ANT COLONY OPTIMIZATION FOR TIME DEPENDENT SHORTEST PATH PROBLEM IN DIRECTED MULTIGRAPH

Leonid Hulianytsky, Anna Pavlenko

Abstract: *The paper concerns the approach for searching shortest path between specified nodes in a given graph that represents scheme of possible flights and takes into account time-dependent price. The path may be constructed according to request constraints: time limits, cost, mandatory transit or prohibited items. To find the lowest path cost we developed the ant colony optimization (ACO) based algorithm. Natural parallelism and iterativity of original ACO processing scheme gives the possibility to get and update the best current solution at any moment taking into account flight data changes. The approach of single-generation ACO, that allows optimizing the use of resources and reducing the processing time is suggested and investigated. The paper presents a formal model of the problem, and describes the basic ACO scheme and properties of suggested approach. For assessment practical effectiveness of single-generation algorithm, the experiments are made. The comparison between offered and classical ACO schemes in time and accuracy is given.*

Keywords: *ant colony optimization, time dependent shortest path problem*

ACM Classification Keywords: *I.2.8 Problem Solving, Control Methods, and Search*

Introduction

Many of real life situations of communication or transportation networks can be well modeled into multigraphs (graphs in which multiple edges between nodes might exist) because of their ability operating multiple edges connecting a pair of nodes. Due to the increasing interest in the dynamic management of transportation systems, there are needs to find shortest paths over a large graph (e.g., a road network), where the weights associated with edges dynamically change over time [Ding, 2008].

Finding shortest path in graphs has been playing an important role in various fields of human activity for over 40 years. Typically, results must be found within a very short time period. In real-time searching systems new routes must be identified within a reasonable time after a customer requests [Fu et al, 2006]. General time-dependent shortest path problem is not new. Some of the first studies were published in 1958 in which Cook and Halsey [Cook & Halsey, 1969] proposed algorithm based on dynamic programming with discretizing time. Alternative ways to solve the problem for different problem variations where investigated by Dreyfus [Dreyfus, 1969], Dijkstra, Halpern, Orden and Rom and others.

The complexity of the problem and its wide application in many fields of human activity stimulates researching different approaches and methods. Much attention is paid to approximate bio-inspired search techniques [Pintea, 2014]. These include ant colony optimization proposed by Dorigo [Dorigo & Stützle, 2004; Dorigo & Stützle, 2010], which is successfully applied to combinatorial optimization problems.

Traditional optimal shortest path techniques often cannot be applied because they are too computationally intensive to be feasible for real-time operations. Numerous heuristic search strategies have been designed for enhancing computational efficiency of shortest path search. Algorithm ant colony optimization (ACO) has been successfully used to solve combinatorial optimization problems, including the traveling salesman problem, routing, sequential ordering, assignment problem, classification, etc. (particularly on dynamic graphs).

The following problem is described below: given an airlines flights scheme between specified set of cities (airports) with appropriate conditions and restrictions. The research concentrates on solving problem of finding the cheapest path for travelling via planes from source city to target through specified points. Worth to mention that flight's price varies over time that makes the problem time-dependent. For simplicity in our approach time space is discretized in a suitable way. Since there may be several flights between the same airports, network is represented via multigraph.

Problem Formulation

Given a directed multigraph $G=(V,A)$ where multiple edges or arcs might exist between pair of vertices and represent flight connections between airports offered by airlines, where $V=\{v_1,...,v_n\}$ is a set of n vertices; A is a set of arcs. Let (v_i,v_j) be a set of arcs from node v_i to v_j, $v_i,v_j \in A$, $N_{ij}=\|(v_i,v_j)\|$ is a number of such arcs, a_{ij}^k is a specific arc $a_{ij}^k \in (v_i,v_j), k \in \{1,...,N_{ij}\}$. It is possible if $(v_i,v_j)=\varnothing$ for some nodes and destinations.

According to the problem, it is required to find *optimal path* from source (starting) node $s \in V$ to destination node $d \in V$ when starting time t_0 (departure time from the source) can be selected in a user given starting-time interval $T=[t_{0_{min}},t_{0_{max}}] \subseteq T$ (it is supposed that at least one such path exists).

Consider a path $x(s,d,t_0)$ from point s to point d starting in time t_0 to be an arcs sequence $(a_{i_1 i_2},a_{i_2 i_3},a_{i_3 i_4},\dots,a_{i_{w-1} i_w})$ if

1. $i_1=s$, $i_w=d$;
2. $a_{ij} \in (v_i,v_j), v_i,v_j \in V, i,j \in \{i_1,i_2,\dots,i_w\}$.

If transit across $a_{kl} \in A$, that belongs to path and corresponds to flight from point k to point l, is starts from k in time t_{k-1}, than arrival time to point l is $t_k = t_{k-1} + \lambda(a_{kl})$, where $\lambda(a_{kl})$ is flight's duration. Travel time is the difference between arrival time and starting time (1). Flight's durations is considered to be fixed.

$$t(x) = \sum_{k=1}^{w-1} \left[t(a_{i_k, i_{k+1}}) + g(a_{i_k}) \right] - t_0 \qquad (1)$$

where $t(x)$ – full path duration;

$t(a_{i_k, i_{k+1}})$ – transition time across $a_{i_k, i_{k+1}}$ arc;

$g(a_{i_k})$ – time of waiting in starting node v_{i_k} of arc a_{i_k}, $g(a_{i_w}) = 0$;

$c(a_{i_k, i_{k+1}}, t)$ is nonnegative transit-time function which represents generalized cost of travelling across arc $a_{i_k, i_{k+1}}$.

The cost of path x is defined as $c(x,t) = \sum_{k=1}^{w-1} c(a_{i_k, i_{k+1}}, t)$, where w – number of arcs in route x.

The goal is to find optimal path $x^*(s,d,t_0)$ in terms of price (or a set of allowed routes with account to additional constraints).

There is also a set of additional constraints (2)-(6): given a set of mandatory vertices $V_{mandatory}$ (2) included in optimal path $x^*(s,d,t_0)$; a set of prohibited vertices $V_{prohibited}$ excluded from $x^*(s,d,t_0)$ (3); c_{max} – maximum allowed cost of x from s to d (4); maximum number of transition nodes n_{max} (5); satisfy the constraint of route time length (6). Worth to mention that it is not obligatory to change nodes at each iteration (every day).

$$V_{mandatory} \subseteq x^*, \qquad (2)$$

$$\forall v \in V_{prohibited} : v \notin x^*, \qquad (3)$$

$$c(x^*, t) \le c_{max}, \qquad (4)$$

$$\left| x^* \right| \le n_{max}, \qquad (5)$$

$$t_{min} \le t(x^*) \le t_{max}, \qquad (6)$$

where t_{min}, t_{max} – minimum and maximum time period;

Time-Dependent Shortest-Path (TDSP) problem is to minimize travel cost (7) among allowed paths:

$$c(x^*, t) = \min\{c(x, t)\}, \forall x(s, d, t_0). \tag{7}$$

ACO Approach Description

Ant Colony Optimization [Dorigo & Stützle, 2004] is inspired by the idea of solving optimization problems using low-level communication behavior of cooperative ants that seek a path between their colony and a source of food.

Same as in real life, ants start randomly wander upon finding food and then return to their anthill. During the walk they leave pheromone trails which make their path more attractive for other ants since their chance to succeed in finding food increases. The following ants will choose trails taking in account the amount of pheromone deposited on the ground and visible path length (distances to neighboring nodes). However pheromone trails are evaporating all the time thereby reducing its attractiveness strength. Obviously short and popular paths have higher density than longer ones. Modelling evaporation process in ACO helps to avoid convergence to local optimal solution. Figure 1 represents general scheme of ACO.

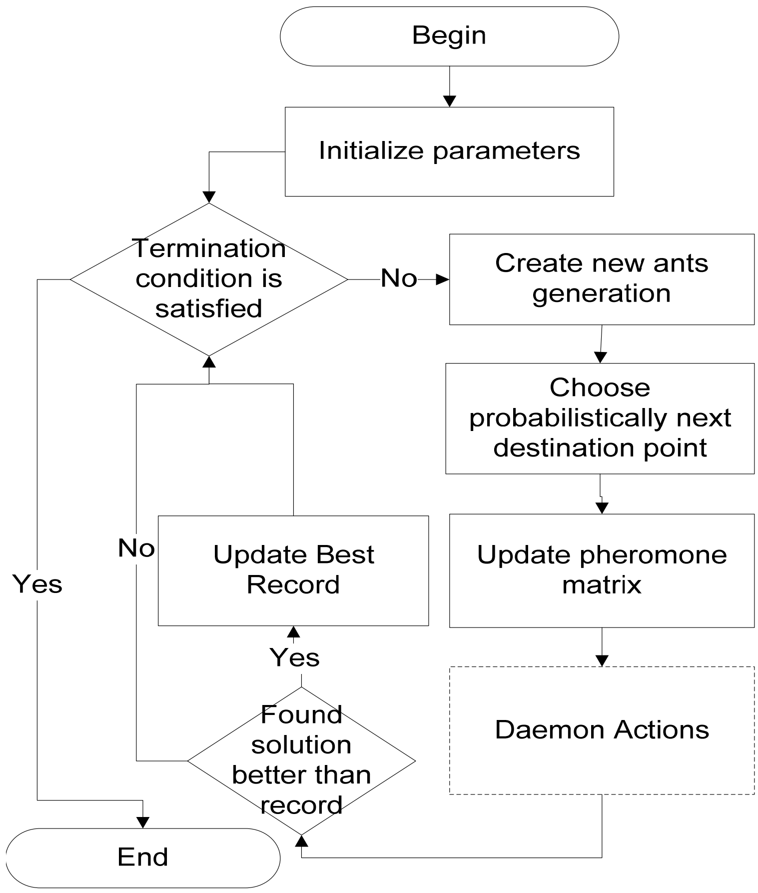

Figure 1. General Ant Colony Optimization Scheme

Artificial ants in ACO represent stochastic procedures that move on the graph and constructs solutions.

The algorithm iteratively generates ants, which stochastically choose transit points and build route. Starting point for each ant is determined via problem constraints. On the first step algorithm creates and initializes matrices of distances between airports and pheromones, initial "best" route cost and other algorithm's variables. The next step creates a generation of ants that simultaneously go from the starting point. Each ant with a probability (8) defines its next point:

$$p^u_{ij} = \frac{\tau^{\alpha}_{ij}\eta^{\beta}_{ij}}{\sum_{allowed\ j}\tau^{\alpha}_{ij}\eta^{\beta}_{ij}} \tag{8}$$

where τ^{α}_{ij} – the amount of pheromone, deposited on the way from i to j, $0 \le \alpha$ – a parameter that controls impact of the deposited pheromone; η^{β}_{ij} – attractiveness of transition ij, based on a priori knowledge of the distance, $\eta_{ij} = 1/d_{ij}$, β – parameter that controls impact of η_{ij}. After all ants of current generation complete building routes, pheromone update occurs via formula (9):

$$\tau_{ij} := (1-\rho)\tau_{ij} + \sum_{u}\Delta\tau^{u}_{ij} \tag{9}$$

where ρ – pheromone evaporation coefficient, $\Delta\tau^{u}_{ij}$ – the amount of deposited pheromone by ant u, which is calculated via (10):

$$\Delta\tau^{u}_{ij} = \begin{cases} c_{predefined}/c(x_u), & if\ (ij) \in x_u\ , \\ 0, & otherwise \end{cases} \tag{10}$$

where $c_{predefined}$ – coefficient, which usually corresponds to the order of optimal route; $c(x_u)$ – ant's found route cost. If ants' generation produces better solution, than current record, it has to be updated. Different types of problems might have own heuristics – a priori knowledge that should be used to improve constructing solutions. Daemon actions are a kind of custom optional procedures that can be applied as a final step of the iteration according to specificity of the problem. After constructing full paths ants release all allocated resources and disappear.

If the termination condition is not satisfied, it creates a new generation of ants and new iteration of the algorithm starts. Termination conditions might be selected according to algorithm processing time, iterations number, number of iterations without updating best records etc.

ACO for Solving Time-Dependent Shortest Path in Multigraphs with Additional Constraints

Current research describes developed ACO based algorithm for finding optimal path in dynamic multigraphs with additional constraints. An important criterion of solving described problem is algorithm's processing time. In the research it is suggested to operate a given number of ants, not generations. Classical scheme uses the same total number of ants through all generations like single-generation approach.

Figure 2 represents general scheme of suggested single-generation approach. During initialization $c_{predefined}$ value is defined by additional run of simple ACO without additional constraints (3) - (7). If $c_{predefined}$ could not be found for specified number of iterations (no acceptable routes where found), the algorithm stops, informs about inability to calculate path and moves to the next request.

```
//Initialization
Initialize flights data matrices;
Run simplified ACO to detect approximate optimal path cost;
Initialize pheromone matrix;
while (termination condition is not met):
Begin
   // Construct solution
   For each ant do:
      Repeat:
         Choose probabilistically arc;
         If (ant's partial solution is not allowed)
         Begin
            Release resources;
            Loose ant;
            Continue;
         End
      Until (ant completes a solution);
      // Update pheromones trails
      Update attractiveness τ for each traversed edge;
      // Update best solution
      If (local best solution better than global solution)
         Save local best solution as global solution;
      End;
      // perform optional daemon actions;
      Perform pheromone evaporation;
      Release resources;
      Loose ant;
   End;
End;
```

Figure 2. Single Generation Ant Colony Optimization Scheme

At the first step all arcs that contain $V_{mandatory}$ (or target node) are predefined with value τ_{max}, others – default τ_0. While choosing transit points ants operate by arc, not nodes as in classical algorithm. It effectively deals with the problem of multiple arcs between some nodes. Once an ant completes building route or its partial route does not satisfy the conditions (2) - (6), the ant disappears and releases resources. Therefore, the algorithm does not hold any unused resources like classical ACO scheme does while synchronizing generation of ants. Worth to mention that ants collaborate with each other through pheromones trails all the time, whereas traditionally ants get information only from preceding ants' generations. Number of ants belongs to algorithm's parameters. Performance comparisons of classical and suggested approaches are presented below.

Pheromone update is performed after each successful ant's completed walk by formula (11) - (13) for arcs that belong to constructed route:

$$\tau_{ij} := \tau_{ij} + \Delta\tau_{ij}^{u}, \tag{11}$$

$$\tau_{min} \le \tau_{ij} \le \tau_{max}, \tag{12}$$

$$(ij) \in x_u, \tag{13}$$

where τ_{min}, τ_{max} – parameters of the algorithm. Pheromone evaporation is performed as daemon actions after each b-th created ant for the whole pheromone matrix (14). This is done to reduce update operations under database.

$$\tau_{ij} := (1-\rho)\tau_{ij}, \ \forall (ij) \in A \tag{14}$$

Computational Results

A number of tests have been conducted under realistic conditions, using the data collected by developed parsing client and APIs from global travel search site SkyScanner and Google (QPX Express). Obtained data includes 15497 flights for one week. According to the experiment's conditions, it is necessary to find the optimal path between Boryspil airport and 113 airports in Europe. For descriptive reasons the following charts contain info about 50 random target cities only. Figure 3 shows comparison of single-generation approach and classical ACO in time (in seconds). Classical scheme is on average 31% slower than suggested scheme. Offered algorithm was unable to find acceptable solutions for 25 target cities from 113, classical scheme – for 16 cities.

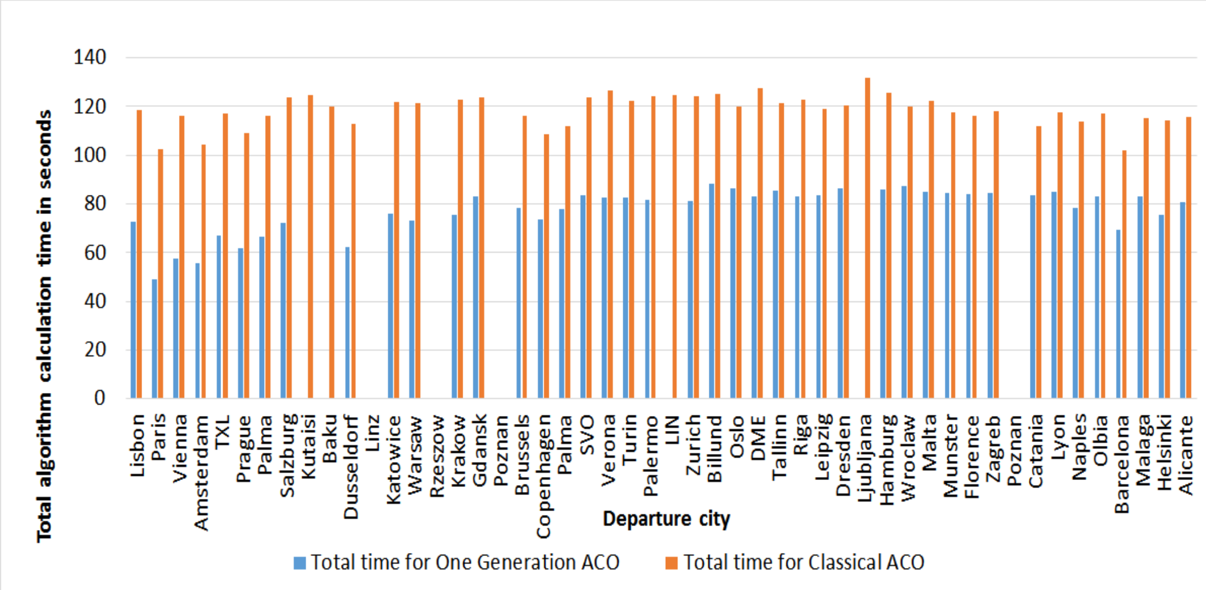

Figure 3. Comparison of Single-Generation Approach and Classical ACO by Algorithm's Processing Time

Figure 4 demonstrates experiment's results for random 50 target cities and illustrates comparison of single-generation approach and classical ACO by best solution in terms of route price (in EUR). Suggested scheme provides 16% better results than classical.

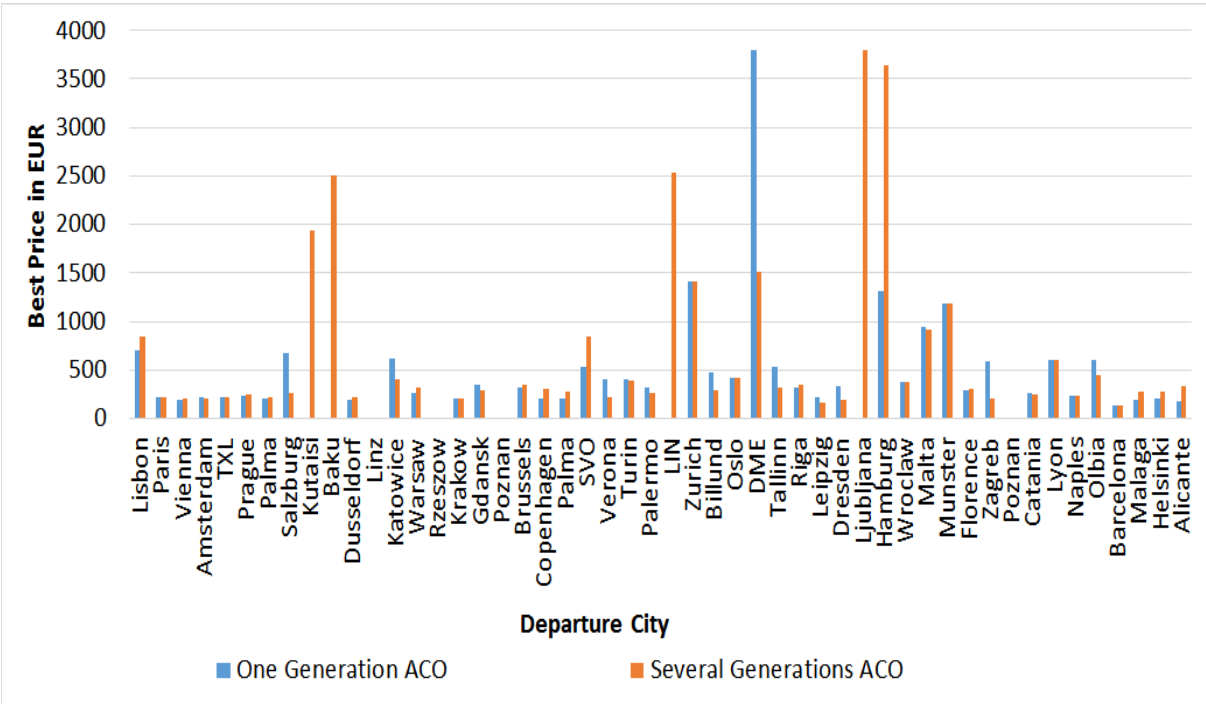

Figure 4. Comparison of Single-Generation Approach and Classical ACO by Best Solution in EUR

Table1 contains algorithm parameters, which were used in the experiments.

Table 1. Algorithm parameters values

Name	Value
Number of runs for each suggested approach	100
Total number of ants produced by algorithm	500
Impact rate of the deposited pheromone α	0.5
Impact rate of $\eta_{ij} - \beta$	0.5
Maximum number of transition nodes n_{max}	10
Generation size for classical ACO	20
Ant number after which starts evaporation in single-generation approach b	20
Default initial pheromone amount τ_0	0.1
Minimum pheromone amount τ_{min} per arc	0.1
Maximum pheromone amount τ_{max} per arc	0.7
Pheromone evaporation coefficient ρ	0.1
Maximum number of iterations to detect $c_{predefined}$	3000
Source airport	KBP

Conclusion

We propose the ACO based algorithm for solving the time dependent shortest path problem for large multigraph.

The description and formalization of time-dependent shortest problem with additional constraints are presented in the paper. For the need to minimize algorithm's processing time, the ACO classical scheme has been modified: the algorithm works with single generation and choosing next point procedure that operates with arches. The approach solves the problem with additional constraints: maximum number of transit cities, maximum price value, prohibited and mandatory cities. The algorithm allows to expand flexibly model by means of additional conditions and constantly to control the feasibility of solutions. In addition, it is proposed to carry out additional updates pheromone matrix along the arcs that contain mandatory and final city.

The paper represents experiments, that using the real data sets, and compares the suggested approach and the classical ant colony optimization algorithm. Our preliminary results show that the proposed algorithm provides good quality results and significantly less processing time algorithm. However, the algorithm often does not find optimal (feasible) solutions than the classical scheme.

Additional research needs optimization algorithm parameters, and the comparison with other search algorithms. Updating pheromone scheme could be improved for work with several user requests reusing existing information.

Bibliography

[Cook & Halsey, 1969] K. L. Cook, E. Halsey, "The shortest route through a network with time-dependent intermodal transit", In J. Math. Anal. Appl., 1966, pp. 493-498.

[Ding, 2008] Ding. "Finding Time-Dependent Shortest Paths over Large", In: Proceedings of the 11th international conference on Extending database technology: Advances in database technology. Nantes, France, 2008, pp. 205-216.

[Dorigo & Stützle, 2004] M. Dorigo M., T. Stützle, "Ant Colony Optimization, A Bradford Book", MIT Press.Cambridge, Massachusetts, London, 2004, 305 p.

[Dorigo & Stützle, 2010] M. Dorigo, T. Stützle, "Ant colony optimization: overview and recent advances", In: Handbook of metaheuristics. Springer US, 2010, pp. 227-263.

[Dreyfus, 1969] S.E. Dreyfus, "An appraisal of some shortest-path algorithms", In Operations Research, 1969, pp. 395-412.

[Fu et al, 2006] L. Fu, D. Sun, L.R. Rilett, "Heuristic shortest path algorithms for transportation applications: State of the art", Computers & Operations Research, Vol. 33, No. 11, 2006, pp. 3324-3343.

[Pintea, 2014] Pintea C.M., "Advances in Bio-inspired Computing for Combinatorial Optimization Problems", Berlin, Heidelberg: Springer-Verlag, 2014, 188 p.

Authors' Information

Hulianytsky Leonid *– Dr.Sc. (Technology), Head of department of Glushkov Institute of Cybernetics of NAS of Ukraine, Professor of NTUU "KPI" (Kyiv); e-mail: leonhul.icyb@gmail.com*

Major fields of scientific research: combinatorial optimization; decision making; mathematical modeling and applications; forecasting.

Pavlenko Anna *– PhD student at Glushkov Institute of Cybernetics of NAS of Ukraine (Kyiv) Glushkov Ave., 40, Kyiv, 03680, Ukraine; e-mail: dmitrieva.anya@gmail.com*

Major fields of scientific research: artificial intelligence; combinatorial optimization; mathematical economics, forecasting