Fix?

- In the literature on exact methods, the number of routes is fixed to a value $K$ (almost always to the minimum possible).
- In the literature on heuristics, the number of routes is not fixed.

# Why fixing the number of routes?

Standard explanation:

- $K$ is the size of the fleet

In our opinion, this is pretty unrealistic

More sophisticated explanation:

- Fixing $K$ to the minimum possible is a way of taking the fixed cost of using a vehicle into account

# Decision: Not To Fix

Decisive argument:

- If CVRP allows solutions with very short and very long routes, why not assigning the two shortest routes to the same vehicle?

Routes do not need to correspond to vehicles.

In some real cases, the same vehicle performs all routes (a 100% homogeneous fleet!)

# Decision: Not To Fix

Moreover:

- The original CVRP definition in Dantzig and Ranser does not fix the number of routes

# Experiments with the new instances

We performed tests with 2 state-of-the-art algorithms:

- The Branch-Cut-and-Price (BCP) in PPPU14 (that will be presented tomorrow)
- The Unifyied Hybrid Genetic Search (UHGS) by Vidal, Crainic, Gendreau and Prins (2014)

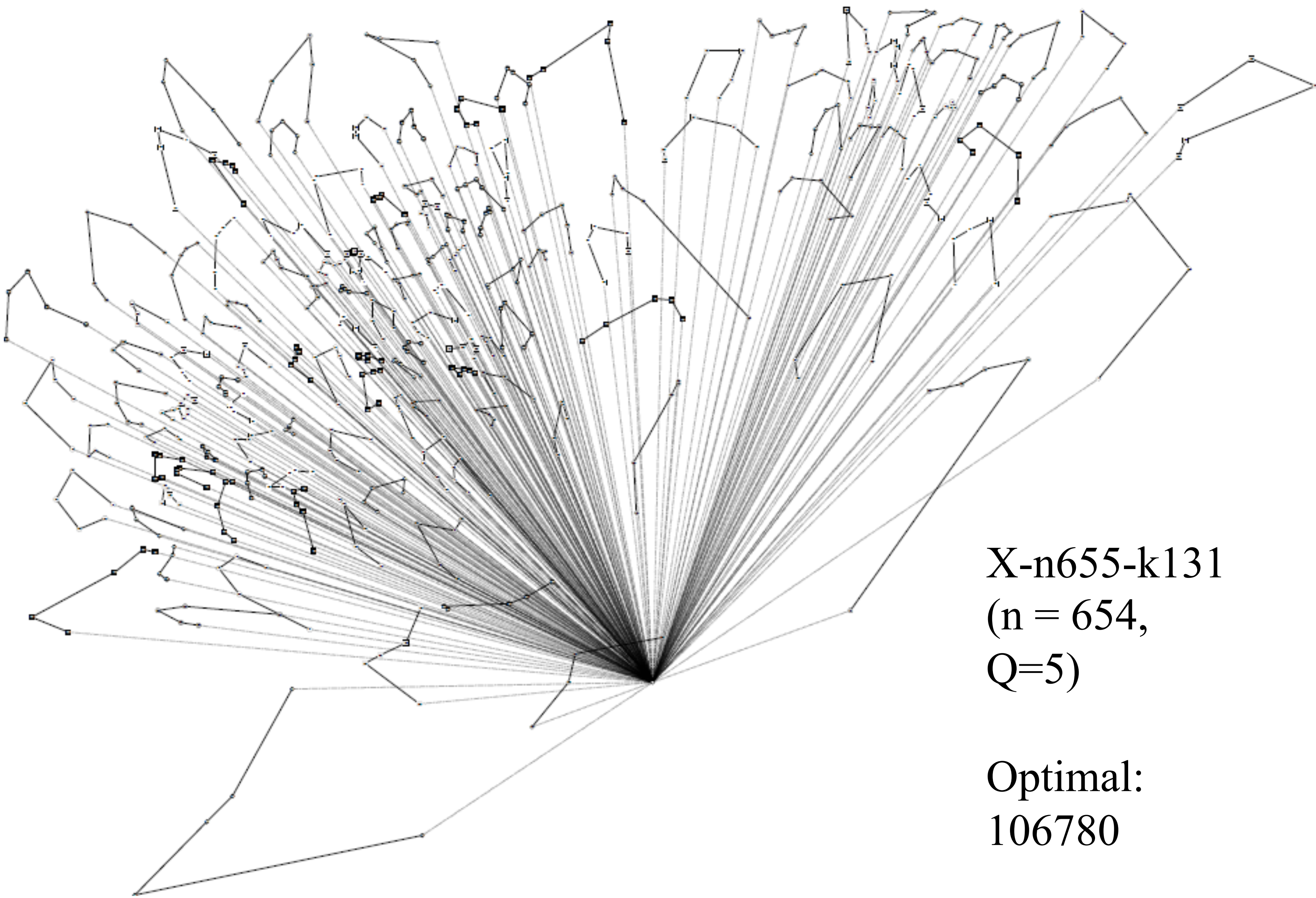
# Experiments with the new instances

We performed tests with 2 state-of-the-art algorithms:

- The Branch-Cut-and-Price (BCP) in PPPU14 (that will be presented tomorrow)
- The Unifyied Hybrid Genetic Search (UHGS) by Vidal, Crainic, Gendreau and Prins (2014)

# BCP

- 38 instances could be solved to optimality
  - $100 \leq n < 200$: 21 out of 22
  - $200 \leq n < 300$: 13 out of 21
  - $300 \leq n < 500$: 3 out of 25
  - $500 \leq n \leq 1000$: 1 out of 32
- Smallest unsolved: X-n190-k8 (projected cpu time: a few months)
- Largest solved: X-n655-k131 (2491s)



X-n655-k131
(n = 654,
Q=5)

Optimal:
106780

# What makes an instance X harder to BCP?

Besides the obvious value of n:

- Large n/K: a very significant effect
- The absolute value of Q: significant effect

Surprisingly, depot and customer positioning does not have a clear influence

# UHGS

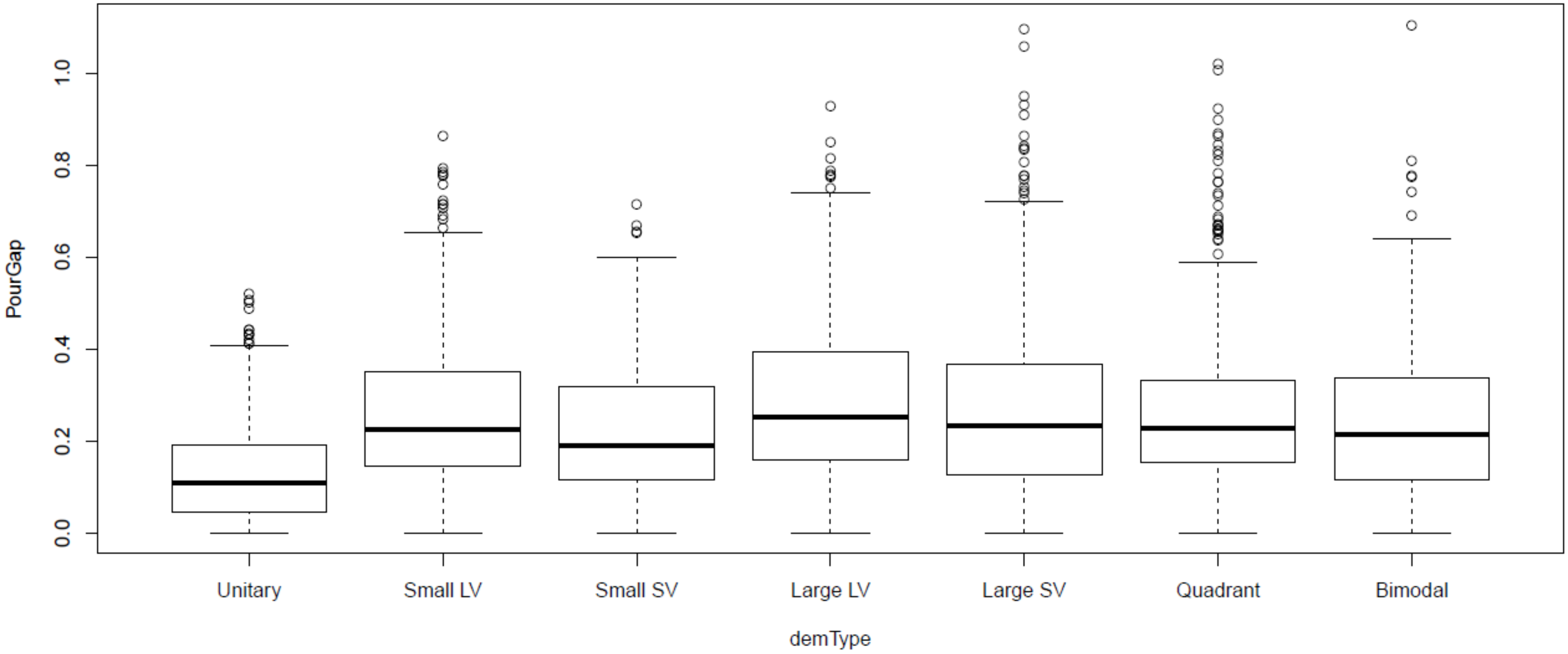
UHGS was run 50 times in each instance:

- It found 34 out of the 38 proven optima. The 4 instances where the optimal was not found have $n/K \leq 5$.
  - Small n/K good for BCP, bad for UHGS

# What else makes an instance X harder to UHGS?

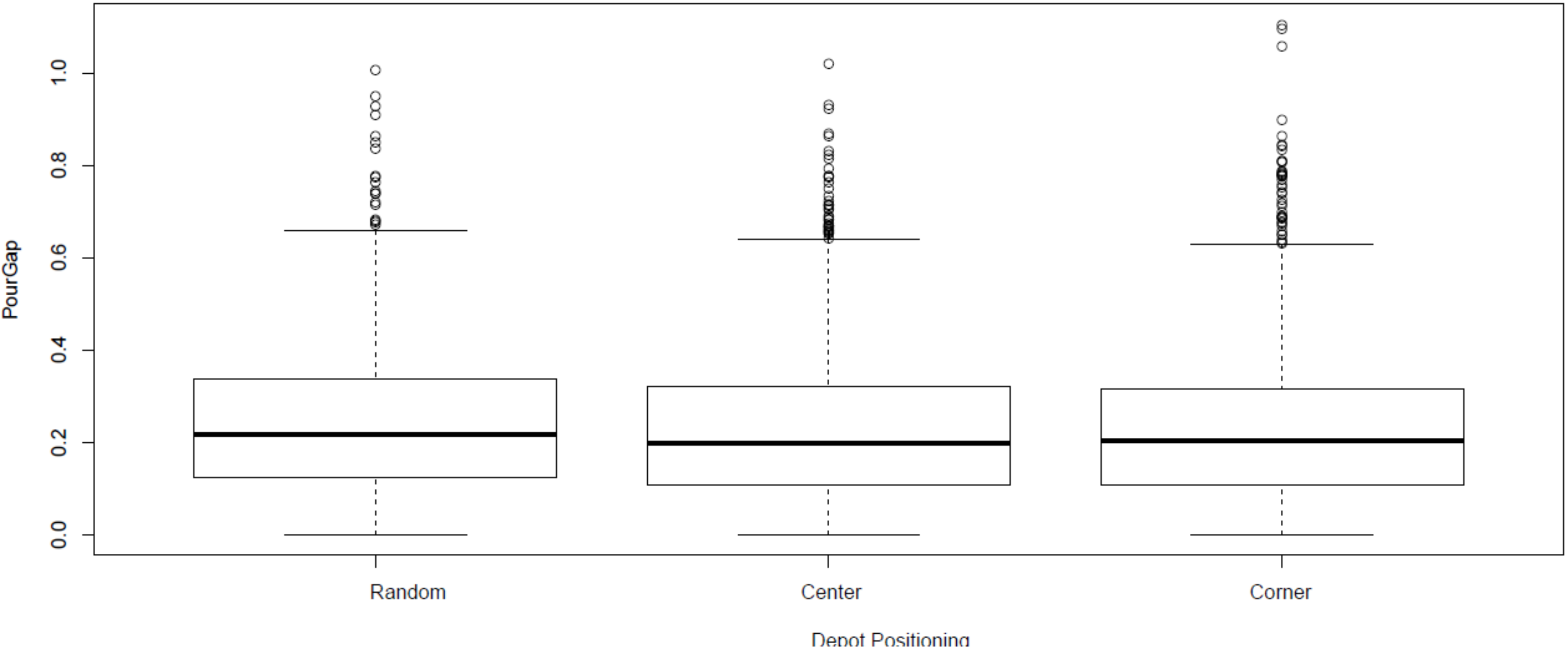
By measuring the average gap between the solution found in each run and the BKS/optimal, it is possible to assess the effect of instance characteristics

# Effect of demand type on UHGS




Box-plots indicate significant influence of some demand types

# Effect of depot position on UHGS




Box-plots do not indicate significant influence

# Web site

The X instances are available at:

http://vrp.atd-lab.inf.puc-rio.br/

- The site also provides detailed information on all other CVRP instances, including plots, optimal/best known solutions.
- Soon, the users will be able to submit improving solutions automatically

# Conclusions

- We are open to feedback from the community
- We hope that “heuristic people” and “exact people” start working on the same instances!


# Tak!