Journal of Physics: Conference Series **681** (2016) 012050 doi:10.1088/1742-6596/681/1/012050

Improving the efficiency of solving discrete optimization problems: The case of VRP

A Belov and S Slastnikov

National Research University "Higher School of Economics", Moscow, Russia

E-mail: sslastnikov@gmail.com

Abstract. Paper is devoted constructing efficient metaheuristics algorithms for discrete optimization problems. Particularly, we consider vehicle routing problem applying original ant colony optimization method to solve it. Besides, some parts of algorithm are separated for parallel computing. Some experimental results are performed to compare the efficiency of these methods.

1. Introduction

Optimization algorithms are known to be useful in many types of problems in mathematics, physics, chemistry, biology, statistics and so on. Particularly, many physical systems are governed by minimization principles. For example, in thermodynamics, a system coupled to a heat bath always takes the state with minimal free energy. Besides, some other applications are well known: determination of the self affine properties of polymers in random media, solution of the protein folding problem, analysis of X-ray data, optimization of lasers and optical fibers, etc. [1].

In general discrete optimization is a branch of mathematical programming, which includes the extremal problems with variables defined on discrete sets. Almost all discrete optimization problems have application in practice. A large number of discrete optimization problems in one form or another can be reduced to a vehicle routing problem. Therefore, the algorithms and principles for solving vehicle routing problems are widely used for solving other discrete optimization problems.

Consider vehicle routing problem with capacity constraints (CVRP). Given a graph $G = (V, A, d)$, where $V = \{ v_0, v_1,..,v_n\}$ is the set of vertices (v_0 is a depot and other vertices are clients). A is a set of edges connecting vertices of the graph G. For each arc (i,j) a non-negative distance d_{ij} between clients v_i and v_j is given . Positive demand c_i is set for client i, capacity of each of the m vehicles is bounded by C_k ($k=1,...,m$). Also the following restrictions are given:

- each client should be visited exactly once;
- the depot is the start and the end point of all the routes.

The aim of the task is the construction of route with minimal total distance which satisfies the demands of all customers and does not violate the restrictions described above.

In section 2 mathematical setting of CVRP is performed. Section 3 is devoted to the original modification of ant colony optimization algorithm for CVRP. The algorithm parallelization scheme and some computational results of its application to the model tasks are presented in section 4. Finally, in section 5 these results are discussed.

2. Mathematical setting

The mathematical formalization of CVRP is the following: to minimize the objective function

$$F = \sum_{k=1}^{m}\sum_{i=0}^{n}\sum_{j=0}^{n} d_{ij} X_{ij}^{k} \to \min \tag{1}$$

under constraints

$$\sum_{i=1}^{n} c_i \sum_{j=0}^{n} X_{ij}^{k} \le C_k, \quad \forall k = 1,\dots,m, \tag{2}$$

$$\sum_{j=1}^{n} X_{0j}^{k} \le 1, \ \ \forall k = 1,\dots,m, \tag{3}$$

$$\sum_{i=1}^{n} X_{i0}^{k} \le 1, \ \ \forall k = 1,\dots,m, \tag{4}$$

$$\sum_{k=1}^{m}\sum_{j=0}^{n} X_{ij}^{k} = 1, \ \ \forall i = 1,\dots,n, \tag{5}$$

$$\sum_{i=0}^{n} X_{ih}^{k} - \sum_{j=0}^{n} X_{hj}^{k} = 0, \ \ \forall h = 1,\dots,n; \forall k = 1,\dots,m, \tag{6}$$

$$X_{ij}^{k} \in \{0,1\}, \quad \forall i = 0,\dots n; \ \ j = 0,\dots,n; \ \ \forall k = 1,\dots,m, \tag{7}$$

$$X_{ii}^{k} = 0 \quad \forall i = 0,\dots,n; \ \ \forall k = 1,\dots,m, \tag{8}$$

if $S_k = \{(i,j) : X_{ij}^{k} = 1\}$, then

$$\forall k = 1,\dots,m \quad \forall (i,j) \in S_k \ne \varnothing \ \ \exists (j_p, j_{p+1}) \in S_k \ \ (p = 0,\dots,l): \ \ j_0 = j_{l+1} = 0, \ \ (i,j) = (j_r, j_{r+1}), \ \ r \le l \,. \tag{9}$$

X_{ij}^{k} equals 1 if the vehicle *k* follows the customer *i* to customer *j*, and 0 otherwise.

The objective function (1) defines the total distance value passed by all vehicles. Conditions (2)–(9) specify the problem constraints and are discussed by many authors (see, for example, [2]).

CVRP is NP-hard problem, so only metaheuristics are suitable to solve it for practical applications. In the next section we consider an effective optimization algorithm based on the modification of ant colony method.

3. Modified ant colony optimization method

Ant colony optimization (ACO) is a swarm intelligence method for solving different hard discrete optimization problems. The approach is to use the model of food searching in colonies of ants that mark the path by a special chemical substance called pheromone. Left traces attract other ants, which, passing along marked paths, in turn enhances the smell of the pheromone. Over the time pheromone evaporates, so the ants could adapt their behavior to the external environment changes. Classical ACO for CVRP could be found in papers [2, 3].

Here we consider the following modification of ACO for CVRP. Initially, each vehicle *k* starts from the depot, and the set M_k of customers, included in its route, is empty. The next customer *j* to visit is chosen by the probabilistic criterion:

$$j = \begin{cases} \arg\max\limits_{u \notin M_k} [\tau_{iu} (d_{iu})^{-\beta}], & \text{with probability } \ p_0, \\ S, & \text{with probability } 1 - p_0, \end{cases} \tag{10}$$

where i is the current client (or depot on the first step); τ_{iu} is pheromone amount on the route between clients i and u; β is the parameter establishing the importance of distance in comparison to pheromone quantity; p_0 is the parameter. S is a discrete random variable with probability distribution $p_k(i,j)$, where $p_k(i,j)$ is the probability for the vehicle k to move from client i to client j. First part of formula (10) we call the "determined rule" for client choosing and the second – "stochastic rule". On the first step (iteration)

$$p_k(i,j)=\begin{cases}\tau_{ij}\dfrac{\tau_{iu}}{d_{iu}{}^{\beta}}\Big/\sum\limits_{v\notin M_k}\tau_{iv}\dfrac{\tau_{iu}}{d_{iu}{}^{\beta}}, & \text{if } j\notin M_k,\\ 0, & \text{if } j\in M_k,\end{cases} \tag{11}$$

and on the next steps is proposed to remember the total distances of best and worst routes of one vehicle (L and R, respectively). In the case, when moving from customer i to customer j is contained in the worst route, then the probability

$$p_k(i,j)=\frac{L}{R}\cdot\frac{\tau_{ij}}{d_{ij}{}^{\beta}}\Big/\left(\sum_{u\neq j,u\notin M_k}\frac{\tau_{iu}}{d_{iu}{}^{\beta}}+\frac{L}{R}\cdot\frac{\tau_{ij}}{d_{ij}{}^{\beta}}\right) \tag{12}$$

and in other cases

$$p_k(i,j)=\begin{cases}\dfrac{\tau_{ij}}{d_{ij}{}^{\beta}}\Big/\left(\sum\limits_{u\neq j,\,u\notin M_k}\dfrac{\tau_{iu}}{d_{iu}{}^{\beta}}+\dfrac{L}{R}\cdot\dfrac{\tau_{ij}}{d_{ij}{}^{\beta}}\right), & \text{if } j\notin M_k,\\ 0, & \text{if } j\in M_k.\end{cases} \tag{13}$$

Route for vehicle is completed when its load capacity is over or all clients are visited, then vehicle returns to the depot. Thus routes for all vehicles are consecutively constructed.

In order to improve future solutions, the pheromone trails should be updated to reflect the quality of the solutions found. Local pheromone update models its natural evaporation and ensures that no route becomes too prevalent. This update is made after route for current vehicle is done:

$$\tau_{ij}{}^{new}=(1-\alpha)\tau_{ij}{}^{old}+\alpha\tau_0, \tag{14}$$

where α is a parameter that controls the speed of evaporation, and τ_0 is the initial pheromone value. We propose to calculate the initial pheromone value as follows:

$$\tau_0=\frac{1}{(n+1)\cdot\min\limits_{i\neq j}d_{ij}}. \tag{15}$$

Global trail updating is performed by adding pheromone to all of the arcs included in the best route of one vehicle and encourages the use of shorter routes in future solutions:

$$\tau_{ij}{}^{new}=(1-\alpha)\tau_{ij}{}^{old}+\frac{\alpha}{L}. \tag{16}$$

This process is repeated until the stopping criterion occurs. The best solution over all iterations is a 'good' approximation of the optimal objective function value.

The efficiency of this modified ACO algorithm was studied and discussed in [2].

4. Algorithm parallelization

In order to reduce the ACO execution time we divided some parts of algorithm into several parallel processes for calculation. On the block diagram presented on the Figure 1 these parts are highlighted by dashed lines.

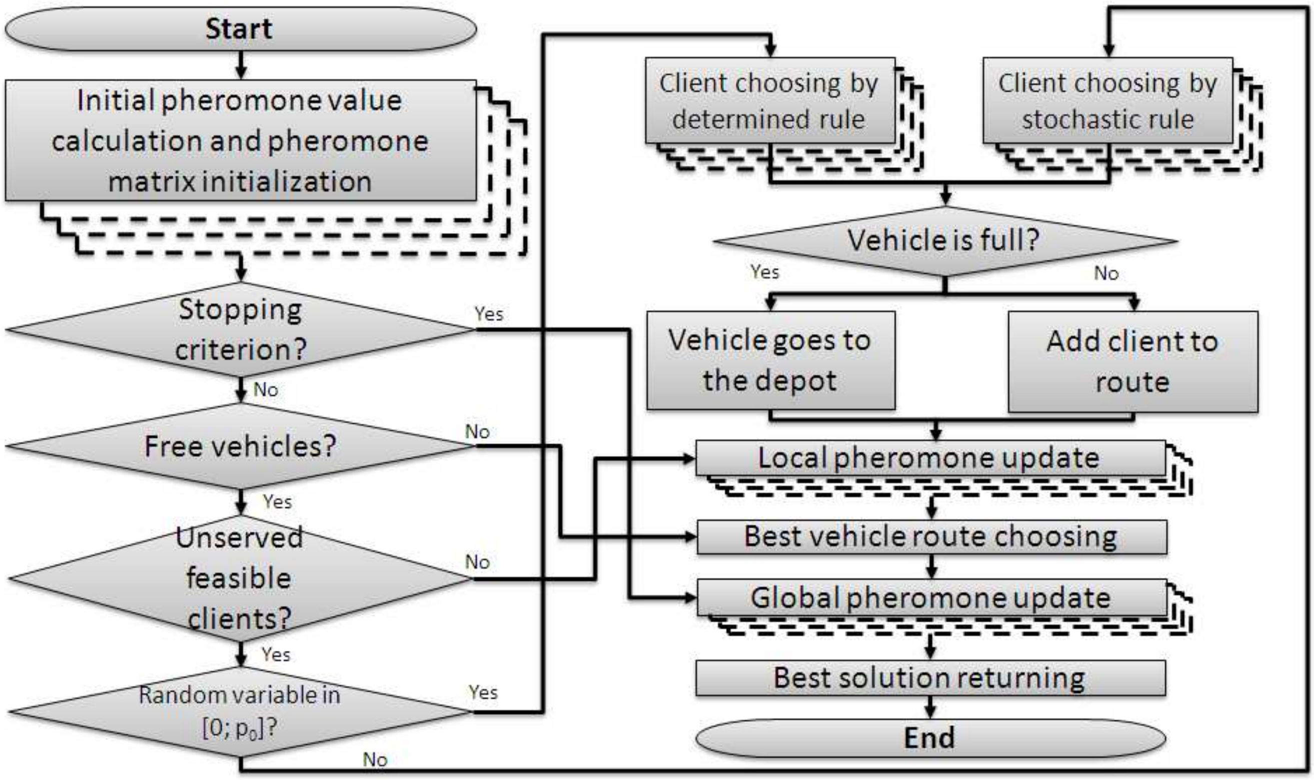

Figure 1. Block diagram of ACO algorithm with parallelization.

The modified ACO algorithms with and without parallel sections were implemented in C++ as dynamic shared object library. The library was written using the precise calculations library GNU Multi-Precision. Therefore, it becomes possible to reduce computing inaccuracies to a minimum without the computation time increasing.

As tools of parallel programming we used *OpenMP* and *BoostThread*. While the main advantage of *OpenMP* is in easy way of development and using, the main lack is in the absence of general support for working with arbitrary containers and custom data types. *BoostThread* is part of libraries designed specifically for the C++ language and provides opportunities for threads creation, management and synchronization. However, from the practical implementation side *BoostThread* has more complicated structure compared with *OpenMP* and is more complex in general.

Table 1 presents the solution results of CVRP by proposed modified ACO algorithm without the use of parallel computing and using the proposed scheme of parallelization. The experiments were conducted on the set of model tasks (benchmarks) from [4] with dimensions (number of clients) from 22 to 101. The table shows the solution time (in seconds) for some tasks using both methods. Improvements are presented both in absolute values (seconds) and relative values (percentage). The least relative improvement was obtained for the problem with 51 clients and amounted to 4.4%. The best result was obtained for the task with 101 clients, which gives relative improvement in 13.7%. Uneven improvements can be justified by the stochastic nature of the considered algorithm.

Table 1. Efficiency of parallel calculations in modified ACO algorithm.

Client amount	Execution time without parallel calculations, *seconds*	Execution time with parallel calculations, *seconds*	Absolute improvement, *seconds*	Relative improvement, %

22	5.4	5	0.4	7.4
30	20.4	19	1.4	6.9
51	118.6	113.4	5.2	4.4
76	337	317.8	19.2	5.7
101	770	664.6	105.4	13.7

5. Discussion and conclusions

First of all, note that the research results described in this paper are used for developing Transportation Management System for Russian Oil Company "Rosneft". After implementation of proposed algorithms the efficiency of transport logistics increased by 10% and decision making time for vehicle routing decreased by 20%.

When we begin to carry out computational experiments with parallel calculations of the algorithm we expected to get better time reducing results, but didn't reach the major improvement in execution time. Analyzing the reasons of such minor enhancement in time we can made the following conclusions.

The main shortcoming of an implementation of parallel calculations into ACO algorithm is the following. The biggest gain in time can be obtained by paralleling directly the process of the route building for each vehicle, i.e. while client choosing by determined and stochastic rules. But the arising problem is that the set of available clients to choose should be the same for all parallel processes. Otherwise there could be cases when the same client is chosen for several routes. Thus, if we follow this way of parallelization, it is necessary to block the set of available clients until the current process finishes client selection.

So the authors' future plans are to develop ACO method parallelization scheme to reduce execution time for big dimensions tasks.

References

[1] Hartmann A K and Rieger H 2002 *Optimization algorithms in physics* (Berlin: Wiley-VCH Verlag Berlin GmbH)
[2] Belov A and Slastnikov S 2014 A Metaheuristic Approach for the Problem of Motor Fuel Distribution *Proc. Comp. Sc.* **31** pp 143–150
[3] Bell J and McMullen P 2004 Ant Colony Optimization Techniques for the Vehicle Routing Problem *Adv. Eng. Inform.* **18** pp 41–48
[4] Augerat P *et al* 1995 Computational results with a branch and cut code for the capacitated vehicle routing problem *Technical report RR 949-M* (Grenoble: University Joseph Fourier)