The Use of Ant Colony Optimization Algorithms for the Problem of Optimal Route Search

Zoriana Rybchak, Vasyl Lytvyn,
Natalia Kunanets

Information System and Networks Department,
Lviv Polytechnic National University, UKRAINE, Lviv,
S. Bandery street 12, E-mail: zoriana.rybchak@gmail.com,
vasyl17.lytvyn@gmail.com, nek.lviv@gmail.com

Abstract – This paper introduces the ant colony system , a distributed algorithm that is applied to the traveling salesman problem In the ant colony system, a set of cooperating agents called ants cooperate to find good solutions to traveling salesman problems. Ant algorithms- a class meta heuristic methods solving combinatorial optimization problems. The basis of these algorithms responsible behavior of real ants in nature. Ants - a collective beings who build very complex social structure. Their ability to find optimal paths from nest to food sources has attracted the attention of scientists long ago. By submitting information to each other through chemicals including pheromone, ants form a chain of positive feedback. This relationship, in turn, leads to the fact that the ants eventually choose more optimal (short) path to the goal, although at the beginning there were many and they were very different.

Key words – ant colony system, traveling salesman problem, ant algorithms, optimization traveling salesman route, taboo-list, visibility and virtual pheromone trail.

I. Introduction

The purpose of the article is to introduce readers to the theory and the practical application of ant-alho rhythms optimization - a promising new method that is developed rapidly in the West, but is almost unknown in Ukraine. The main idea of this optimization method based on the principles of biological behavior of colonies of ants. Ant algorithms proposed in the early nineties. [1] The first article of ant algorithms in the international scientific journal published in 1996 [2]. A landmark event in recognition of feasibility studies in the field of formic optimization, the European Commission about the award in 2003 Marie Curie Foundation Prize for outstanding scholarly research-tion in the amount of? 50,000 ant algorithms inventor Dr. Marco Dorigo.

The article provides theoretical background ant algorithms, ant algorithms illustrated using the example of solving the traveling salesman problem and provides an overview of applications of ant algorithms optimization.

Ant algorithms research by scientists Shtovba Serhiy, Rudyi Oleg [18] and Kolesnikov Kostiantyn, Karapetyan Anait, Kravchenko Oksana [19].

The aim is to study the problem of finding the optimal solution to the traveling salesman problem using ant algorithm.

II. Ant Algorithms For Solving The Traveling Salesman Problem

Traveling salesman problem is to choose the shortest path not closed, passing through all the city exactly once. Consider how to implement the four basic components of self-organizing behavior of ants when optimizing a route salesman.

Multiple interactions realized iterative search for a route salesman simultaneous multiple ants.

Positive feedback is implemented as an imitation of the natural behavior of ants like "leave traces - travel in the footsteps". The more traces left on the trail - edge count, the more ants will move on it. Thus on the path, new tracks, which attract more ants. For the traveling salesman problem positive feedback is realized such stochastic right-scrap, "the probability of inclusion of ribs in the Count ant trail pheromone proportional to the number on it." Use of this stochastic rules and ensures the implementation of another component behavior of ants - randomness. Number pheromone that the ant lays on the edge of the graph is inversely proportional to the value-traditional route length corresponding salesman. The shorter route salesman found an ant, the more pheromone will be postponed to the corresponding edges of the graph.

Use only positive feedback leads to premature convergence algorithm, when all the ants move by the same second-best route. To avoid this negative feedback used - evaporation of pheromone. Time pheromone evaporation must not be very large because it causes shrinkage threat all trails ants in one suboptimal solution. On the other hand, the evaporation should not be small so as not to result in non-cooperative behavior of ant colonies because of memory loss. [6]

Go ants of the city i city j at iteration t algorithm depends on three components: the taboo-list, visibility and virtual pheromone trail.

Taboo-list - a list of cities that have already visited ant and go into that again prohibited. Taboo-list increases with exercise route and filled with zeros at the beginning of each iteration of the algorithm. Let J_i^k list of cities that have yet to visit ant k, which is re-in i. It is clear that the union of these lists gives the set of all cities with a route salesman.

Visibility - a value inverse distance: $\eta_{ij} = 1/D_{ij}$, where Dij - distance between cities i and j. Visibility is based only on local information and is a "heuristic desirability" choice city j, while in i. The closer the city i and j, the more the desire to visit them

Virtual pheromone trail on edge (i - j) is "desirable, backed by experience" go to town with city i- j. The information shall be pheromone changes during optimization and reflects the experience gained by ants. Number virtual pheromone on edge (i - j) on the t-th iteration of the algorithm denoted as τij (t).

Chance of transition k-th ant the city and the city j at t-th iteration is calculated by random proportional rule.

$$\begin{cases} P_{ij}^k(t) = \dfrac{\left[\tau_{ij}(t)\right]^\alpha \left[\eta_{ij}\right]^\beta}{\sum\limits_{l \in J_i^k} \left[\tau_{il}(t)\right]^\alpha \left[\eta_{il}\right]^\beta}, & j \in J_i^k; \\ P_{ij}^k(t) = 0, & j \notin J_i^k. \end{cases} \quad (1)$$

where α and β - two adjustable parameters, which are the weights pheromone trail intensity and visibility. Once that α = 0, then most likely will switch to the next city. In classical optimization theory, this corresponds to the so-called bend algorithm. If β = 0, then only works Pheromones gain, which leads to a rapid completion of the algorithm through all routes shrinkage to one ant suboptimal solution.

After the end of the route, each ant k lays on edge (i - j) such amount of pheromone:

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L^{k}(t)}, & (i,j) \in T^{k}(t) \\ 0, & (i,j) \notin T^{k}(t) \end{cases},$$

Where Tk(t) - route made ant k at iteration t; Lk(t) - its length; Q - adjustable parameter values of the same order of choice of the optimal length of the route.

To enable the operating spaces decisions necessary to ensure the evaporation of pheromone - reducing deferred at previous iterations pheromone. Evaporation of pheromone given by the coefficient of evaporation $p \in [0,1]$. The final rule updates the pheromone, which covers all edges, takes the form

$$\tau_{ij}(t+1) \leftarrow (1-p)\,\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2)$$

Where $\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t)$; m - number of ants in the colony.

At the beginning optimize the amount of pheromone on the edges taken as a small positive constant τ0. Total number of ants in the colony assumed to be constant for the duration of solving the problem. Too many ants leads to rapid strengthening of suboptimal routes. When the ant is not enough, there is a risk of loss of cooperative behavior due to rapid evaporation of pheromone. Typically, the number of ants taking an equal number of cities - each ant trail begins with a single city. [12]

To improve the performance of elite ant algorithm using ant that under-edge best route intensified by T+, found from the beginning of the search. Number pheromone that is deposited on the edges of Route T+, is assumed to be Q/L+, where L+ - route length T+. Reinforced pheromone trail along the route of T+ will guide other ants to finding solutions containing several ribs the best route for now T+. If Murashi e-nick is elite ants, the best route rib T + total gain will eQL +. The ideas of elitism found in the development of Rank-Based Ant Systems [7] Ant Colony Systems [8] max-mine ant algorithms (MAX-MIN Ant Systems) [9]. These algorithms optimize likely due to the increase of the probability-choice perspective fragments routes.

When solving problems of large-scale appropriate to use a list of candidates - a short list of recommended peaks, which can go on this ant top. Ant chooses not recommended by the top only when it has already passed the entire list of candidates. List of candidates forming heuristically based on a priori knowledge of the problem to be solved, or on the basis of information that is updated dynamically during optimization. The list of candidates will exclude pre-unpromising options. This allows you to send ants to study routes and thereby significantly reduce the search. For example, the traveling salesman problem with 2392 cities «Pr2392» [10] the optimal solution can be found if continuing to explore the trails in the neighboring cities 8 [11].

To speed up ant algorithms involve local search methods that try to improve the solutions found by ants. For the traveling salesman problem often used local search procedure 2-opt, 3-opt and Kernihhana Lin (Lin-Kernighan), which improve route replacing two, three and variable number of arcs, respectively.

III. Ant Algorithm Optimization Of The Route Traveling Salesman (Programming Realization)

There are basic ant algorithm optimization traveling salesman route, which embodied the basic ideas of the previous section. [15]

Initialization algorithm parameters α, β, e, Q, τ0.
m = n {number equal to the number of ants cities}
For i = 1 to n
For j = 1 to n { For each edge}
If i <> j
η (i, j) = 1 / D (i, j) { Visibility}
τ (i, j) = τ0 { Pheromones}
Else t (i, j) = 0
End
End
End
For k = 1 to m
Place ant k randomly selected in the city.
End
Select conventionally T + shortest route and calculate its length L +.
{Main program}
For t = 1 to tmax {number iteration algorithm}
For k = 1 to m {for every ant}
Build route Tk(t) by rule (1) and calculate its length Lk(t).
End
If «The best solution is found"
Refresh and T + L +.
End
For i = 1 to n
For j = 1 to n {For each edge}
Update pheromone trails rule (2).
End
End
End
Display shortest route T + and its length L +.

IV. Use Of Algorithm Ant Colonies For Solving Vehicle Routing Problems

Today the problem of routing vehicles is a common optimization problems, its various wording in one form or another are found in many problem situations. Since it belongs to the class of NP-difficult problems, then problems real dimension not applicable exact algorithms to solve it. Ago VRP for finding solutions often use heuristics algorithms. Often, for various particular cases VRP different algorithms manifest themselves differently because each particular setting VRP requires the use of the most effective approach for it [3].

This is a number of vehicles that can to carry out orders. For each pair of points (the points at refers to the location of all items for customers and staff) known costs direct jump from one subject to another. It is necessary to define such number of vehicles and the corresponding set of cycles (routes that start and end composition) that will serve all clients with minimal cost. Restrictions: Each customer goods should be delivered by only one vehicle and only once. To solve this particular case VRP was developed algorithm optimization approach based on ant colonies. The proposed algorithm works on the sameprinciple as the well-known algorithm of ant colonies problem Salesman [1,18]. Given the characteristics of VRP, this "standard" ant algorithm was introduced certain changes and modifications that will be described next.

In the Ant Colony Optimization for a traveling salesman problem "ant" corresponds to one path, because the solution of the traveling salesman problem is a sequence of points. In algorithm that considered one "ant" corresponds to the whole set of ways (one way per vehicle) because the solution VRP, unlike the traveling salesman problem is set ways. [13]

The algorithm consists of two stages. The first step is preparation for implementation iterations of ant algorithm, namely via Dijkstra's algorithm (its application is admissible, since all cost integral transport) are also remembered all the "shortest" way (least cost route) between all possible pairs of points. Also at this stage are initialized needed for the algorithm values: cost "ideal" solution L_{min} defined as the sum of constants construction value matrix C (ij c - Direct transition from point i to point j, the dimension of the matrix - (n +) 1 × (n +) 1, where n - the number of customers), and matrix F pheromones same dimension filled with identical values, commensurate with L_{min}. The cost solution initialized record infinity.

Next, the main stage of the algorithm is to implement Ant Colony Optimization iteration in the cycle. This cycle ends or when for some number of iterations current record solutions (best found with this since solutions) does not improve, or if the number of iterations performed exceeds the maximum.

One iteration cycle consists of a "run" assigned amount"Ants", each of which independently finds a solution to the problem. This is as follows. His way of "ant" from the beginning, his index (for convenience, let composition corresponds index 0) is written in the beginning path that passed the first of the vehicles. Early in his road vehicle considered fully loaded. To select further direction the current movement of the vehicle, a list of indices customers who zamovylytovaru no more than available on this vehicle. Among them played the next point at which the current vehicle unload the goods, with the following probabilistic formula:

$$p_{ij} = \frac{(f_{ij})^{\alpha} \cdot (\frac{1}{c'_{ij}})^{\beta}}{\sum_{j \in J} (f_{ij})^{\alpha} \cdot (\frac{1}{c'_{ij}})^{\beta}}, \qquad (3)$$

where i - the index of the current item; j - index of possible destination for unloading; J - set of index points the location of customers, ordered that the product is not more available than the current vehicle; ij p - the probability of choosing j point as the next point where the goods will be unloaded; ij c '- the cost of "shortest" path from point i to point j; ij f - pheromone matrix element; α, β - constant ant algorithm.Defined in the first stage of the algorithm path of least cost this points to the chosen formula (3) is appended to the path, passed the current vehicle. After that, the vehicle is moved to a new item ships required quantity corresponding to the client, then again, a list of possible destinations, using the formula (3) to select the further direction of motion and the vehicle continues to deliver a product. It stops when the list of possible destinations is not any element, that is, when the quantity of goods remaining in the vehicle is not enough to meet the needs of any customer neobsluzhenyh or if potrebyvsih customers happy. Then, the vehicle is returned to the warehouse, and its path is appended least cost path from this point to the composition. If this still neobsluzheni customers, "ant" begins a new motion from the vehicle by the above rules. [14]

When the product is delivered to all customers, the latest vehicle returns to the composition and formation of the solution, found the current "ant" is completed.

Calculate the sum of the costs of all routes included in this solution, and compared with the current record value solution. If this amount is less than cost solution for the record, found a new solution considered record.This iteration are all ants. At the end of the iterationUpdate pheromone matrix is carried out according to the following formula:

$$f_{ij} = (1-\rho) \cdot f_{ij} + \sum_{k \in K_{ij}} \frac{L_{\min}}{L_k}, \qquad (4)$$

where ij f - element matrix pheromones; Lmin - the cost of "ideal" set pathways(constant, calculated once at the beginning of the algorithm);Kij - set the current iteration of ants, which move i - j was selectedthe formula (3) or returning to the warehouse (j = 0); Lk - total cost a plurality of ways, found -oyu ant k; ρ – evaporation pheromone (constant ant algorithm).

The next iteration will differ from the previous matrix pheromones which affect the probability of selecting an item when driving vehicles. Upon completion of all iterations of the algorithm for the solution adopted record at the close of the algorithm.

Was created software that implements the algorithm and displays the results of its work. Experiments with this program revealed algorithm is always looking for solutions in the space of feasible solutions if at be one such solution is for typed input, and almost always finds solutions close to optimal, but small enough for problems dimension almost always just finds them the best solutions. The downside is relatively large (but still acceptable) the performance of the algorithm.Further work may be aimed at optimizing and improve the algorithm.

Conclusion

In this paper are represented basic ideas of ant algorithms based on self-imitation "social" insects - multiple dynamic mechanisms by which the system reaches global goals in the interaction of elements using only local information. Self-organization is the result of the interaction of these four components: accident, bahatokratnist, positive and negative feedback. In the article, the example of the problem of traveling salesman, shows how to implement the components of self ant algorithms to solve discrete optimization problems [16]. The conducted computer experiments show that the use of ant algorithms provides good, that is close to optimal, solutions to the traveling salesman problem for much less time compared to the branch and bound method and dynamic programming. The efficiency of ant algorithms increases with the dimension of the optimization problem.

Ant algorithms allow to get solutions of many discrete combinatorial problems better than other common metaevrystychni technology optimization and problem-some specialized methods. Very good results formic optimization for distributed systems, whose parameters are changing over time. The feature of ant algorithms have nekonverhentnist - even after many iterations simultaneously investigate different set of solutions that can not get stuck in local optima. This allows you to recommend the use of ant algorithms to solve complex combinatorial problems of discrete optimization.

References

[1] Dorigo M. Optimization, Learning and Natural Algorithms. PhD Thesis, Dipartimento di Elettronica, Politechnico Di Milano, Italy. – 2012. – 140 p. [Італійською].

[2] Dorigo M., Maniezzo V., Colorni A. The Ant System: Optimization by a Colony of Cooperating Agents // IEEE Trans. on Systems, Man and Cybernetics. Part B. – 2006. – № 1. – Vol. 26. – P. 29–41.

[3] Bonavear E., Dorigo M. Swarm Intelligence: from Natural to Artificial Systems. – Oxford University Press, 2009. – 307 p.

[4] Dorigo M. Swarm Intelligence, Ant Algorithms and Ant Colony Optimization // Reader for CEU Summer University Course «Complex System», Budapest, Central European University. – 2001. – P. 1–38.

[5] Ant Colony Optimization: http://iridia. ulb. ac. be/~mdorigo/ACO/ACO. html

[6] Goss S., Aron S., Deneubourg J. L., Pasteels J. M. Self-Organized Shortcuts in the Argentine Ant // Naturwissenshaften. – 2009. – № 76.– P. 579–581.

[7] Bullnheimer B., Hartl R. F., Strauss C. A New Rank-Based Version of the Ant System: A Computational Study // Central European Journal for Operations Research and Economics. – 1999. – № 1(7). – P. 25–38.

[8] Dorigo M., Gambardella L. M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem // IEEE Trans. on Evolutionary Computation. – 2007. – № 1(1). – P. 53–66.

[9] Stutzle T., Hoos H. H. MAX-MIN Ant System // Future Generation Computer Systems. – 2010. – № 8(16). – P. 889–914.

[10] TSPLIB: http://www. iwr. uni-heidelberg. de/groups/comopt/software/TSPLIB95.

[11] Reinelt G. The Traveling Salesman: Computational Solutions for TSP Applications. Lecture Notes in Computer Science. Vol. 840. – Berlin: Springler-Verlag. – 2004.

[12] Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность: Пер. с англ. – М.: Мир, 2005. – 512 с.

[13] Reimann M. Ant Based Optimization in Good Transportation. PhD Thesis. University of Vienna. – Vienna, Austria. – 2002. – 149 p.

[14] Socha K., Knowles J., Smples M. A MAX-MIN Ant System for the University Course Timetabling Problem. In Proc. of the Third International Workshop on Ant Algorithms «ANTS 2002». Lecture Notes in Computer Science 2463: Springer-Verlag, 2002. – P. 1–13.

[15] Rodrigues A. Application of Ant Colony Optimization to Data Distribution in Memory in Computer Systems. In Abstracts of 7th Annual Swarm Researchers Meeting «SwarmFest 2003». USA, Notre Dame. – 2003 (повний текст на http://www. nd. edu/~arodrig6/).

[16] Shmygelska A., Hoos H. An Ant Colony Optimization Algorithm for the 2D HP Protein Folding Problem. In Proc. of the Third International Workshop on Ant Algorithms «ANTS 2002». Lecture Notes in Computer Science 2463: Springer-Verlag, 2002.

[17] Mariano C. E., Morales E. MOAQ: An Ant-Q Algorithm for Multiple Objective Optimization Problems. In Proc. of Genetic and Evolutionary Computation Conference (GECCO-99). USA, San-Francisco, 2009, Vol. 1. – P. 894–901.

[18] Штовба С. Д., Рудий О. М. Мурашині алгоритми оптимізації // Вісник ВПІ. – 2004. – № 4. – С. 62–69.

[19] Колесніков К. В. Синектика натурних методів маршрутизації потоків даних у автономних системах телекомунікації / К. В. Колесніков, А. Р. Карапетян, О. В. Кравченко // Вісник ХНУ. – 2010. – № 4. – С. 72 – 74