Warehouse allocation

Jakub Rada

3.1.2022

Problem description

- N warehouses, M customers
- set-up price s_w, capacity cap_w $\qquad \forall w = 1, \ldots, N$
- demand d_c $\qquad \forall c = 1, \ldots, M$
- delivery cost t_{cw} $\qquad \forall c = 1, \ldots, M \quad w = 1, \ldots, N$

$$
\begin{aligned}
\text{minimize} \quad & f = \sum_{w=1}^{N} \left(I(|a_w| > 0) \cdot s_w + \sum_{c \in a_w} t_{cw} \right) & \\
\text{subject to} \quad & \sum_{c \in a_w} d_c \leq cap_w & \forall w = 1, \ldots, N \\
& \sum_{w=1}^{N} I(c \in a_w) = 1 & \forall c = 1, \ldots, M
\end{aligned}
$$

Representation and fitness function

Candidate

1	2	$\cdots$	$M-1$	M
w_1	w_2	$\cdots$	w_{M-1}	w_M

Fitness function

- fitness f
- constraint violations $g = \sum_{w=1}^{N} \max(0, \sum_{c \in a_w} d_c - cap_w)$

1. $h = f + a \cdot g \qquad a \in \mathbb{R}$
2. c_1 is better than c_2 *if*
   $g(c_1) < g(c_2)$ or $(g(c_1) = g(c_2)$ and $f(c_1) < f(c_2))$

Local search

- standard local search with first-improving strategy

Candidate is invalid

Find assignment of warehouse with exceeded capacity and change it randomly. If it improved, keep it and find another.

Candidate is valid

Randomly change one assignment.

EA

- Population size: 50
- Selection: 5 parents using tournament selection
- Crossover: breed all pairs of parents with single point crossover
- Mutation: randomly change one assignment with probability 0.5 or randomly shuffle assignments with probability 0.3

Memetic algorithm

- Population size: 20
- Selection: 5 parents using tournament selection
- Crossover: breed all pairs of parents with uniform crossover
- Mutation: randomly change one assignment with probability 0.2 or randomly shuffle assignments with probability 0.6

Local search

After breeding phase, run 100 iterations of local search on each offspring with probability 0.3. In each iteration a mutation operator is selected with probability 0.5.
Mutation operators:

- randomly change n assignments
- find the most expensive assignment and replace it with cheapest

Experiments

- 8 instances

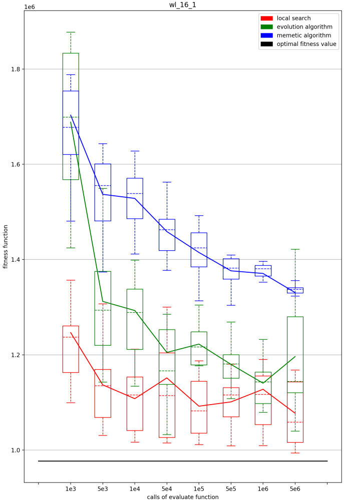
	customers	warehouses	optimal solution
wl_16_1	50	16	976738.625
wl_25_2	50	25	796648.438
wl_50_1	50	50	793439.562
wl_100_4	1000	100	17765201.949
wl_200_1	200	200	2686.479
wl_500_1	500	500	2608.148
wl_1000_1	1000	1000	5283.757
wl_2000_1	2000	2000	10069.803

- fitness function evaluations range from 1000 to 5000000
- each solved 11 times

Experiments

- valid solution found after 1000 evaluations every time
- memetic algorithm is significantly better on large instances, on smaller ones is slightly behind ls and ea
- ea is by far the slowest, ls fastest

wl_16_1

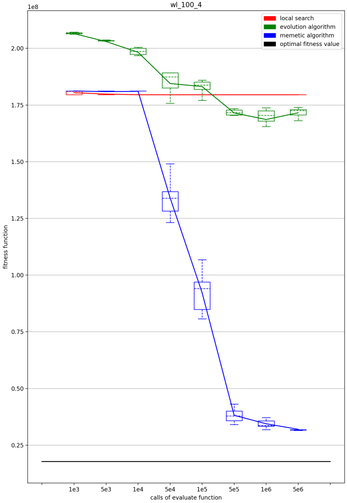


M	N	optimum
50	16	976738.625

algorithm	value	ratio
ls	993907.0	1.02
ea	997198.0	1.02
memetic	1266930.0	1.3

Table: Best found solutions

wl_100_4

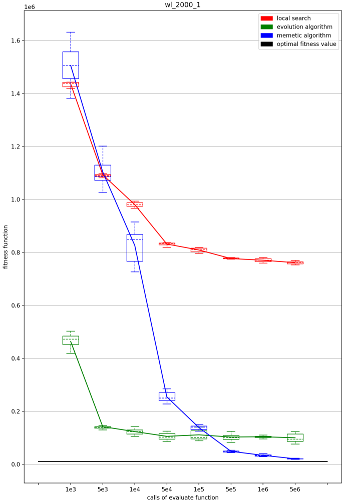


M	N	optimum
1000	100	17765201.949

algorithm	value	ratio
ls	1.80e8	10.11
ea	1.65e8	9.31
memetic	3.14e7	1.77

Table: Best found solutions

wl_2000_1



M	N	optimum
2000	2000	10069.803

algorithm	value	ratio
ls	7.53e5	74.78
ea	7.65e4	7.56
memetic	1.85e4	1.84

Table: Best found solutions