Vehicle Routing Problem (VRP)

$V = \{1, \ldots, n\}$ - fleet of vehicles

$D = \{1, \ldots, k\}$ - set of depots (*usually there is one*)

$C = \{k+1, \ldots, k+m\}$ - set of clients (requests)

$location_j \in R^2$ - coordinates of a client/depot

$\vartheta(i, j)$ - distance between locations i and j

$r_i = (i_1, \ldots, i_{p(i)})$ - permutation of requests assigned to vehicle i (i_1 and $i_{p(i)}$ are depots)

$$COST(VRP) = \min_{(r_1, \ldots, r_n)} \sum_{i=1}^{n} \sum_{j=2}^{p(i)} \vartheta(i_j, i_{j-1})$$

$$\forall_{l\in\{1,\ldots,m\}}\exists!_{i\in\{1,\ldots,n\}}\,(l+k)\in r_i$$

Capacitated VRP ((C)VRP)

Additional capacity constraints:

$capacity$ - capacity of each vehicle (fleet V is homogeneous)

$size_l \leq capacity$ - size of request of client l

A sum of requests' sizes between consecutive visits to the depots (for a given vehicle) must not exceed its $capacity$.

Dynamic Capacitated VRP (D(C)VRP)

Various additional time constraints imposed:

working hours of depots

time window for delivery at a certain location

time for unloading a cargo

time of availability (for the solver) of a given request

traffic jams

VRPDR [Kilby et al., 1998]

$(start_j, end_j), \quad 0 \le start_j < end_j$ - working hours of depot j

$unld_l \in R$ - time required to unload a cargo for client l

$time_l \in R$ - point in time at which request for client l becomes available

VRPDR

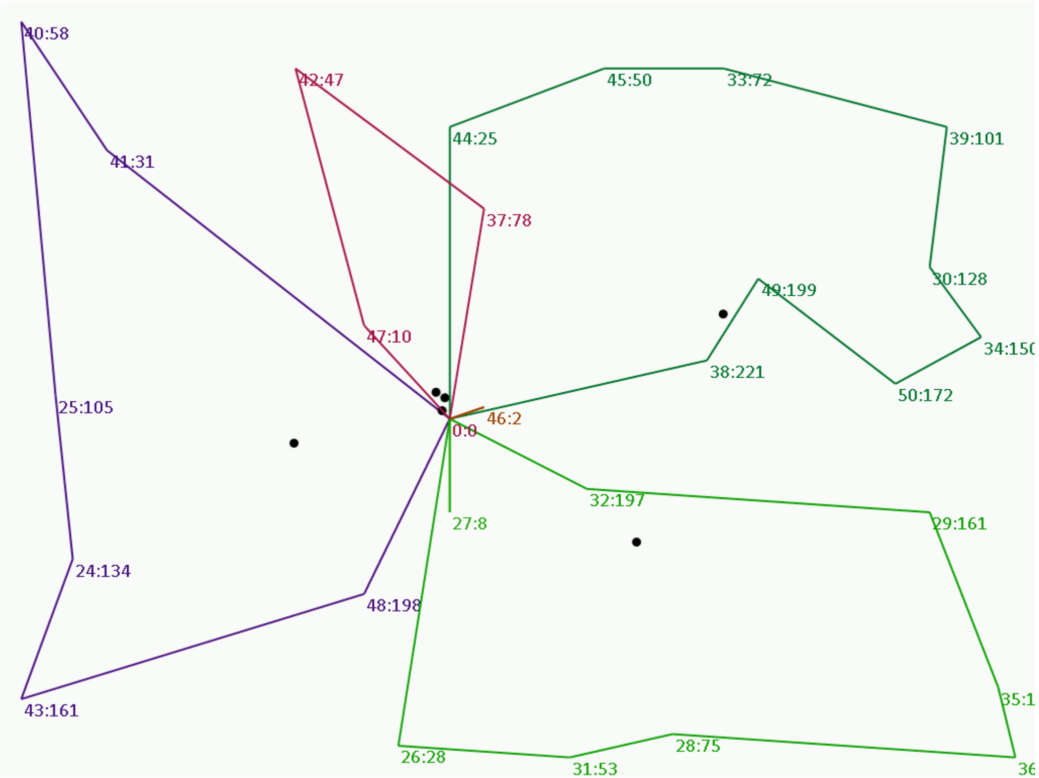
Additional constraints:

$$\forall_{i\in\{1,\dots,n\}}\forall_{j\in r_i\setminus i_1}\ arv_{i_j} \geq time_{i_j} + \vartheta(i_j, i_{j-1})$$

$$\forall_{i\in\{1,\dots,n\}}\forall_{j\in r_i\setminus i_1}\ arv_{i_j} \geq arv_{i_{j-1}} + unld_{i_{j-1}} + \vartheta(i_j, i_{j-1})$$

$$\forall_{i\in\{1,\dots,n\}} arv_{i_{p(i)}} \leq end_{i_{p(i)}}$$

D(C)VRP vs. (C)VRP



- Clients
  - Location
  - *Request size*
  - **Request availability time**
  - **Unload time**
- Vehicles
  - *Capacity*
- Depot
  - Location
  - **Opening hours**
- Distances between locations
- Goal
  - Minimize total length of vehicles' routes
- Constraints
  - Each route starts and ends in a depot
  - Each request is served by one vehicle
  - **Vehicles must return to the depot before it is closed**

Difficulty of DVRP

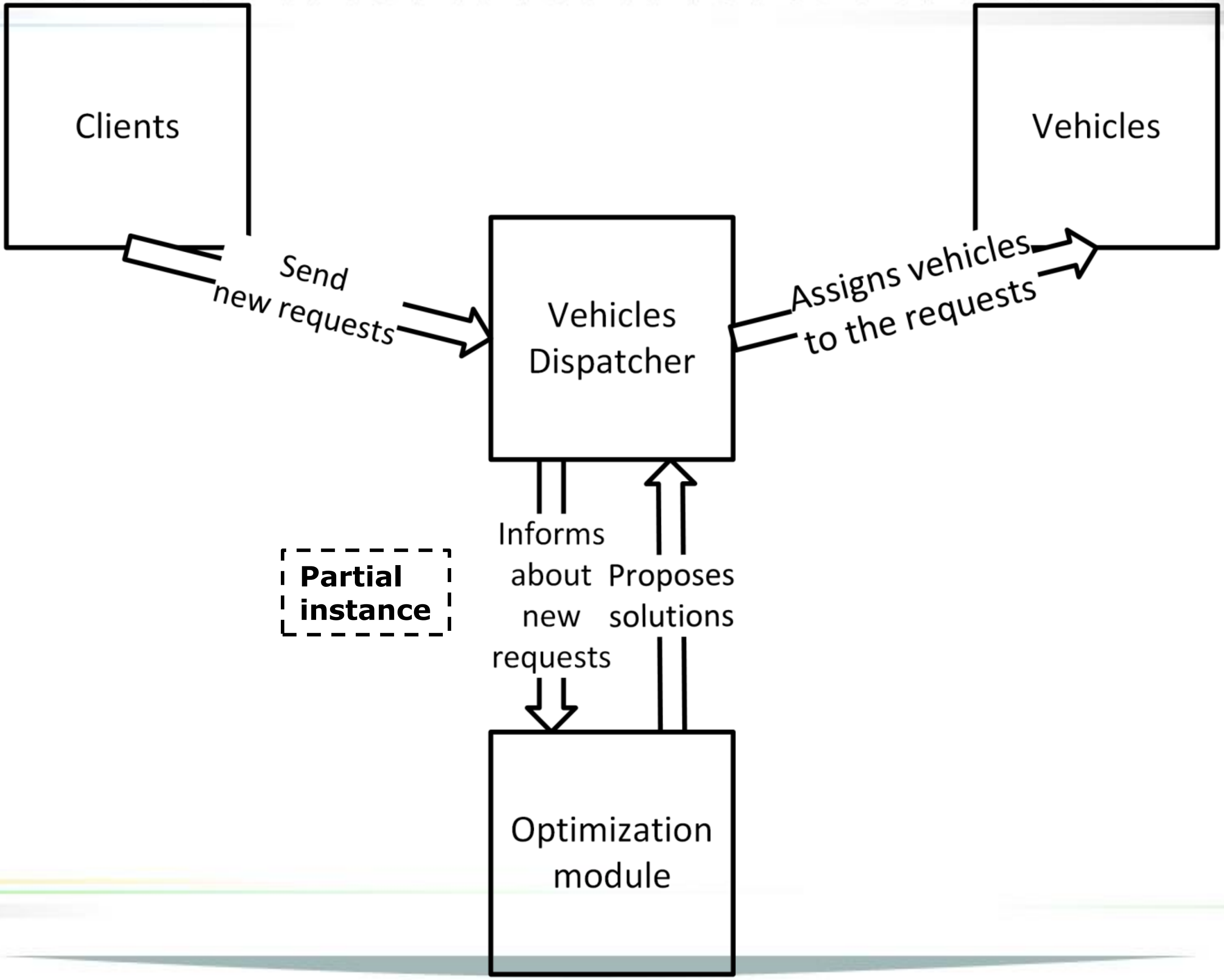
- A challenging NP-Hard problem
- ***Natural environment for testing efficient combinations of various optimization approaches***
- Difficulty stems from several types of time conditions imposed on the test instances
- ***Practical relevance – close-to-reality problem definition***
- Reality ~ Multi-depot Pickup & Delivery VRP with Dynamic Requests, Time Windows and Traffic Jams

Solution Methods

DVRP – CI-based metaheuristics

- Kilby et al. – 1998 – standard benchmark sets
- Tabu Search (TS) – 1999 and 2007
- Ant Colony Systems (ACS) – 2005
- **Genetic algorithms (GA) –** 2007
- **Particle Swarm Optimization (PSO) –** 2010

DVRP - Solution scheme



The „degree of dynamicity”

T_{co} - *cut-off* time $(T_{co} = 0.5)$

n_{ts} - number of *time slices* $(n_{ts} = 25, 50, 100)$

T_{ac} - *advanced commitment* time $\left(T_{ac} = 2\dfrac{end - start}{n_{ts}}\right)$

$$start := \min_{j \in \{1,\dots,k\}} start_j$$

$$end := \max_{j \in \{1,\dots,k\}} end_j$$

A general schema of one-phase methods

- The initial problem = all the requests received after the *cut_off time*

- For each time slice
  - Solve the current VRP instance
  - Select the best solution found (for this time slice)
  - Commit orders to respective vehicles (with processing time within the next $\frac{T}{n_{ts}} + T_{ac}$ seconds)
  - Include requests received during this time slice in the next static instance (next time slice)
  - Send back to the depot any vehicle which might otherwise be late at the depot closing
  - Send back to the depot any vehicle that used its capacity

Genetic Algorithms – one phase method
[Hanshar and Obuki-Berman (2007)]

- A genotype is a sequence of requests assigned to each vehicle ➔ variable length

- Negative values represent vehicles and the parts of routes already committed to them

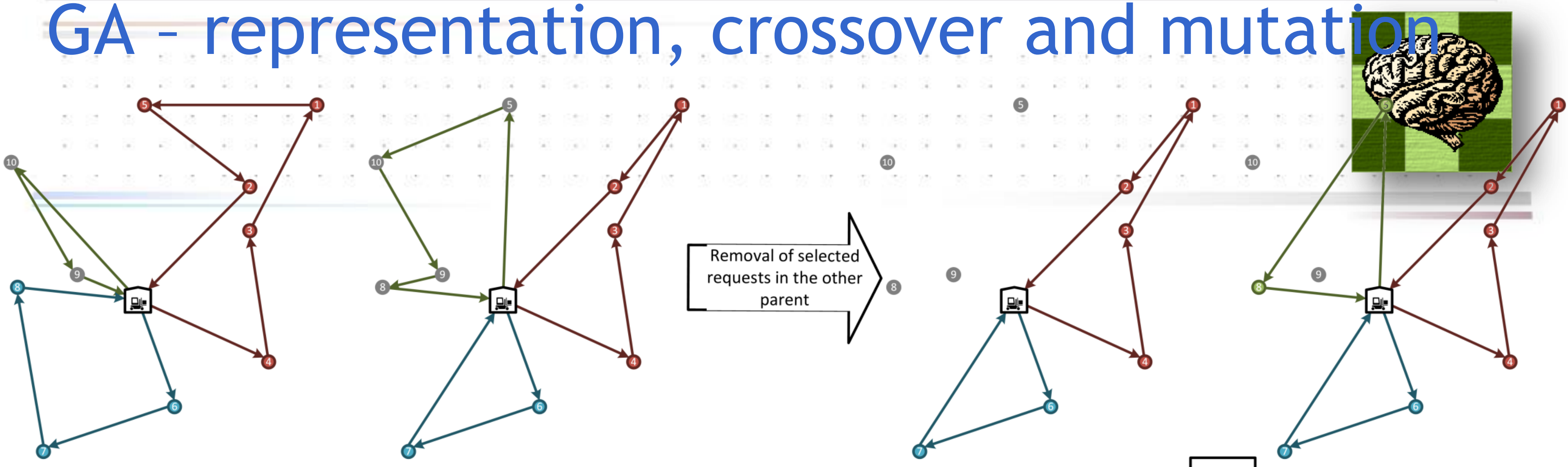
  -3,4,3,1,5,2,**-1**,6,7,8,**-2**,10,9

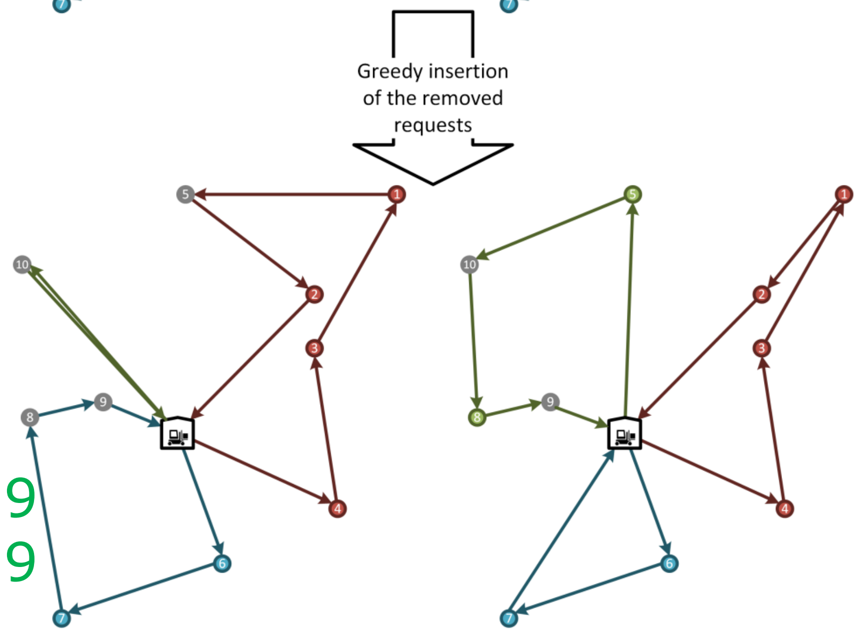
- Straightforward cost function:

$$COST_3(r_1, r_2, \dots, r_n) = \sum_{i=1}^{n} \sum_{j=2}^{|r_i|} \vartheta(i_j, i_{j-1})$$

- K-tournament selection. Pre-defined threshold for promoting the best one.

GA – representation, crossover and mutation





Assignment & route

-3,4,3,1,5,2,**-1**,6,7,8,**-2**,10,9

Mutation - inversion

-3,4,3,1, | 5,2,**-1**,6,7,8,**-2**,|10,9

-3,4,3,1, | 8,7,**-1**,6,2,5,**-2**,|10,9

MAPSO – one phase method [Khouadjia et al. (2010-13)]

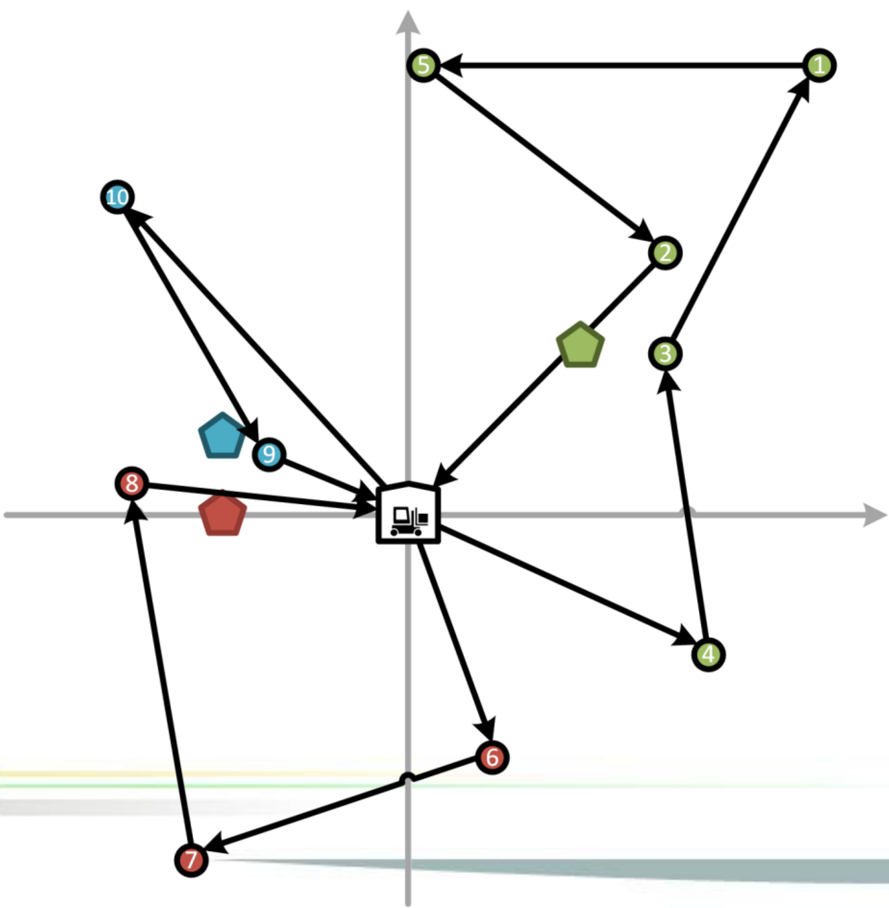
- Multi-swarm optimization
- The route for each vehicle is constructed by a combination of PSO and greedy algorithm and optimized with 2-OPT
- Fitness function is the sum of routes for all vehicles

$$COST_2(r_1, r_2, \ldots, r_n) = \sum_{i=1}^{n} \sum_{j=2}^{|r_i|} \vartheta(i_{2OPT(j)}, i_{2OPT(j-1)})$$

- Yields better results than GA approach

MAPSO

- Integer search space and particles velocity
- Re-initialization preserves previous best particles positions



Assignment of requests

1,1,1,1,1,2,2,2,3,3

1,3,1,1,1,2,2,2,3,3

Every particle represents associations of client's requests to vehicles

- each element of a vector corresponds to one request
- its value is the vehicle id

Velocity vector is calculated in real values, rounded to the nearest integer and added to the position vector modulo *m* (m=3 in this case)

2MPSO [Okulewicz, Mańdziuk]

- Multi-swarm optimization
- Initialization with previous solutions
- Continuous search space and particles velocity
- 2-phase PSO algorithm. **In each time slice**:
  - Partition of the requests optimized as a clustering task
  - Finding the shortest route for each vehicle as a separate optimization task

2MPSO – clustering phase (phase 1)

- Every particle represents coordinates of cluster centers
  - search space dimensionality is equal to the number of vehicles times two
  - the number of required vehicles is estimated with minimum spanning forest method (+buffer)
  - requests are associated with the vehicle corresponding to the closest cluster center
  - one of the particles is initialized at cluster centers generated by capacitated minimum spanning forest

- A fitness function:

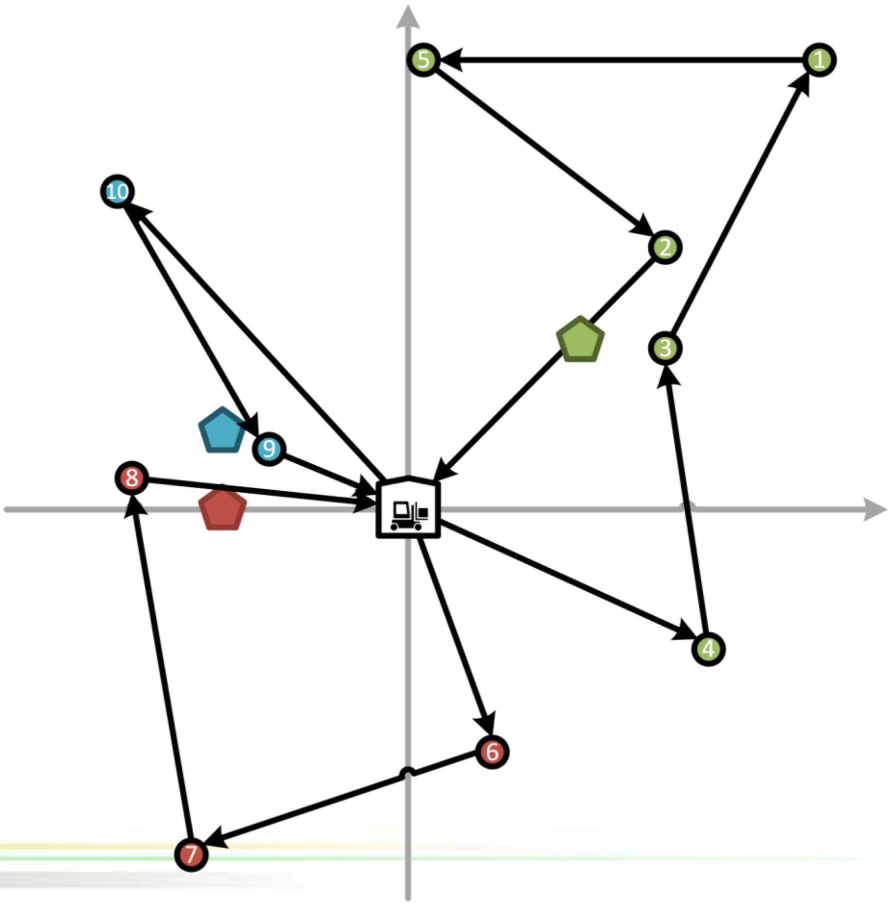
$$COST_2(r_1, r_2, \ldots, r_n) = \sum_{i=1}^{n} \sum_{j=2}^{|r_i|} \vartheta(i_{2OPT(j)}, i_{2OPT(j-1)})$$

2MPSO – route optimization phase (phase 2)

- Every particle represents the order of requests in the vehicle's route
  - search space dimensionality is equal to the number of requests to be served by the vehicle
  - requests are ordered ascending according to their respective values in a vector

- Fitness function is the length of the route optimized with 2-OPT

2MPSO - encoding



Assignment – phase 1

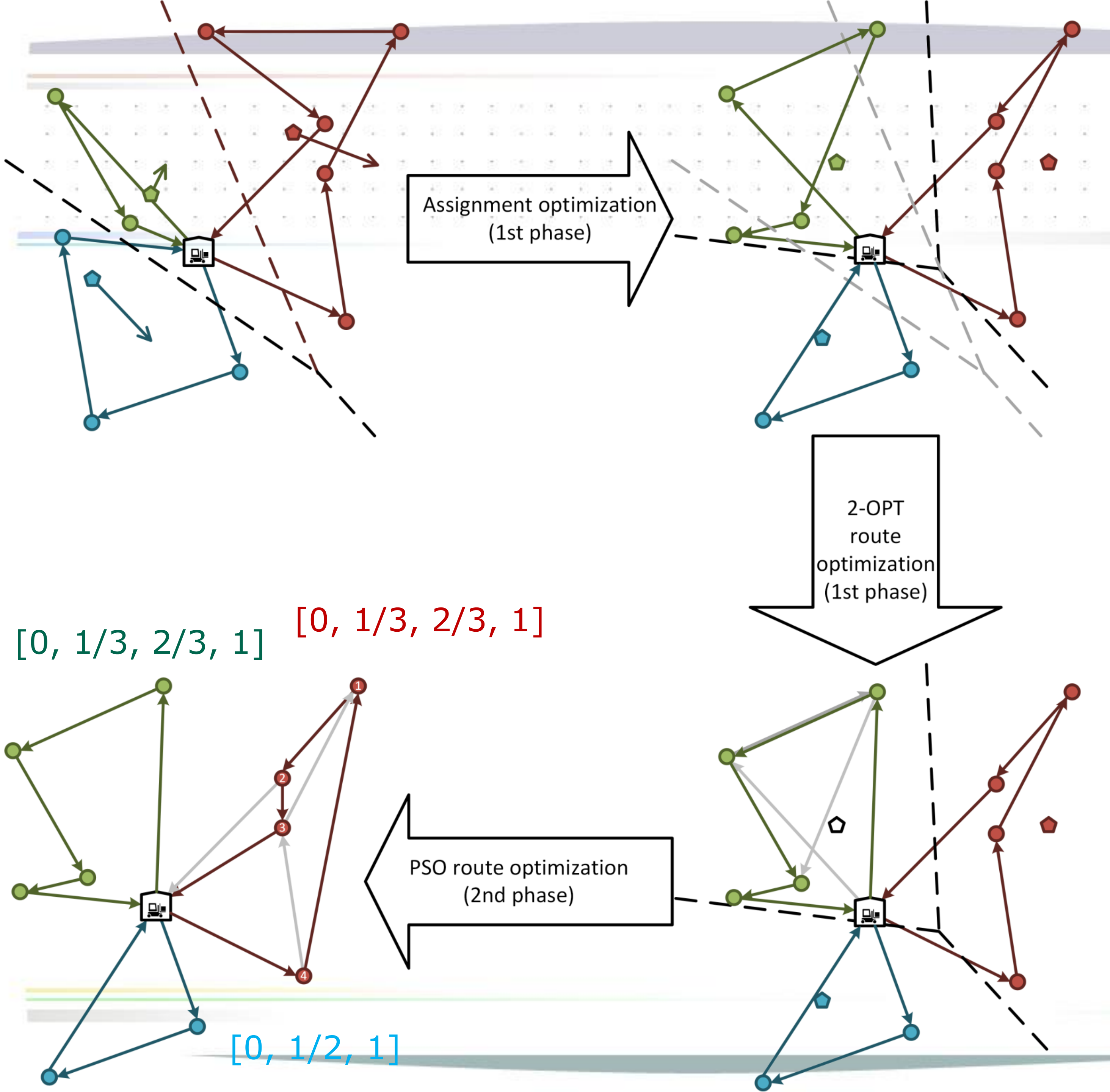
1,1,-1,0,-1,0.1

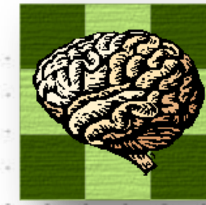
Route – phase 2

0.5, 0.7, 0.2, 0.1,0.6

0.25, 0.3, 0.33

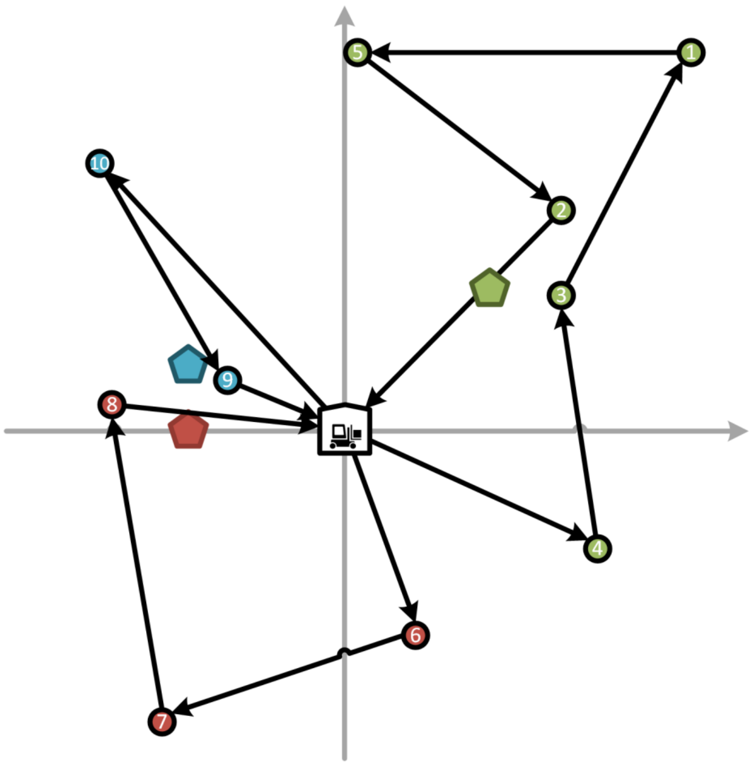
0.5, 0.4





2MPSO Algorithm

Summary of (D)VRP encodings inmetaheuristics



- GA
  - Assignment & route
    - 4,3,1,5, **,-1**,6,7,8**,-2**,10,9
- DAPSO/MAPSO
  - Assignment
    - 1,1,1,1,1,2,2,2,3,3
- 2MPSO
  - Assignment
    - 1,1,-1,0,-1,0.1
  - Route
    - 0.5, 0.7, 0.2, 0.1
    - 0.25, 0.3, 0.33
    - 0.5, 0.4

Knowledge transfer

- Between consecutive time slices
  - Critical issue in dynamic problems
  - It is assumed that solutions in the neighboring slices are linked
  - The initial swarm location is defined around the cluster centers of the previous best known solution within a given radius

- Between swarms (within a given time slice)
  - Each swarm works in isolation
  - One swarm per thread
  - At the end of allotted time all the best overall solution (among all swarms) is selected as the current best solution and spread among the threads
  - MAPSO – migration of particles (with some probability).

Benchmark Sets

Benchmark sets [Kilby et al., 1998]

Christofides – 7 problems

N. Christofides and J. Beasley, The period routing problem. *Networks*, 14:237-256, 1984

Fisher – 2 problems

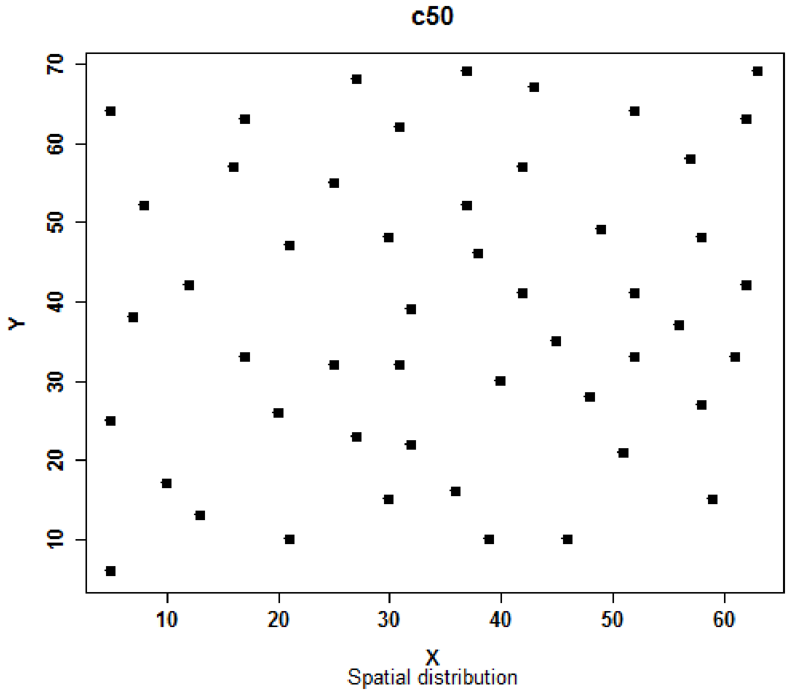
M. Fisher and R. Jakumar, A generalized assignment heuristic for vehicle routing. *Networks*, 11:109-124, 1981

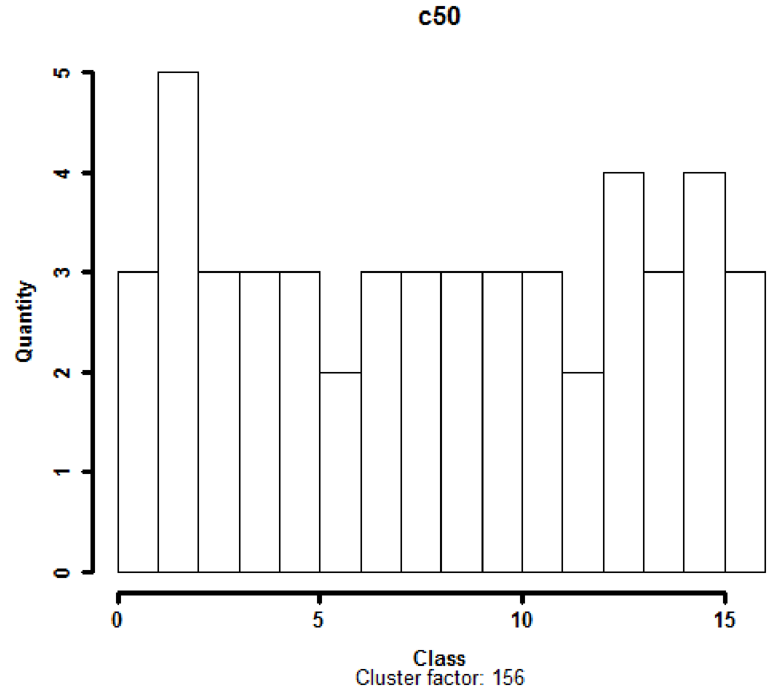
Taillard – 12 problems

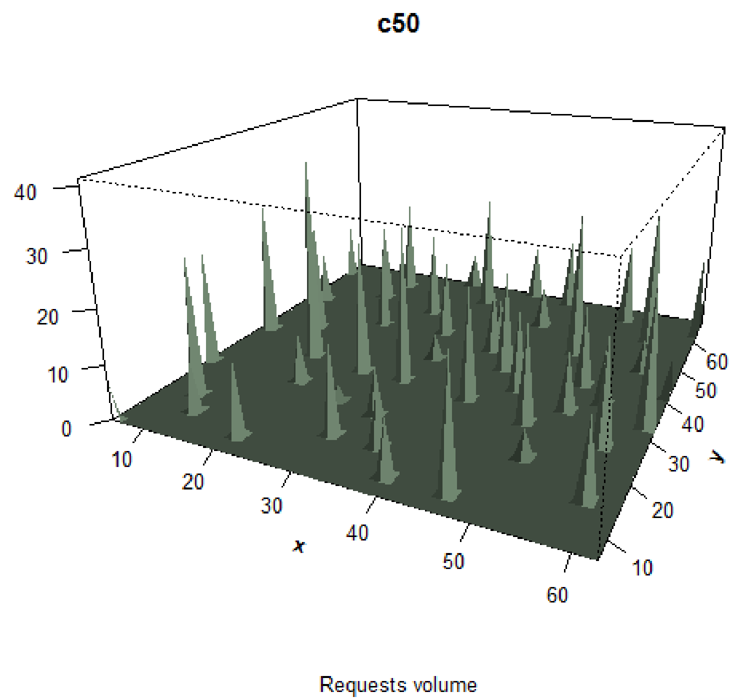
E. Taillard, Parallel iterative search methods for vehicle-routing problems. *Networks*, 23:661-673, 1993

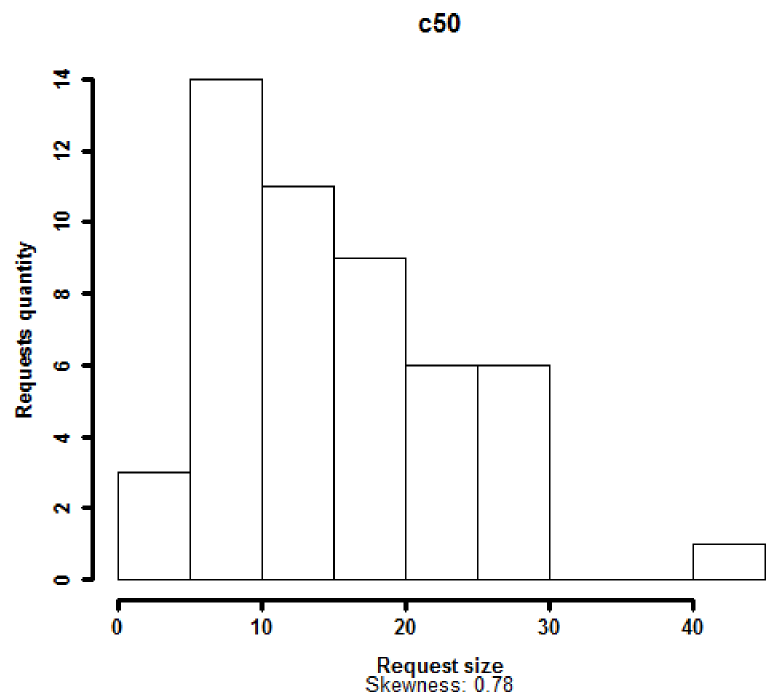
Benchmarks' structure

```
VRPTEST 1.0
COMMENT: Converted from tai150a
COMMENT: by convert2vt
COMMENT: Best known objective: 3055.23
NAME: tai150a
NUM_DEPOTS: 1
NUM_CAPACITIES: 1
NUM_VISITS: 150
NUM_LOCATIONS: 151
NUM_VEHICLES: 50
CAPACITIES: 1544
DATA_SECTION
DEPOTS
  0
DEMAND_SECTION
  1 13
  2 122
  3 30
  4 3
  5 5
  6 422
              ...
150 32
LOCATION_COORD_SECTION
  0 0 0
  1 -4 106
  2 -11 106
  3 -12 115
  4 4 123
  5 7 113
  6 -13 125
              ...
150 -35 92
DEPOT_LOCATION_SECTION
  0 0
VISIT_LOCATION_SECTION
  1 1
  2 2
  3 3
  4 4
  5 5
  6 6
              ...
150 150
DURATION_SECTION
  1 30
  2 30
  3 30
  4 30
  5 30
  6 30
              ...
150 30
DEPOT_TIME_WINDOW_SECTION
  0 0 1062
COMMENT: TIMESTEP: 21
TIME_AVAIL_SECTION
  1 10
  2 19
  3 21
  4 31
  5 33
  6 35
              ...
150 1056
EOF
```

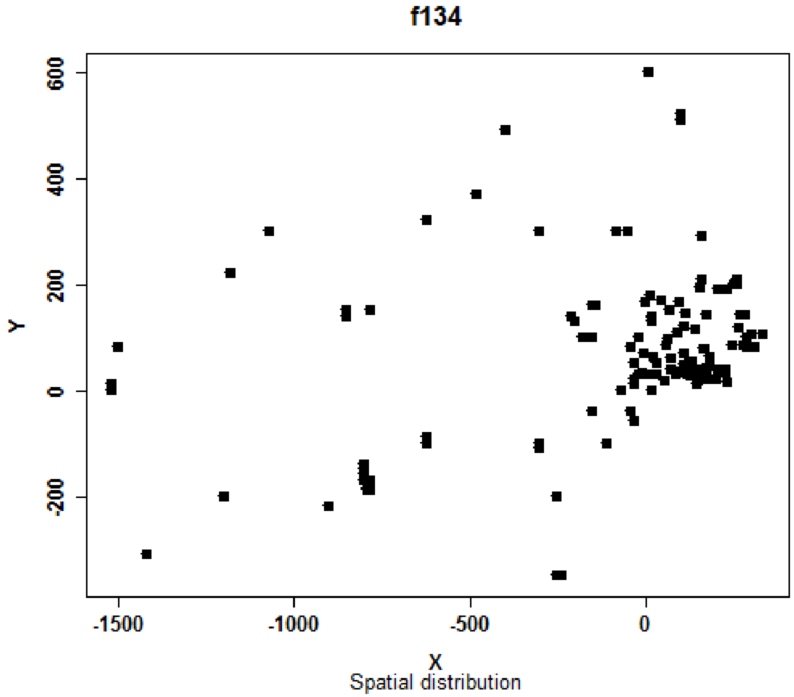

c50
Y
X
Spatial distribution

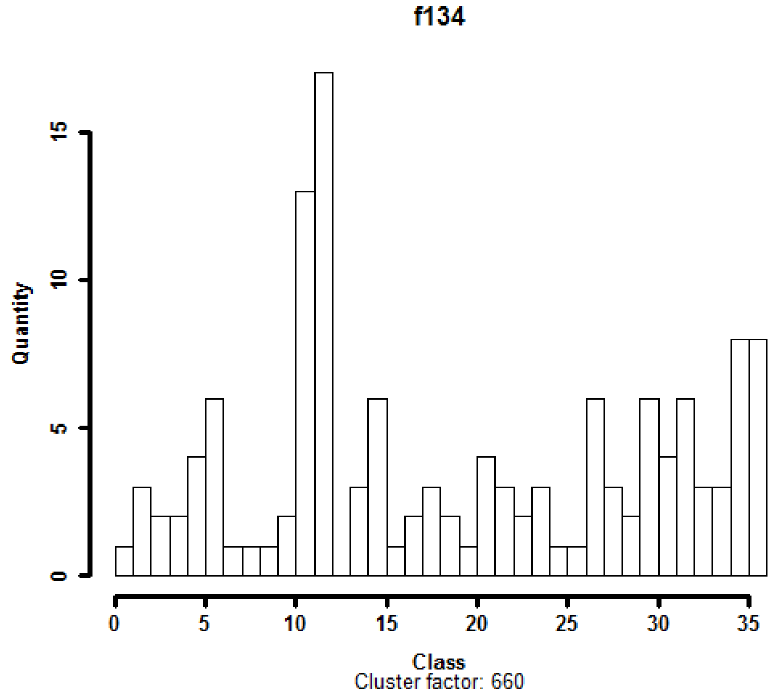

c50
Quantity
Class
Cluster factor: 156

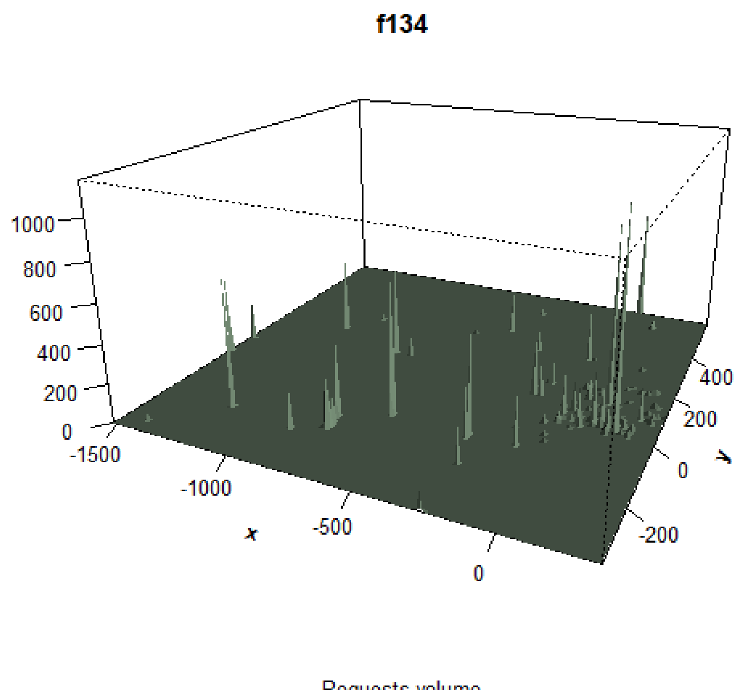

c50
x
y
Requests volume

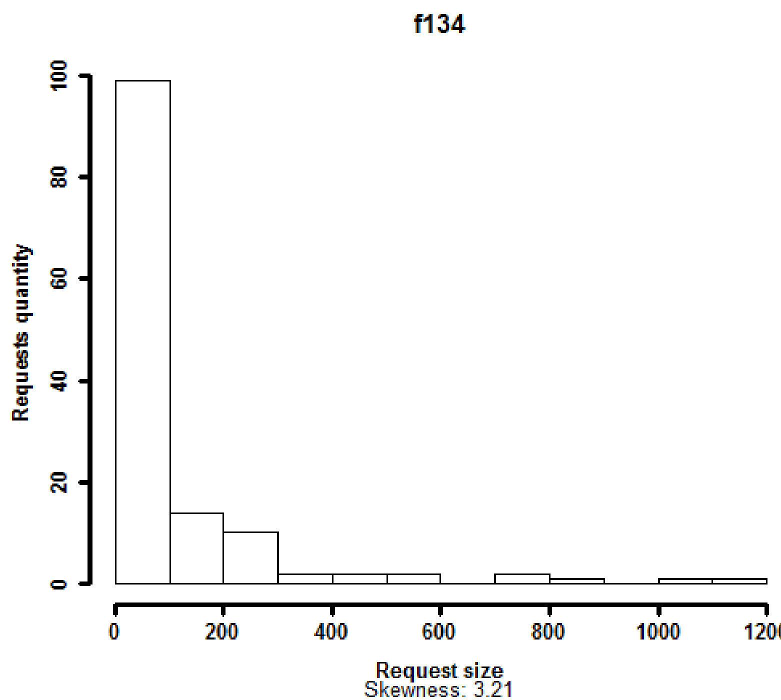

c50
Requests quantity
Request size
Skewness: 0.78

c50


f134
Y
X
Spatial distribution

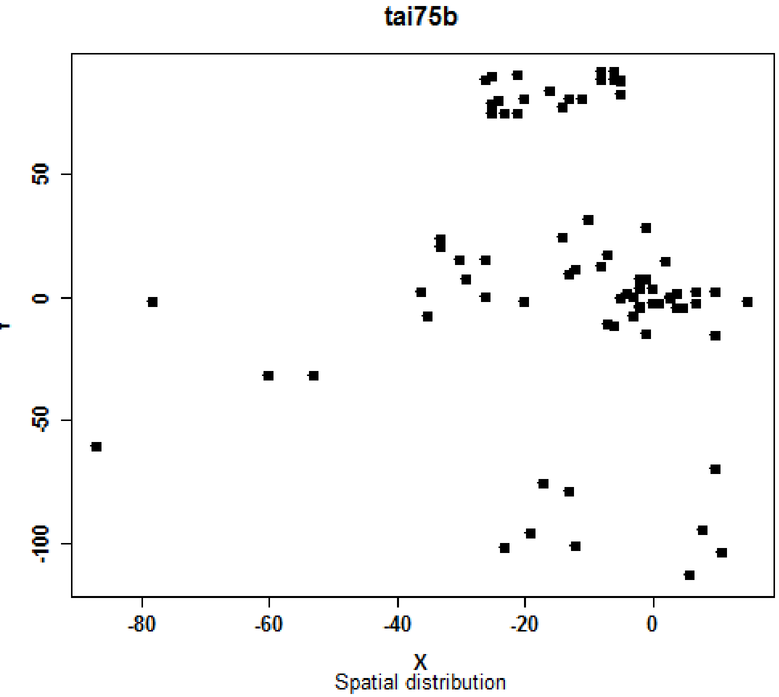

f134
Quantity
Class
Cluster factor: 660

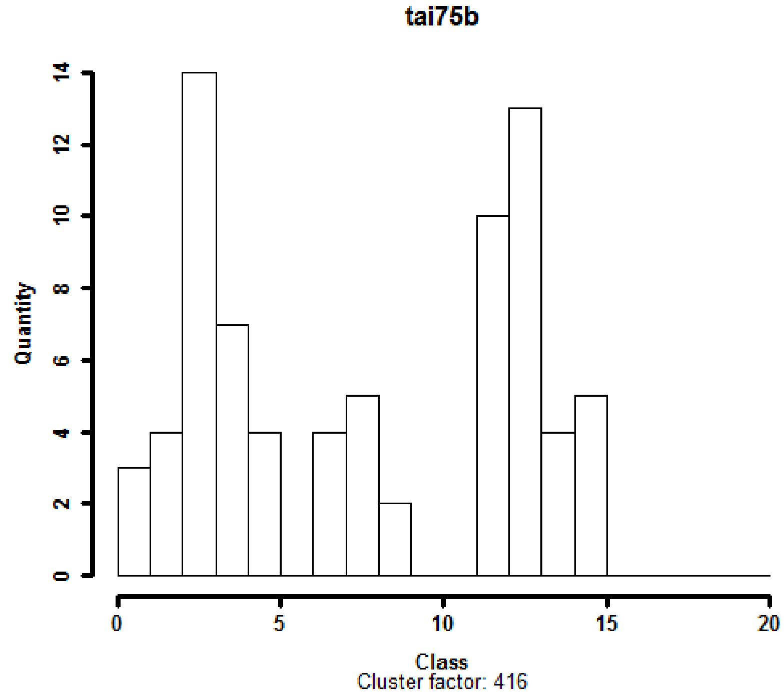

f134
x
y
Requests volume

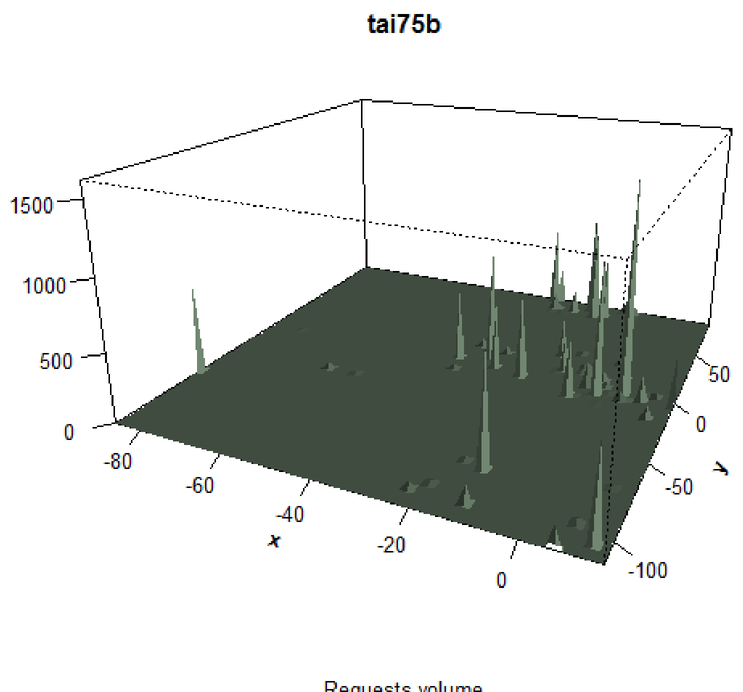

f134
Requests quantity
Request size
Skewness: 3.21

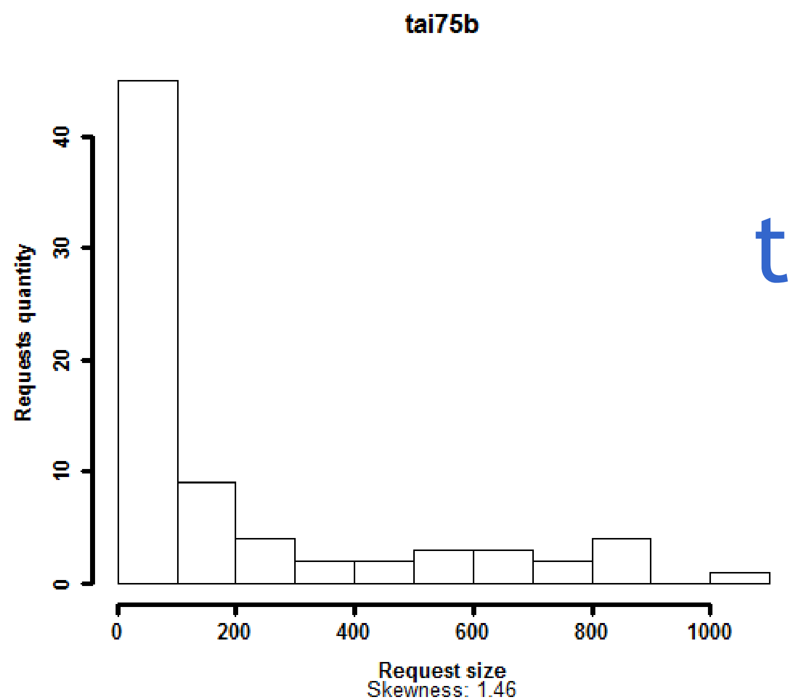
f134

tai75


tai75b
50
0
-50
-100
-80
-60
-40
-20
0
X
Spatial distribution


tai75b
Quantity
0
2
4
6
8
10
12
14
0
5
10
15
20
Class
Cluster factor: 416


tai75b
1500
1000
500
0
-80
-60
-40
-20
0
x
50
0
-50
-100
y
Requests volume


tai75b
Requests quantity
0
10
20
30
40
0
200
400
600
800
1000
Request size
Skewness: 1.46

Results

2MPSO – parameters and eval. fun.

8 swarms of 40 particles each
50 time slices

Phase 1: 50 iter./ts
Phase 2: 12 iter./ts

Results overview – 2MPSO vs. other methods

name	MAPSO	MEMSO	GA	TABU
c50	+	.	.	+
c75	+	+	+	+
c100	.	.	.	+
c100b	.	.	+	+
c120	+	+	+	+
c150	+	+	+	+
c199	+	+	+	+
f71	.	.	.	.
f134	+	+	+	+
tai75a	.	.	.	.
tai75b	-	-	.	+
tai75c	.	.	.	.
tai75d	-	.	.	.
tai100a	-	-	.	.
tai100b	.	.	.	+
tai100c	.	.	.	.
tai100d	+	+	+	+
tai150a	-	-	-	.
tai150b	-	-	.	+
tai150c	-	-	.	+
tai150d	-	-	.	.
sum	+	+	+	+

2MPSO vs. MAPSO/MEMSO - 10^6 evaluations

	MPSO$_2$		MPSO$_3$		MAPSO		MEMSO	
name	Min	Avg	Min	Avg	Min	Avg	Min	Avg
c50	**558,07**	599,73	573,17	**587,14**	571,34	610,67	577,6	592,95
c75	894,96	926,25	**888,66**	**914,82**	931,59	965,53	928,53	962,54
c100	918,79	961,47	**898,16**	**960**	953,79	973,01	949,83	968,92
c100b	827,85	861,92	**825,65**	**856,95**	866,42	882,39	864,19	878,81
c120	**1068,69**	**1110,24**	1076,72	1112,56	1223,49	1295,79	1164,63	1284,62
c150	1129,86	1185,4	**1123,1**	**1175,65**	1300,43	1357,71	1274,33	1327,24
c199	1389,48	1451,61	**1393,47**	**1432,69**	1595,97	1646,37	1600,57	1649,17
f71	280,78	298,59	**279,95**	298,71	287,51	296,76	283,43	**294,85**
f134	**11997,78**	12540,97	12054,18	**12501,29**	15150,5	16193	14814,1	16083,82
tai75a	**1739,35**	1870,45	1745,65	1882,97	1794,38	1849,37	1785,11	**1837**
tai75b	1450,61	1530,9	1428,16	1515,81	**1396,42**	1426,67	1398,68	**1425,8**
tai75c	**1465,14**	1569,98	1470,97	1536,12	1483,1	**1518,65**	1490,32	1532,45
tai75d	1434,41	1472,95	1428,45	1470,86	1391,99	**1413,83**	**1342,26**	1448,19
tai100a	2195,04	2302,25	2252,76	2301,82	2178,86	2214,61	**2170,54**	**2213,75**
tai100b	**2064,5**	2175,12	2068,86	**2168,1**	2140,57	2218,58	2093,54	2190,01
tai100c	1517,45	1593,97	1526,07	1598,64	**1490,4**	**1550,63**	1491,13	1553,55
tai100d	1746,34	**1819,1**	1753,79	1823,46	1838,75	1928,69	**1732,38**	1895,42
tai150a	3495,08	3724,91	3490,06	3628,22	3273,24	3389,97	**3253,77**	**3369,48**
tai150b	3072,9	3158,09	3017,35	3128,64	**2861,91**	**2956,84**	2865,17	2959,15
tai150c	2618,44	2769,2	2631,67	2727,57	2512,01	2671,35	**2510,13**	**2644,69**
tai150d	2996	3134,44	3023,47	3103,75	**2861,46**	**2989,24**	2872,8	3006,88
sum	**44861,52**	47057,53	44950,32	**46725,74**	48104,13	50349,66	47463,04	50119,29

2MPSO [2.2*10^7 evaluations] vs. MAPSO/MEMSO

	$MPSO_2$		$MPSO_3$		MAPSO		MEMSO	
name	Min	Avg	Min	Avg	Min	Avg	Min	Avg
c50	**571,53**	614,61	**571,53**	610,89	571,34	610,67	577,6	**592,95**
c75	**896,33**	**930,94**	906,72	933,4	931,59	965,53	928,53	962,54
c100	920,11	957,49	**918,61**	**953,13**	953,79	973,01	949,83	968,92
c100b	848,5	881,7	**823,19**	**871,07**	866,42	882,39	864,19	878,81
c120	**1057,94**	1174,65	1112,73	**1174,04**	1223,49	1295,79	1164,63	1284,62
c150	**1121,5**	**1168,59**	1150,13	1187,27	1300,43	1357,71	1274,33	1327,24
c199	**1404,46**	**1461,58**	1427,73	1487,29	1595,97	1646,37	1600,57	1649,17
f71	302,5	317,66	311,59	319,23	287,51	296,76	**283,43**	**294,85**
f134	**11988,76**	12324,98	12022,64	**12312,74**	15150,5	16193	14814,1	16083,82
tai75a	**1727,89**	**1812,55**	1760,27	1821,01	1794,38	1849,37	1785,11	1837
tai75b	1400,33	1438,5	1391,74	1426,39	**1396,42**	1426,67	1398,68	**1425,8**
tai75c	**1440,2**	**1491,64**	1446,85	1529,36	1483,1	1518,65	1490,32	1532,45
tai75d	1439,27	1470,93	1435,92	1465,55	1391,99	1413,83	**1342,26**	**1448,19**
tai100a	**2146,53**	2260,21	2169,24	2248,91	2178,86	2214,61	2170,54	**2213,75**
tai100b	2045,24	2119,37	**2039,31**	**2094,49**	2140,57	2218,58	2093,54	2190,01
tai100c	1469,12	1516,97	**1463,83**	**1516,35**	1490,4	1550,63	1491,13	1553,55
tai100d	**1685,53**	**1775,09**	1690,89	1778,74	1838,75	1928,69	1732,38	1895,42
tai150a	3345,88	3402,23	3319,48	3430,9	3273,24	3389,97	**3253,77**	**3369,48**
tai150b	2885,21	**2942,49**	2901,2	2973,39	**2861,91**	2956,84	2865,17	2959,15
tai150c	**2472,7**	2543,47	2483,39	**2525,04**	2512,01	2671,35	2510,13	2644,69
tai150d	**2844,7**	**2949,2**	2868,94	2956,74	2861,46	2989,24	2872,8	3006,88
sum	**44014,23**	**45554,84**	44215,93	45615,94	48104,13	50349,66	47463,04	50119,29

Summary of Results

- 2-phase PSO approach proved its efficiency.
- The 2MPSO method is on average better by 1.8% in the best routes and 1.9% in the averages than MEMSO (for the same number of function evaluations)
- It is possible to furher improve the results with the higher number of evaluations
- We were able to report 17 (out of 21) new best literature solutions for the given benchmark set with *cut-off time* set to 0.5

Swarm Robotics …