An Ant Colony Optimization Approach for the Two Disjoint Paths Problem with Dual Link Cost Structure

정지복* · 서용원**
*대전대학교 경영학과, jbchung@dju.ac.kr
**단국대학교 경영학과, seoyw@dankook.ac.kr

Abstract

The ant colony optimization (ACO) is a meta-heuristic inspired by the behavior of real ants. Recently, ACO has been widely used to solve the difficult combinatorial optimization problems. In this paper, we propose an ACO algorithm to solve the two disjoint paths problem with dual link cost structure (TDPDCP). We propose a dual pheromone structure and a procedure for solution construction which is appropriate for the TDPDCP. Computational comparisons with the state-of-the-arts algorithms are also provided.

Keywords: Ant Colony Algorithm, Disjoint Paths Problem, Meta-heuristic

1. Introduction

The problem of finding disjoint paths in a network has been given much attention due to its practical significance in many applications such as survival design of communication networks and layout design of integrated circuits. In a survivable communication networks, traffic is carried on the working path (WP) unless it is affected by link (node) failure, upon which, the traffic is re-routed along the backup path (BP). It is useful to find a link (node) disjoint path pair to protect traffic against any link (node) failure.

There are some various problems related with the disjoint paths problem (DPP) depending on the objectiveness of the problem [10]. Under the assumption that there are two different costs for each link, MinSum problem is to minimize the sum of the costs of two disjoint paths while MinMax (MinMin) problem is to minimize the larger (smaller) one of two disjoint paths respectively. In this paper, we consider a MinSum problem in a network with dual link cost structure (TDPDCP).

ACO is a meta-heuristic which was formalized by Dorigo and co-workers [2]. ACO was first proposed to tackle the famous traveling salesman problem (TSP) and has been successfully applied to many other combinatorial optimization problems (COPs) such as Graph Coloring Problem [8], Steiner Tree Problem [9], Maximum Independent Set Problem [1], Multiple Knapsack Problem [3][5] and Bin Packing Problem [6].

This paper is organized as follows. In section 2, we describe a mathematical formulation of the TDPDCP. In section 3, general overview of the ACO is provided. Section 4 is devoted to explain how the proposed ACO can be used to solve the TDPDCP. In section 5, computational experiments are performed to show the performance of the proposed ACO.

2. Problem Description

In this paper, we consider a graph G = (N,E,A,B) with node set N and link set E. There exist two different costs on every link, where A=(a_{ij}) is the set of cost for working link and B=(b_{ij}) is the set cost fro backup link. We consider the problem of finding two link-disjoint paths from node s to node t such that the total cost of two paths is minimized, where the cost of working (backup) path is associated with the working (backup) link. Generally a link disjoint paths algorithm can be extended to a node disjoint algorithm with the concept of node splitting, i.e. replacing one node with two nodes that are linked together by a link with zero weights.

Li et al.[7] showed that the computational complexity of TDPCDP belongs to NP-Hard.

Let $x_{ij} = 1$ if link (i,j) is used for working path and zero otherwise. Let $y_{ij} = 1$ if link (i,j) is used for backup path and zero otherwise. Then integer linear programming (ILP) formulation for the TDPDLC can be formulated as follows:

$$\text{Minimize } \sum_{(i,j)\in E} a_{ij} x_{ij} + \sum_{(i,j)\in E} c_{ij} y_{ij} \quad (1)$$

$$\text{s.t } \sum_j x_{ij} - \sum_j x_{ji} = 0, \forall i \neq s, d \quad (2)$$

$$\sum_j x_{sj} - \sum_j x_{js} = 1 \quad (3)$$

$$\sum_j x_{dj} - \sum_j x_{jd} = -1 \quad (4)$$

$$\sum_j y_{ij} - \sum_j y_{ji} = 0, \forall i \neq s, d \quad (5)$$

$$\sum_j y_{sj} - \sum_j y_{js} = 1 \quad (6)$$

$$\sum_j y_{dj} - \sum_j y_{jd} = -1 \quad (7)$$

$$x_{ij} + y_{ij} \leq 1, \forall (i,j) \in E \quad (8)$$

$$x_{ij}, y_{ij} \in \{0,1\}, \forall (i,j) \in E$$

The objective function (1) implies to minimize the sum of path costs. The constraints (2),(3),(4) imply the flow conservation for working path and the constraints (5),(6),(7) imply the flow conservation for backup path. The constraints (8) imply the link disjoint between the working path

and backup path.
Although heuristic algorithm can not guarantee the optimality, usually it can find the feasible solution more quickly than the exact algorithm.

3. Overview of Ant Colony Optimization
ACO was firstly proposed for the well-known Traveling Salesman Problem (TSP) by Dorigo at 1997 [2]. But today there exists a considerable amount of references which have showed quite good performance in many applications [1][5][6].
ACO can be thought as a randomized construction heuristic that probabilistically makes a decision as a function of artificial pheromone traits and available heuristic information based on the problem instances being solved.
A colony of ants moves through adjacent states of the problem concurrently and asynchronously. They move by applying probabilistic decision policy that makes use of the pheromone trails and heuristic information.
ACO algorithm makes use of simple agent called (artificial) ants which iteratively construct candidate solutions to a COP. The ant's solution is guided by pheromone trail and heuristic information of a given problem. ACO can be applied to any COP by defining solution components which the ants use to construct candidate solution and on which they may deposit a pheromone.

At each step, each ant k computes a set of feasible expansion to its current states and moves to one of these according to a probability distribution.

For ant k, the transition probability of $p_{ij}^k(t)$ moving from a state i to state j depends on the combination of two values, pheromone level and heuristics information. τ_{ij} is a pheromone level deposited between state i and j. η_{ij} is a heuristic information which means a attractiveness of moving from a state i to state j.
Thus the higher the two values, the more desirable it is to include state j in the partial solution.

4. Proposed ACO for TDPDCP
ACO can be applied to any COP for which some solution construction mechanism can be conceived. Given a COP, the first step is to define an adequate model to represent a COP. After defining generic problem representation, then construct a solution by defining the ant's behavior probabilistically. Finally update the solution component. This step is called pheromone evaporation and is required to avoid a too rapid convergence of the algorithm [2].

4.1 Construct Ant Solution
In our problem, we introduce two kinds of ants and pheromones since there are two kinds of cost for each link. For simplicity, we call them as blue or red respectively.

Let τ_{ij}^b (τ_{ij}^r) be the amount of blue (red) pheromone on the link ij and η_{ij}^b (η_{ij}^r) be a local information for the blue (red) link ij. Let Pb (Pr) be the set of links which is included in the working (backup) path from node s to node t.
d_j^* (f_j^*) is the shortest path length of the working (backup) path from node j to destination node t.
Furthermore, α and β are positive parameter, whose value determines the relative importance of pheromone versus local information and ρ be a evaporation ratio.

Using the above notation, ACO for TDPCDP can be proposed as follows:

Let P_{ij}^b (P_{ij}^r) be the probability that blue (red) ant positioned at node i may select node j as a next movement.

$$P_{ij}^b = \frac{(\tau_{ij}^b)^\alpha * (\eta_{ij}^b)^\beta}{\sum_{ij} (\tau_{ij}^b)^\alpha * (\eta_{ij}^b)^\beta}, \quad P_{ij}^r = \frac{(\tau_{ij}^r)^\alpha * (\eta_{ij}^r)^\beta}{\sum_{ij} (\tau_{ij}^r)^\alpha * (\eta_{ij}^r)^\beta} \quad (9)$$

For our problem, we define η_{ij}^b (η_{ij}^r) as follows.
η_{ij}^b (η_{ij}^r) = $1/(a_{ij} + d_j^*)$ ($1/(c_{ij} + f_j^*)$).

4.2 Update Pheromones
The aim of pheromone update is to increase the pheromone values associated with good or promising solutions. This process is called pheromone evaporation where pheromone level is changed as follows:

$$\tau_{ij}^b(t+1) = (1-\rho)\tau_{ij}^b(t) + T^b,$$

$$\tau_{ij}^r(t+1) = (1-\rho)\tau_{ij}^r(t) + T^r \quad (10)$$

where $T^b = Q / S^b$ if link ij is included in S^b and zero otherwise and $T_m^r = Q / S_m^r$ if link ij is included in S^r and zero otherwise. And S^b is the path cost of blue ant and S^r is the path cost of red ant. In our problem, we set Q as the lower bound of the problem.

The overall procedure of algorithm is displayed at figure 1 below.

Figure 1. Overall procedure of algorithm

```
Step 0: Initialization.
    Place two ants (blue and red) at source node
    Compute d_j^* (f_j^*) by running the dijkstra algorithm
    from all nodes to node t
Step 1: Construct Solution.
    Select blue or red ant randomly
    For the selected ant
        Select an adjacent node with probability given by (9) and merge
        it with Pb or Pr depending on the color of ant until node t is
        included.
    Delete the path Pb or Pr from G temporarily and apply above step
    for the remaining ant
Step 2: Update.
    Update the pheromones on each link given by (10)
```

5. Experimental results

To know the performance of our approach, we do the experiment on randomly generated network G(15, 94). And the link costs are randomly generated between 1 and 10.
Although there are many implementation options for ACO, firstly we select two options and tested on each case. The first option is whether to use a heuristic information (η) or not and the second option is related with the pheromone level update.

In table 1, options 1 means not to use heuristic information and all ants update the pheromone level. Options 4 means to use heuristic information and only the best ant update the pheromone level.

Table 1. Experimental Environment

	Heuristic Info.	Best Ant Update
option 1	X	X
option 2	O	X
option 3	X	O
option 4	O	O

The results are displayed at figure 2-5. The computation time is about 110~128 (msec) at all cases. The “obj” means the objective value obtained at each iteration and “minObj” means the best objective value obtained so far.
As you can see in figure 2-5, “heuristic information” and “best ant update” options contribute to greatly improve the performance of algorithm.

To test the robustness of algorithm, more experiments in various environments are required.

Reference

[1] Choi,H., N.Ahn and S.Park, An Ant Colony Optimization Approach for the Maximum Independent Set Problem, Journal of the Korean Institute of Industrial Engineering, v33 n4, pp447-456, 2007
[2] Dorigo, M and K.Socha, An Introduction to Ant Colony Optimization, IRIDIA/2006-10, Technical Report, 2007
[3] Fidanova,S., Ant Colony Optimization for multiple knapsack problem and heuristic model, heuristic model, 2004
[4] Garey, M.R. and D.S.Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, 1979
[5] Kong, M., P.Tian, Y.Kao, A new ant colony optimization algorithm for the multidimensional Knapsack problem, Computers and Operations Research, 35, pp2672-2683., 2008
[6] Levine, J. and Frederick Ducatelle, Ant Colony Optimisation and Local Search for Bin Packing and Cutting Stock Problem, JORS, 2003
[7] Li, C., S.T.McCormick and D.Simchi-Levi, Finding disjoint paths with different path costs: Complexity and algorithms, Networks, v22, pp653-667, 1992
[8] Salari, E. and K.Eshghi, An ACO Algorithm for the Graph Coloring Problem, Int.J.Contemp. Math.Sciences, v3 n6, pp293-304, 2008
[9] Seo, M. and D. Kim, An Ant Colony Optimization Algorithm to Solve Steiner Tree Problem, Journal of Korean Operations Research and Management Sceince Society, v33 n3, pp17-28, 2008
[10] Xu D., Chen Y, Xiong Y, Qiao, C, He X. On Finding Disjoint Paths in Single and Dual Link Cost Networks, Proc. of IEEE Infocom 2004

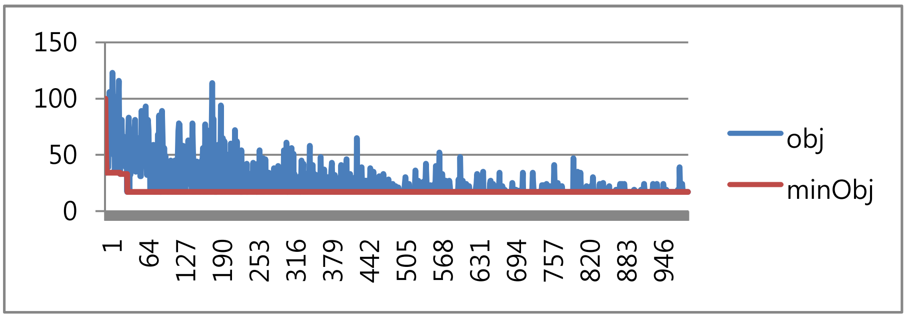

Figure 2. Experimental results of option 1

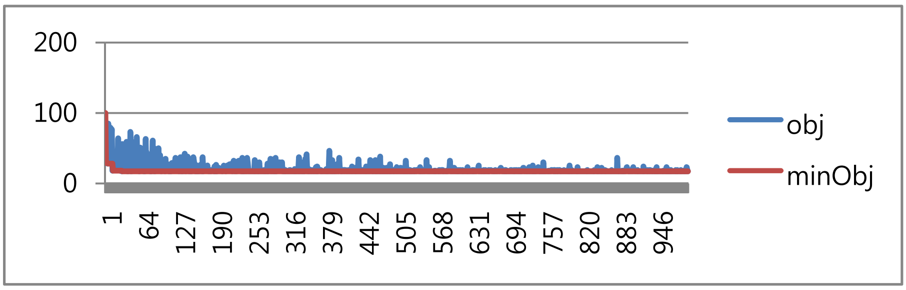

Figure 3. Experimental results of option 2

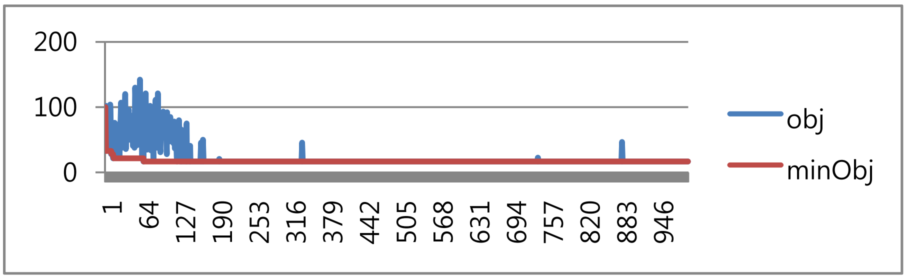

Figure 4. Experimental results of option 3

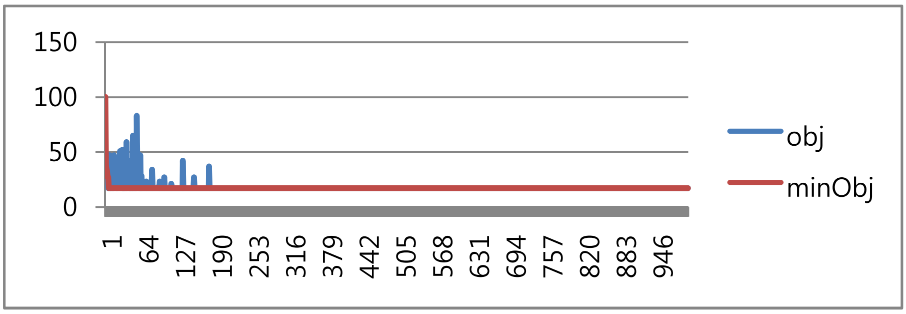

Figure 5. Experimental results of option 4