ВЕСТНИК ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
2020 Управление, вычислительная техника и информатика № 52

УДК 519.854.2
DOI: 10.17223/19988605/52/6

R.V. Gindullin

HYBRID OPTIMIZATION APPROACH BASED ON GRAVITATIONAL ANALOGY AND PARTICLE SWARM FOR SOLVING SINGLE VEHICLE PICKUP AND DELIVERY PROBLEM

The Pickup and delivery problem (PDP) with single vehicle (SPDP) with capacity constraints is considered. The problem requires constructing the shortest cyclic route for delivery of homogeneous cargo (e.g. passengers) from all the producers to specific customers with one capacitated vehicle. A heuristic procedure, based on the gravitational analogy, for finding approximate solution is developed. A particle swarm optimization procedure is combined with the developed heuristics. Efficiency of developed procedures is empirically analyzed and compared with each other.

Keywords: vehicle routing; optimization; metaheuristics.

The truck dispatching problem is the first example of vehicle routing problems (VRP) [1]. Many modifications of the VRP were examined since then. Information about VRP research is accumulating in [2].

In this article, we consider one of the variants of VRP is called the pickup and delivery problem with single vehicle (SPDP). The problem requires constructing the shortest cyclic route for delivery of homogeneous cargo (e.g. passengers) from all the producers to specific customers with one capacitated vehicle. This problem is classified as NP-hard. The alternative names of SPDP are Pickup-Delivery Traveling Salesman Problem and Traveling Salesman Problem with Pickups and Deliveries [3].

SPDP without vehicle capacity constraint and cargo weights was solved with exact approach in [4]. Example with 15 nods was solved with the combination of branch-and-cuts and greed search algorithms.

Authors of [5] and [6] applied different approaches with perturbations in cycles to solve SPDP without vehicle capacity constraint and cargo weights. Approaches were used on a group of 108 examples from TSPLIB, including the example with 441 nods.

Algorithms, based on real world analogies, were applied to various variants VRP. In particular, lightning inspired search algorithm (LISA) was proposed in [7]. In [8] it was adapted for the traveling salesman problem. Authors in [9] demonstrated the applicability of another method, called gravity-based algorithm for the multiple traveling salesmen problem.

1. Problem definition

Assume, that $P = \{1,\ldots,n\}$ is a set of nodes of cargo pickups and $D = \{1 + n, \ldots, 2n\}$ is a set of nodes of cargo deliveries. For the node i from P, the amount of cargo that needs to be picked up is equal q_i, while for the node $i + n$ from D the amount of cargo that needs to be delivered is equal to the negative amount $(-q_i)$. Set of all the nodes is denoted as $V = P \cup D \cup \{0\}$, where $\{0\}$ represents the depot. Distances between all pairs of nodes (c_{ij}) are known. A vehicle with the capacity S must visit all the nodes once and deliver cargo from the nodes i to the nodes $i + n$. A route must start and end in the depot. The goal of the problem is to find the shortest possible route. The problem is NP-hard, because if we assume $S = q_1 = \ldots = q_n = 1$, then the problem is the equivalent of the asymmetric traveling salesman problem with $n + 1$ nodes.

Minimal allowed vehicle capacity is equal to $\max\{q_i\}$. Clearly, no route could be made if $S < \max\{q_i\}$. If $S = \max\{q_i\}$, there are at least $n!$ allowed routes, for example: $0 - 1 - (1 + n) - 2 - (2 + n) - \ldots - n - 2n - 0$.

If vehicle capacity is unlimited, then the number of the allowed routes is equal $(2n)!/2n = (2n - 1)!$. Number of all permutations is equal $(2n)!$, and each allowed permutation leads to $2n$ permutations by switching places of nodes i and $i + n$ for all $i = 1, 2, \ldots, n$.

With $S \geq \max\{q_i\}$ any allowed section of a route could be continued. For example, if the vehicle is empty and there are unvisited clients, then the vehicle can move to some node i to pick up its cargo. Availability of some cargo in the vehicle, means, that some node i is visited, but the cargo is not yet delivered to the node $i + n$ and, as a result, the vehicle can deliver this cargo to the node $i + n$.

Problem formalizations are listed in [10].

2. Heuristic, based on the gravitational analogy

The proposed heuristic procedure is based on the greed algorithm in which the choice of the next destination depends on both a distance between nodes and the cargo size of the corresponding node.

Values of p_i, $i = \overline{1, 2n}$, denoting weights, are assigned to the vehicle and every node. These values change during the procedure. Initially, all producer weights are positive, and all consumer weights are equal to zero. Weights are normalized to [0, 1] range. Weight of the vehicle, p_0, is equal to the average weight of the producers multiplied on previously defined coefficient R.

$$p_0 = \frac{R}{n}\sum_{i=1}^{n} p_i, \;\; p_i = \frac{q_i}{q_{\max}}, \;\; p_{i+n} = 0, \;\; i = \overline{1,n}, \;\; q_{\max} = \max_{1 \leq i \leq n} q_i, \tag{1}$$

On every step of the procedure, the vehicle is experiencing the gravitational pull from nodes. F_{ji} denotes the force of the gravitational pull from the node i to the vehicle stationed in the node j. Vehicle moves to the node with the highest F_{ji}. After being visited by the vehicle, the weight of producer i, will become equal to 0 ($p_i = 0$), while the corresponding consumer will be assigned with the previous weight p_i ($p_{i+n} = p_i$). After being visited by the vehicle, the weight of consumer $i + n$ will become equal to 0 ($p_{i+n} = 0$).

Gravitational pull of a producer i, if the vehicle is placed in node j, is calculated as follows:

$$F_{ji} = \frac{p_0 p_i^s}{\left(c_{ji}^*\right)^t}, \; i = \overline{1,n}, \;\; c_{ji}^* = \frac{c_{ji}}{c_{\max}}, \;\; c_{\max} = \max_{i,j} c_{ij}, \tag{2}$$

Gravitational pull of a consumer $i + n$, if the vehicle is placed in node j, is calculated as follows:

$$F_{j,i+n} = \frac{p_{i+n}^{s+1}}{\left(c_{j,i+n}^*\right)^t}, \; i = \overline{1,n}, \;\; c_{j,i+n}^* = \frac{c_{j,i+n}}{c_{\max}}, \;\; c_{\max} = \max_{i,j} c_{ij}, \tag{3}$$

After visiting all the nodes, the vehicle returns to depot.

Values of s and t are selected before the start of the procedure.

Results of the heuristic will depend on the selected values of parameters R, s and t, that are assigned before the start of the procedure. Preliminary experiments demonstrated, that [–2, 2] is an efficient range of values for the parameter s.

The procedure has the same weakness as all greed algorithms: the distance between the depot and the last visited consumer could be large and this will significantly decrease the quality of the solution. To mitigate this potential problem, it is proposed to construct paths both from the start and from the end (forward and reverse searches, accordingly) for the selected values of R, s and t.

The search procedure with the set values of R, s, t and search direction will take time $O(n^2)$.

The procedure allows to construct an initial path for such procedures as 2-*opt*, 3-*opt*, 4-*opt*, etc. [5, 6].

The adjustment of values R, s, t and search direction could be performed by a simple enumeration procedure with set discretization within the selected ranges. The adoption of metaheuristics (genetic algorithm, particle swarm optimization is proposed, etc.) is also possible.

3. Hybrid approach

To show the efficiency of metaheuristics, compared to the enumeration approach, a variant of particle swarm optimization was adapted.

Function $f(R, s, t, d)$ denotes the result of the heuristic, based on the gravitational analogy. Parameter d denotes the search direction. If $d = -1$ then the forward search is used. If $d = 1$ then the reverse search is used.

N particles with different parameters R, s, t and d are generated. Each particle i will have its own value $f_i = f(R_i, s_i, t_i, d_i)$.

Between particles i and j the distance r_{ij} is calculated:

$$r_{ij} = r_{ji} = \left(R_i - R_j\right)^2 + \left(s_i - s_j\right)^2 + \left(t_i - t_j\right)^2 + \left(d_i - d_j\right)^2, \tag{4}$$

On every step k, particle i will change its position (R_i, s_i, t_i, d_i). The particle with the lowest f_i will move slightly in a random direction. Other particles will move to the particle j which has $\max_{j \neq i,\, f_i^k > f_j^k} \left(\left(f_i^k - f_j^k \right) e^{-\gamma r_{ij}^k} \right)$. Parameter γ is an input parameter which value is set before the procedure.

Coordinates of the particle i in step $k + 1$ after the move to the selected particle j will be calculated as follows:

$$R_i^{k+1} = R_i^k + \left(R_i^k - R_j^k\right) \cdot e^{-\gamma r_{ij}^k} + 1 \cdot \left(\text{rand} - \frac{1}{2}\right), \ R_i^{k+1} \geq 0.01, \tag{5}$$

$$s_i^{k+1} = \left\lfloor s_i^k + \left(s_i^k - s_j^k\right) \cdot e^{-\gamma r_{ij}^k} + 2 \cdot \left(\text{rand} - \frac{1}{2}\right) \right\rfloor, \tag{6}$$

$$t_i^{k+1} = \left\lfloor t_i^k + \left(t_i^k - R_j^k\right) \cdot e^{-\gamma r_{ij}^k} + 2 \cdot \left(\text{rand} - \frac{1}{2}\right) \right\rfloor, \ t_i^{k+1} \geq 1, \tag{7}$$

$$d_i^{k+1} = \begin{cases} -1, \ if \ d_i^k + \left(d_i^k - d_j^k\right) \cdot e^{-\gamma r_{ij}^k} + 1 \cdot \left(\text{rand} - \frac{1}{2}\right) < 0 \\ +1, \ if \ d_i^k + \left(d_i^k - d_j^k\right) \cdot e^{-\gamma r_{ij}^k} + 1 \cdot \left(\text{rand} - \frac{1}{2}\right) \geq 0 \end{cases}, \tag{8}$$

where rand is a randomly generated number from 0 to 1.

On each step k, the particle with the best solution is registered. The procedure is running until the chosen stopping rule is applied.

4. Computational experiments

To conduct numerical experiments, PC with Intel Core i5-4670 3.4GHz, 16Gb RAM and Windows 10 64bit was used. The hybrid procedure was realized in NetBeans IDE 8.1 (Java(TM) SE Runtime Environment 1.8.0_45-b14).

For comparative analysis of the models, standard library of symmetrical examples, TSPLIB [11], was used.

Nodes coordinates are taken accordingly to data from an example library. Hereafter:

– for the examples with even number of nodes, the last node is removed;
– for the examples with odd number of nodes, all nodes are used;
– first node is depot;
– the first half of remaining nodes are producers (i), and the other half are consumers ($i + n$).

For each of the selected examples three types of problems were considered:

Type 1: cargo weights of the first half of the producers were equal 1, cargo weights of the second half of the producers were equal 2, i.e. $q_i = 1$, $q_{i+n} = -1$ ($i < n/2$), $q_i = 2$, $q_{i+n} = -2$ ($i \geq n/2$). Vehicle capacity S was equal to 2, 6 and 10.

Type 2: cargo weights were equal 1, 2, 3, 4 and 5 cyclically. Vehicle capacity S was equal to 5, 10, 15 and 20.

Type 3: cargo weights were equal i, i.e. $q_i = i$, $q_{i+n} = -i$. Vehicle capacity S was equal $3n$.

For each of the problem two methods were used: enumeration of the values R, s, t and search direction for the heuristic based on the gravitational analogy (HBGA) and hybrid approach.

HBGA was performed two times with different discretization. First run of the enumeration (HBGA-1) was performed looking at R from 0.01 to 20 with step 0.01, s from -2 to 2 with step 1, t from 1 to 20 with step 1, both forward and reverse searches. Second run of the enumeration (HBGA-2) was performed looking at R from 0.1 to 20 with step 0.1, s from –2 to 2 with step 1, t from 1 to 20 with step 1, both forward and reverse searches.

In all experiments with the hybrid approach (HA), 20 steps were performed (k_{max} = 20). Number of particles N was equal 10, 20, 50, 100 and 200. Parameter γ was equal 1, 2 and 3. Hybrid procedure was performed once for each set of values N and γ.

Particles were generated as follows. Parameter R was taken equal to 0.01, 0.03, 0.09, 0.27, 0.81, 2.43, 7.29 and 21.87 cyclically. Parameter s was taken equal to -2, -1, 0, 1 and 2 cyclically. Parameter t was taken equal to 1, 4, 7, 10, 13, 16, 19, 22 and 25 cyclically. Parameter d was taken equal to –1 and 1 cyclically.

The difference between path lengths in different approaches is shown in table 1. The length of the path that was found by HBGA-1 for a problem was taken for 100% and other approaches were compared to it.

Table 1

Comparison of path lengths between algorithms

Example	Type	S	HBGA-2	HA, $N = 10$, $\gamma = 1$	HA, $N = 10$, $\gamma = 2$	HA, $N = 10$, $\gamma = 3$	HA, $N = 20$, $\gamma = 1$	HA, $N = 20$, $\gamma = 2$	HA, $N = 20$, $\gamma = 3$	HA, $N = 50$, $\gamma = 1$	HA, $N = 50$, $\gamma = 2$	HA, $N = 50$, $\gamma = 3$	HA, $N = 100$, $\gamma = 1$	HA, $N = 100$, $\gamma = 2$	HA, $N = 100$, $\gamma = 3$	HA, $N = 200$, $\gamma = 1$	HA, $N = 200$, $\gamma = 2$	HA, $N = 200$, $\gamma = 3$
rat99	1	4	0,0%	0,0%	–0,1%	1,6%	0,5%	–0,1%	0,0%	0,0%	0,0%	0,0%	–0,1%	0,7%	0,6%	–0,1%	–0,1%	0,0%
		6	1,2%	1,8%	0,3%	3,2%	1,8%	1,2%	0,0%	1,3%	1,3%	1,1%	0,3%	1,3%	0,1%	1,1%	1,2%	1,2%
		10	0,2%	0,2%	1,3%	0,2%	0,2%	0,0%	0,2%	–0,6%	0,2%	0,2%	0,2%	0,2%	0,2%	–0,5%	–0,5%	–0,5%
	2	5	0,0%	0,8%	1,0%	1,0%	0,8%	0,8%	0,8%	0,8%	0,3%	0,8%	0,0%	0,8%	0,0%	0,0%	0,0%	0,0%
		10	0,0%	0,0%	1,5%	0,1%	0,4%	0,0%	1,2%	0,6%	0,4%	0,4%	0,1%	0,0%	0,4%	0,1%	0,3%	0,3%
		15	0,0%	0,1%	1,6%	1,6%	0,0%	0,1%	0,0%	0,7%	0,0%	0,1%	0,0%	0,1%	0,0%	0,0%	0,0%	0,0%
		20	0,0%	2,0%	2,0%	0,0%	0,5%	1,1%	2,0%	0,5%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
	3	3*n	0,6%	0,9%	5,8%	0,6%	3,3%	0,6%	2,5%	0,6%	0,6%	1,3%	0,0%	0,6%	0,0%	0,6%	0,6%	1,0%
gr137	1	4	0,0%	0,1%	0,4%	0,1%	0,0%	0,0%	0,1%	0,0%	0,1%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
		6	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
		10	0,0%	3,0%	1,4%	2,5%	2,5%	3,0%	0,7%	0,7%	0,4%	1,4%	0,5%	0,6%	0,0%	0,6%	0,6%	0,0%
	2	5	0,0%	0,2%	0,3%	0,2%	0,0%	–0,1%	0,0%	–0,3%	0,0%	–0,3%	–0,1%	0,0%	0,0%	–0,3%	–0,3%	–0,3%
		10	0,0%	1,2%	0,4%	0,0%	0,0%	0,4%	0,2%	0,0%	1,2%	0,2%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
		15	0,2%	0,2%	0,3%	0,7%	–0,7%	–0,7%	0,2%	–0,7%	0,2%	–0,7%	0,2%	–0,7%	–0,7%	–0,7%	–0,7%	–0,7%
		20	1,7%	1,7%	1,7%	3,6%	2,4%	3,6%	2,4%	1,7%	1,7%	1,7%	0,8%	1,7%	0,0%	0,0%	0,0%	1,6%
	3	3*n	0,3%	2,9%	2,1%	0,7%	0,3%	0,3%	0,3%	1,1%	1,1%	0,3%	0,3%	0,3%	0,3%	0,3%	0,3%	0,3%
gr229	1	4	0,0%	0,1%	0,7%	0,6%	0,2%	0,3%	0,4%	0,1%	0,1%	–0,1%	0,4%	0,2%	0,2%	0,2%	0,1%	0,1%
		6	0,0%	1,0%	0,1%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
		10	0,0%	1,8%	1,8%	2,0%	2,5%	0,8%	0,8%	0,0%	2,0%	1,0%	0,9%	0,0%	0,0%	0,1%	0,1%	0,0%
	2	5	0,0%	0,8%	0,4%	4,5%	0,7%	0,0%	0,4%	0,7%	0,6%	0,0%	0,0%	0,4%	0,0%	0,4%	0,0%	0,4%
		10	0,0%	1,9%	2,1%	2,2%	1,5%	1,5%	1,9%	0,3%	1,4%	1,5%	1,0%	0,8%	1,4%	0,8%	1,1%	0,6%
		15	0,0%	0,2%	0,2%	0,2%	0,0%	0,0%	0,0%	0,1%	0,0%	0,1%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
		20	0,0%	4,2%	4,1%	4,2%	4,1%	4,1%	4,2%	2,7%	0,0%	3,7%	3,7%	2,6%	3,7%	3,7%	0,1%	2,7%
	3	3*n	0,0%	0,2%	1,0%	0,2%	0,1%	0,9%	0,0%	0,2%	0,2%	0,0%	0,0%	0,2%	0,2%	0,0%	0,0%	0,0%
rd400	1	4	0,0%	3,7%	2,5%	2,5%	2,5%	3,6%	2,2%	2,4%	1,4%	0,0%	2,0%	1,4%	2,4%	1,4%	0,0%	1,4%
		6	0,7%	3,2%	1,3%	1,0%	2,9%	3,4%	2,9%	–0,8%	–0,8%	2,1%	1,3%	–0,8%	–0,6%	–0,8%	–0,8%	–0,8%
		10	0,0%	3,4%	2,2%	0,9%	2,2%	2,6%	0,7%	1,5%	–0,2%	0,9%	1,5%	0,7%	0,7%	–0,2%	0,7%	–0,2%
	2	5	0,0%	3,2%	3,2%	3,2%	3,2%	4,6%	4,8%	3,1%	3,1%	3,2%	3,2%	3,1%	3,1%	2,4%	2,4%	3,1%
		10	2,9%	7,4%	7,4%	7,0%	6,4%	6,4%	6,4%	6,3%	6,4%	3,1%	5,6%	4,9%	5,1%	3,9%	3,1%	6,2%
		15	0,7%	3,9%	5,1%	5,0%	2,8%	3,6%	2,8%	2,7%	2,5%	2,3%	1,6%	1,9%	2,0%	2,1%	0,9%	2,1%
		20	0,2%	6,1%	6,4%	2,4%	2,6%	4,0%	1,2%	1,2%	2,4%	2,4%	1,4%	1,7%	1,7%	1,2%	1,7%	0,9%
	3	3*n	0,0%	4,4%	5,0%	4,5%	4,4%	4,8%	4,1%	4,9%	5,3%	4,1%	4,1%	3,1%	4,1%	2,7%	3,4%	1,6%
pa561	1	4	0,9%	1,5%	2,1%	2,1%	1,6%	2,5%	1,7%	1,0%	1,2%	1,2%	1,7%	0,6%	1,2%	0,0%	0,7%	0,7%
		6	0,0%	3,1%	3,1%	4,8%	4,0%	3,6%	4,2%	0,3%	1,9%	2,8%	2,5%	2,8%	0,9%	0,0%	0,6%	0,1%
		10	0,0%	2,1%	4,2%	2,5%	1,3%	3,8%	0,0%	1,6%	–1,1%	0,8%	0,2%	–0,7%	0,8%	–0,7%	–0,8%	–2,0%
	2	5	0,0%	1,8%	0,7%	3,0%	2,4%	3,3%	1,4%	1,1%	0,8%	0,4%	0,3%	0,5%	0,3%	0,3%	0,5%	0,5%
		10	0,0%	5,3%	6,2%	6,6%	1,3%	3,2%	0,9%	2,1%	0,9%	1,3%	3,3%	0,4%	1,4%	0,7%	0,4%	0,1%
		15	0,0%	–0,4%	1,4%	2,8%	2,0%	1,2%	1,2%	1,2%	1,2%	0,0%	0,5%	0,1%	1,0%	0,0%	0,1%	0,4%
		20	0,0%	2,6%	1,3%	0,1%	0,1%	0,1%	1,3%	1,4%	0,1%	0,0%	0,1%	0,0%	0,0%	0,0%	0,0%	0,0%
	3	3*n	0,0%	3,0%	3,0%	2,8%	3,1%	3,6%	3,8%	2,5%	0,2%	2,7%	3,0%	2,5%	0,2%	0,2%	2,5%	1,8%

In most examples, HBGA-2 showed same result as HBGA-1. In 30% of examples HBGA-2 showed result worse, than HBGA-1, up to +5% compared to the initial length.

The quality of the solutions in hybrid approaches heavily depends on the number of particles N and almost unaffected by γ. Parameter $\gamma = 3$, on average, gave slightly better results than $\gamma = 1$ and $\gamma = 2$, but the difference is too small to make a conclusive statement.

For $N = 10$, an average solution was +2% longer than HBGA-1. For $N = 50$, an average solution was +1% longer than HBGA-1. For $N = 200$, an average solution was +0.5% longer than HBGA-1. It is worth noting, that, for $N = 200$, in 12–15% examples hybrid approach resulted with the path length shorter than the path length of HBGA-1.

Comparison of the calculation time is shown in table 2. Since there was no significant difference between results for various values of γ for HA, only $\gamma = 1$ is presented in the table 2.

Table 2

Comparison of computational time between algorithms

Example	Type	*S*	HBGA-1	HBGA-2	HA, N = 10, $\gamma = 1$	HA, N = 20, $\gamma = 1$	HA, N = 50, $\gamma = 1$	HA, N = 100, $\gamma = 1$	HA, N = 200, $\gamma = 1$
rat99	1	4	21,92	2,76	0,03	0,06	0,15	0,30	0,65
		6	22,29	3,07	0,03	0,07	0,16	0,33	1,20
		10	23,58	3,12	0,04	0,09	0,21	0,53	0,90
	2	5	21,95	2,66	0,05	0,08	0,27	0,43	0,82
		10	23,44	2,96	0,03	0,06	0,17	0,45	0,78
		15	23,95	2,96	0,03	0,07	0,17	0,59	0,88
		20	24,27	2,93	0,04	0,12	0,29	0,58	0,81
	3	3**n*	23,09	2,96	0,03	0,06	0,17	0,34	0,95
gr137	1	4	41,78	5,41	0,13	0,19	0,29	0,75	1,58
		6	43,11	6,43	0,06	0,16	0,39	0,76	1,42
		10	44,65	6,86	0,06	0,12	0,37	0,62	1,30
	2	5	40,64	5,44	0,06	0,11	0,29	0,62	1,17
		10	43,33	6,14	0,06	0,12	0,30	0,64	1,24
		15	44,54	6,48	0,06	0,13	0,30	0,60	1,26
		20	44,19	6,24	0,06	0,14	0,30	0,61	1,25
	3	3**n*	43,20	6,33	0,06	0,12	0,29	0,61	1,26
gr229	1	4	117,66	17,23	0,17	0,34	0,93	1,85	3,53
		6	116,77	16,93	0,16	0,32	0,81	1,73	3,34
		10	116,90	15,92	0,16	0,32	0,96	1,82	3,76
	2	5	108,85	15,28	0,17	0,32	0,79	1,60	3,47
		10	111,82	16,35	0,17	0,31	0,85	1,69	3,81
		15	117,01	15,66	0,19	0,36	0,93	1,82	3,67
		20	125,66	17,39	0,18	0,34	0,85	1,78	3,77
	3	3**n*	125,86	16,82	0,19	0,45	0,93	1,89	3,65
rd400	1	4	367,59	51,98	0,61	1,05	2,67	6,23	13,93
		6	342,25	50,72	0,70	1,16	2,95	7,12	15,43
		10	380,01	51,32	0,50	1,04	2,87	5,77	10,47
	2	5	395,55	51,90	0,45	0,96	2,73	5,35	10,54
		10	343,70	45,34	0,47	1,08	2,56	5,08	14,50
		15	351,94	44,01	0,56	1,29	2,80	7,79	12,97
		20	352,15	67,50	0,43	0,84	2,12	4,37	12,49
	3	3**n*	348,30	63,12	0,54	1,28	3,02	6,23	11,41
pa561	1	4	735,16	87,14	1,17	2,24	7,26	12,99	27,03
		6	742,46	96,85	1,19	2,25	6,08	12,79	25,13
		10	753,99	97,82	1,19	2,70	7,13	13,47	27,73
	2	5	790,23	97,92	1,09	2,54	6,18	14,63	32,16
		10	766,68	96,35	1,20	2,39	7,04	14,77	30,12
		15	719,73	115,84	1,21	2,88	8,38	13,55	30,92
		20	810,83	103,39	1,23	2,58	6,67	14,92	26,39
	3	3**n*	729,35	112,16	1,34	2,51	6,21	12,57	25,13

Computational time of the hybrid approach changes linearly, depending on the number of particles N, and quadratically, depending on the number of nodes n. With the highest number of particles, $N = 200$, computational time of HA was 3–4 times lower than the computational time of HBGA-2 and 20–30 times lower than the computational time of HBGA-1.

Conclusion

The hybrid heuristic approach combining particle swarm optimization and the heuristic, based on the gravitational analogy, for the single vehicle pickup and delivery problem is proposed. The particle swarm optimization is used to estimate optimal parameters for the greed search, based on gravitational analogy. The proposed variant of greed search uses both distances and cargo weights to choose the next destination node.

A series of computational experiments was conducted to evaluate properties of the proposed procedure. Computational experiments showed that the hybrid approach is significantly faster than enumeration approach. Both the quality of the solution obtained by the hybrid approach and the computational time required for the approach depend solely on the number of particles used. The proposed approach gives shorter route lengths, than the approach of enumeration of parameters with low discretization, and is faster to perform. The enumeration of parameters with high discretization gives shorter paths in most cases, but the overall difference is not high enough to justify the performance costs, compared to the proposed hybrid approach.

The proposed hybrid approach could be used to find an initial solution for other algorithms (e.g., various variants of k-opt).

Source code for the proposed approach is available in [12].

REFERENCES

1. Danzig, G. & Ramser, J. (1959) The Truck Dispatching Problem. *Management Science*. 6(6). pp. 80–91. DOI: 10.1287/mnsc.6.1.80
2. NEO Research Group. (n.d.) *Vehicle Routing Problem*. [Online] Available from: http://neo.lcc.uma.es/vrp/ (Accessed: 13th February 2018).
3. Parragh, S., Doerner, K. & Hartl, R. (2008) A survey on pickup and delivery problems. Part II: Transportations between customers and depot. *Journal fur Betriebswirtschaft*. 58. pp. 21–51.
4. Ruland, K.S. & Rodin, E.Y. (1997) The pickup and delivery problem: Faces and branch-and-cut algorithm. *Computers and Mathematics with Applications*. 33(12). pp. 1–13. DOI: 10.1016/S0898-1221(97)00090-4
5. Renaud, J., Boctor, F.F. & Ouenniche, J. (2000) A heuristic for the pickup and delivery traveling salesman problem. *Computers and Operations Research*. 27(9). pp. 905–916. DOI: 10.1016/S0305-0548(99)00066-0
6. Renaud, J., Boctor, F.F. & Laporte, G. (2002) Perturbation heuristics for the pickup and delivery traveling salesman problem. *Computers and Operations Research*. 29(9). pp. 1129–1141. DOI: 10.1016/S0305-0548(00)00109-X
7. Shareef, H., Ibrahim, A.A. & Mutlag, A. H. (2015) Lightning search algorithm. *Applied Soft Computing*. 36. pp. 315–333. DOI: 10.1016/j.asoc.2015.07.028
8. El Majdouli, M.A. & El Imrani, A.A. (2015) Lightning inspired search algorithm: Introduction & application to the traveling salesman problem. *10th Int. Conf. on Intelligent Systems: Theories and Applications (SITA)*, Rabat. pp. 165–171. DOI: 10.1109/SITA.2015.7358401
9. Rostami, A.S., Mohanna, F., Keshavarz, H. & Hosseinabadi, A.A.R. (2015) Solving Multiple Traveling Salesman Problem using the Gravitational Emulation Local Search Algorithm. *Applied Mathematics & Information Sciences*. 2. pp. 699–709. DOI: 10.12785/amis/090218
10. Bronshtein, E.M., Gindullina, E.V. & Gindullin, R.V. (2017) Formalization of pickup and delivery problem. *Vestnik Yuzhno-Ural'skogo universiteta. Seriya: Matematika. Mekhanika. Fizika*. 9(1). pp. 13–21. DOI: 10.14529/mmph170102
11. *MP-TESTDATA-The TSPLIB Symmetric Traveling Salesman Problem Instances*. [Online] Available from: http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/ (Accessed: 10th February 2019)
12. *PDP-Algorithms*. [Online] Available from: https://github.com/RamizGindullin/PDP-Algorithms (Accessed: 26th May 2020).

Received: September 10, 2019

Gindullin R.V. (2020) HYBRID OPTIMIZATION APPROACH BASED ON GRAVITATIONAL ANALOGY AND PARTICLE SWARM FOR SOLVING SINGLE VEHICLE PICKUP AND DELIVERY PROBLEM. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitelnaja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 52. pp. 52–58

DOI: 10.17223/19988605/52/6

Гиндуллин Р.В. ГИБРИДНЫЙ АЛГОРИТМ ОПТИМИЗАЦИИ, ОСНОВАННЫЙ НА ГРАВИТАЦИОННОЙ АНАЛОГИИ И АЛГОРИТМЕ РОЯ ЧАСТИЦ, ДЛЯ РЕШЕНИЯ ЗАДАЧИ О ЗАБОРЕ И ДОСТАВКЕ ГРУЗА ОДНИМ ТРАНСПОРТНЫМ СРЕДСТВОМ. *Вестник Томского государственного университета. Управление, вычислительная техника и информатика.* 2019. № 52. С. 52–58

Рассмотрена задача о заборе и доставке груза одним транспортным средством (The Pickup and delivery problem with single vehicle, SPDP) ограниченной вместимости. Предложена эвристика, основанная на гравитационной аналогии. Данная эвристика является расширением жадного поиска, в ней выбор следующего пункта зависит от расстояния и веса грузов. Длина найденного пути зависит от заранее выбранных параметров R, s и t. Предварительные эксперименты показали, что для параметра s достаточно рассматривать целые значения между –2 и 2. Процедура обладает недостатком, присущим всем жадным алгоритмам: расстояние между депо и последним посещенным пунктом может оказаться значительным, что приведет к ухудшению качества решения. Для сглаживания данной проблемы предложен вариант процедуры, в которой путь строится с конца при заданных R, s и t (прямой и обратный поиски). Процедура поиска при заданных параметрах занимает время $O(n^2)$.

Разработан гибридный подход, который объединяет метод роя частиц с предложенной эвристикой. На первом шаге генерируется N частиц с различными значениями параметров R, s, t and d (обозначает направление поиска). На каждом последующем шаге они изменяют значения параметров в зависимости от длин путей, которые получаются при применении значений параметров частицы и ее соседей к эвристике, основанной на гравитационной аналогии. Процедура продолжается, пока не будет выполнено правило останова.

Результаты вычислительных экспериментов показали, что качество решений, получаемых гибридным подходом, зависит от количества частиц N. При $N = 200$ средняя длина пути оказалась на +0,5% длиннее пути, получаемого перебором значений параметров эвристики, основанной на гравитационной аналогии. В 12–15% примеров гибридный подход дал меньшую длину пути, чем длина пути, получаемая перебором значений параметров эвристики, основанной на гравитационной аналогии.

При наибольшем количестве частиц, $N = 200$, затраты времени гибридного подхода ниже до 30 раз в сравнении с перебором значений параметров эвристики, основанной на гравитационной аналогии.

Ключевые слова: транспортная маршрутизация; оптимизация; метаэвристики.

GINDULLIN Ramiz Vilevich (Candidate of Physics and Mathematics, Associate Professor, Bashkir State University, Ufa, Russian Federation).
E-mail: ramizgindullin@gmail.com