Expert Systems with Applications 40 (2013) 7607–7616

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An adaptive amoeba algorithm for constrained shortest paths

Xiaoge Zhang [a], Yajuan Zhang [a], Yong Hu [b], Yong Deng [a,c,*], Sankaran Mahadevan [c]

[a] School of Computer and Information Science, Southwest University, Chongqing 400715, China
[b] Institute of Business Intelligence and Knowledge Discovery, Guangdong University of Foreign Studies, Guangzhou 510006, China
[c] School of Engineering, Vanderbilt University, Nashville, TN 37235, USA

ARTICLE INFO

Keywords:
Constrained shortest path
Adaptive amoeba algorithm
Lagrangian relaxation
Optimization

ABSTRACT

The constrained shortest path problem (CSP) is one of the basic network optimization problems, which plays an important part in real applications. In this paper, an adaptive amoeba algorithm is combined with the *Lagrangian relaxation* algorithm to solve the CSP problem. The proposed method is divided into two steps: (1) the adaptive amoeba algorithm is modified to solve the shortest path problem (SPP) in a directed network; (2) the modified adaptive amoeba algorithm is combined with the *Lagrangian relaxation* method to solve the CSP problem. In addition, the evolving processes of the adaptive amoeba model have been detailed in the paper. Two examples are used to illustrate the efficiency of the proposed method. The results show that the proposed method can deal with the CSP problem effectively.

1. Introduction

The shortest path problem (SPP) is one of the basic network optimization problems, which has been applied in many fields, such as road navigation in transportation (Deng, Chen, Zhang, & Mahadevan, 2012; Liu, Zheng, & Cai, 2013; Mahdavi, Nourifar, Heidarzade, & Amiri, 2009; Zhang, Zhang, Deng, & Mahadevan, 2013a; Zhu, Zhang, Song, & Li, 2011), traffic routing in communication networks (Cai, Zhang, Zhou, Cao, & Tang, 2012; Cappanera & Scaparra, 2011; Junior, Nedjah, & de Macedo Mourelle, 2013), schedule planning in robotic systems (Chen, Deng, & Wu, 2013; Marzouk & Ali, 2013) and others (Deng, Liu, Hu, & Deng, 2013; Gao, Lan, Zhang, & Deng, 2013; Kang, Deng, Sadiq, & Mahadevan, 2012; Liu, Chan, Li, Zhang, & Deng, 2012;Sura & Mahadevan, 2011). The constrained shortest path problem (CSP) refers to finding the shortest path from the source node to the sink node under additional constraints for the paths in the network. Constrained shortest path applications arise frequently in practice since in many contexts a company (e.g. a package delivery firm) wants to provide its services at the lowest possible cost and yet ensure a certain level of service to its customers (Hester, Adams, & Mahadevan, 2010; Mahadevan & Overstreet, 2012). The additional constraints to the SPP problem make CSP become NP-hard (Carlyle, Royset, & Kevin Wood, 2008; Dumitrescu & Boland, 2003).

Many methods have been developed to solve CSP problems (Carlyle et al., 2008; Deng, Chan, Wu, & Wang, 2011a; Deng, Jiang, & Sadiq, 2011b; Royset, Carlyle, & Wood, 2009). These methods can be divided into two categories: one is the k shortest paths algorithm (Lefebvre, Puget, & Vilím, 2011), and the other is based on *Lagrangian relaxation* (Carlyle et al., 2008). In recent years, researchers have explored to bio-inspired algorithms due to their flexibility and simplicity (Beheshti, Shamsuddin, & Hasan, 2013; Das & Mishra, 2013; Dudek, 2013; Gómez-Gasquet, Andrés, & Lario, 2012; Lee, Lai, Chen, & Yang, 2013; Mousavi, Hajipour, Niaki, & Alikar, 2013; Zhang, Hu, Chen, & Shen, 2012; Zhang, Deng, Chan, & Zhang, 2013c; Zhang et al.; Zhang & Wang, 2013). For example, Mendes presented a genetic algorithm for Resource Constrained Project Scheduling (Mendes, Gonçalves, & Resende, 2009) in 2009. Mohemmed investigated on the application of particle swarm optimization (PSO) to solve shortest path (SP) routing problems (Mohemmed, Sahoo, & Geok, 2008).

Recently, a large amoeboid organism, the plasmodium of *Physarum polycephalum*, has been shown to be capable of solving many graph theoretical problems (Baumgarten, Ueda, & Hauser, 2010; Nakagaki & Yamada, 2000; Tero, Kobayashi, & Nakagaki, 2006; Tero et al., 2010; Zhang et al., 2013; Zhang, Zhang, Zhang, Wei, & Deng, 2013), including finding the shortest path (Miyaji & Ohnishi, 2008; Nakagaki, Yamada, & Tth, 2001; Nakagaki et al., 2007; Nakagaki et al., 2007). Inspired by this intelligent organism, a path finding mathematical model has been established (Tero, Kobayashi, & Nakagaki, 2007). Bonifaci has proved that the mathematical model can convergence to the shortest path (Bonifaci, Mehlhorn, & Varma, 2012). However, the model is only designed for undirected networks while there are many directed networks in real applications, such as transportation networks. What's more, it can only deal with SPP without constraints by now.

In this paper, we improve the amoeba model to solve the CSP problem. The proposed method can be divided into 2 parts: (1)

* Corresponding author at: School of Computer and Information Science, Southwest University, Chongqing 400715, China. Tel./fax: +86 023 68254555.
E-mail addresses: ydeng@swu.edu.cn, prof.deng@hotmail.com (Y. Deng).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.07.054

the adaptive amoeba algorithm is modified to solve the shortest path problem (SPP) in directed network; (2) the modified adaptive amoeba algorithm is combined with *Lagrangian relaxation* method to solve the CSP problem. In addition, the evolving processes have been described in the paper.

The remainder of this paper is organized as follows. Section 2 introduces the basic theory including CSP and the amoeba model. In Section 3, an adaptive amoeba algorithm for CSP problem is proposed. Section 4 gives two examples to illustrate the efficiency of the proposed method. Section 5 ends the paper with conclusions.

2. Basic theories

In this section, some basic theories including constrained shortest path problem, *Lagrangian relaxation*, amoeba model are introduced.

2.1. Constrained shortest path problem

Let $G=(V,E)$ be a network, where $V=1,\cdots,n$ is the set of nodes and $E=(i,j):i,j\in n, i\neq j$ is the set of edges. Each edge has two non-negative weights c_{ij} and t_{ij}. c_{ij} represents the generalized cost, and t_{ij} represents the constrained variable, for instance travel time. The constrained shortest path problem can be stated as the following integer programming problem:

$$\min \sum_{i,j} c_{ij}x_{ij} \tag{1}$$

subject to

$$\sum_{i,j} x_{ij} - \sum_{j,i} x_{ji} = \begin{cases} 1 & \text{for } j=1, \\ 0 & \text{for } j=2,\ldots,n-1 \\ -1 & \text{for } j=n \end{cases} \tag{2}$$

$$\sum_{i,j} t_{ij}x_{ij} \leqslant T \tag{3}$$

$$x_{ij}\in\{0,1\}, \forall i,j \tag{4}$$

where x_{ij} is binary variable, which is defined as follows:

$$x_{ij} = \begin{cases} 1 & \textit{if } x_{ij} \text{ is in optimal path}, \\ 0 & \textit{others} \end{cases} \tag{5}$$

The parameter T represents the maximum value allowed for the sum of t_{ij}. The shortest path problem is formulated by Eqs. (1), (2) and (4). Constraint (3) make that CSP problem belongs to the set on NP-hard problem.

2.2. Lagrangian relaxation

Consider the following optimization problem:

$$\begin{aligned} \min \quad & c^T x \\ s.t. \quad & Ax=b \\ & x\in X \end{aligned} \tag{6}$$

The *Lagrangian relaxation* method uses the idea of relaxing the explicit linear constraints by bringing them into the objective function with associated vector λcalled the *Lagrange multiplier*. The following problem can be treated as *Lagrangian relaxation* of the original problem (6).

$$\begin{aligned} \min \quad & c^T x + \lambda^T(Ax-b) \\ s.t. \quad & x\in X \end{aligned} \tag{7}$$

Then, the function $L(\lambda)=\min\{c^Tx+\lambda^T(Ax-b)|x\in X\}$ is the Lagrangian function.

2.3. Amoeba model

From the experiments on the amoeboid organism as described in Nakagaki, Yamada, and Toth (2001), the mechanism of tube formation can be obtained: tubes thicken in a given direction when shuttle streaming of the protoplasm persists in that direction for a certain time. It implies positive feedback between flux and tube thickness, as the conductance of the sol is greater in a thicker channel.

According to the mechanism, two rules describing the changes in the tubular structure of the amoeboid organism are: first, open-ended tubes, which are not connected between the two food sources, are likely to disappear; second, when two or more tubes connect the same two food sources, the longer tube is likely to disappear (Tero et al., 2007). With these two rules, a mathematical model for maze solving problems has been constructed.

Using the maze illustrated in Fig. 1, the model can be described as follows. Each segment in the diagram represents a section of tube. Two special nodes, which are also called food source nodes, are named N_1 and N_2, and the other nodes are denoted as N_3, N_4, N_5, and so on. The section of tube between N_i and N_j is denoted as M_{ij} . If several tubes connect the same pair of nodes, intermediate nodes will be placed in the center of the tubes to guarantee the uniqueness of the connecting segments. The variable Q_{ij} is used to express the flux through tube M_{ij} from N_i to N_j. Assuming the flow along the tube as an approximately Poiseuille flow, the flux Q_{ij} can be expressed as:

$$Q_{ij}=\frac{D_{ij}}{L_{ij}}(p_i-p_j) \tag{8}$$

where p_i is the pressure at the node N_i, D_{ij} is the conductivity of the edge M_{ij}.

Assume zero capacity at each node; by considering the conservation law of sol the following equation can be obtained:

$$\sum Q_{ij}=0 (j\neq 1,2) \tag{9}$$

For the source node N_1 and the sink node N_2 the following two equations hold

$$\sum_i Q_{i1}+I_0=0 \tag{10}$$

$$\sum_i Q_{i2}-I_0=0 \tag{11}$$

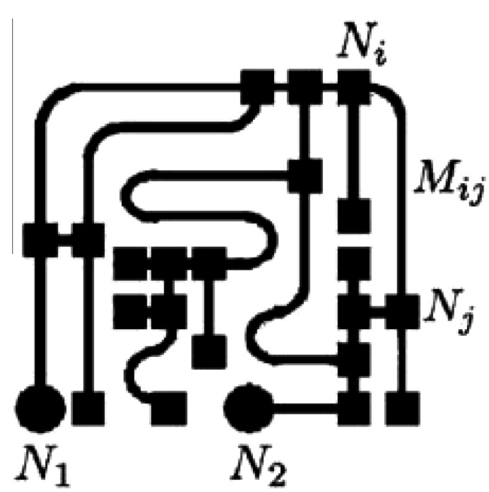


Fig. 1. Graphical maze: the source node N_1 and the sink node N_2 are indicated by solid circles and other nodes are shown by solid squares Tero et al. (2007).

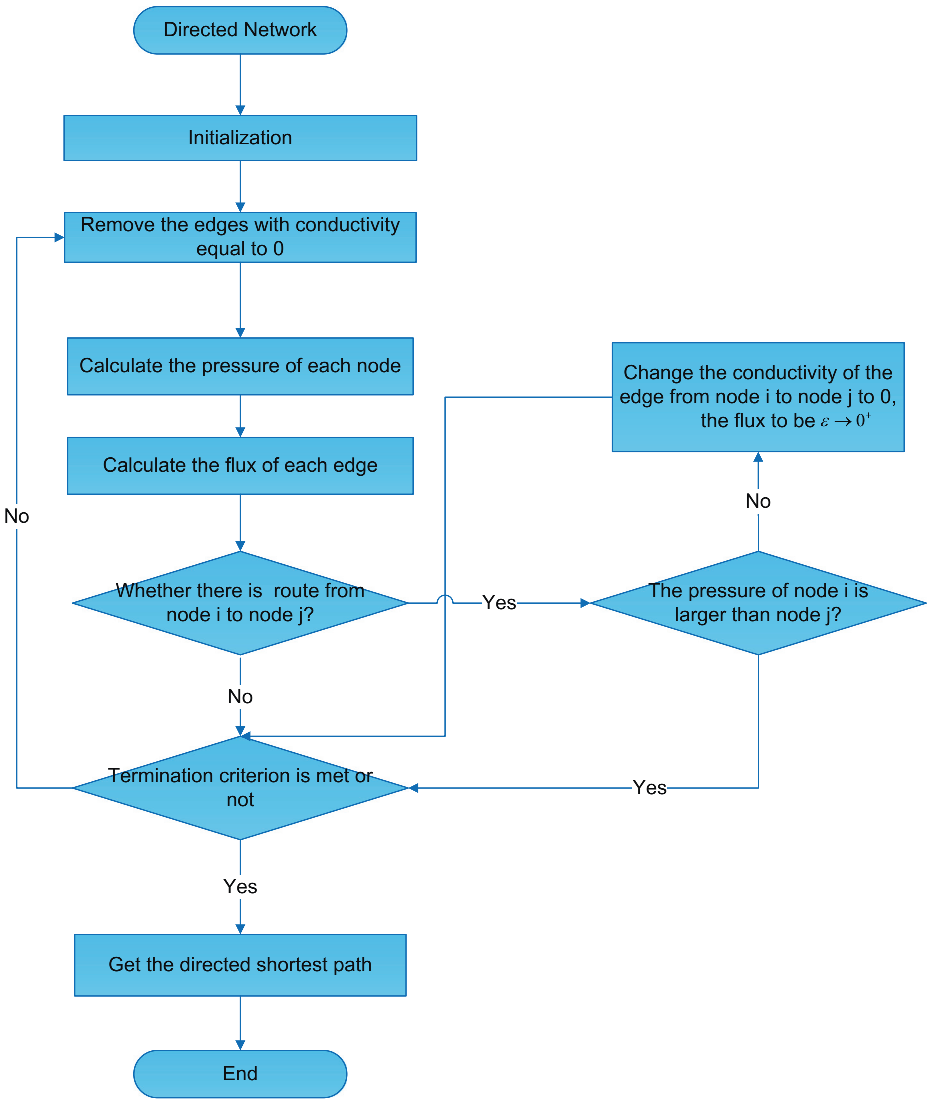


Fig. 2. The flow chart of the proposed method.

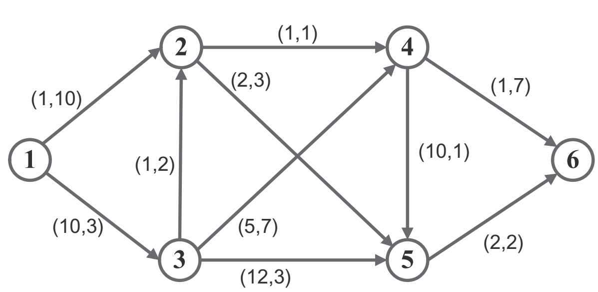


Fig. 3. Time-constrained shortest path problem.

where I_0 is the flux flowing from the source node. It can be seen that I_0 is a constant value in this model.

In order to describe such an adaptation of tubular thickness we assume that the conductivity D_{ij} changes over time according to the flux Q_{ij}. The following equation for the evolution of $D_{ij}(t)$ can be used

$$\frac{d}{dt}D_{ij} = f(|Q_{ij}|) - rD_{ij} \tag{12}$$

where r is a decay rate of the tube. The equation implies that the conductivity ends to vanish if there is no flux along the edge, while it is enhanced by the flux. f is monotonically increasing continuous function satisfying $f(0) = 0$.

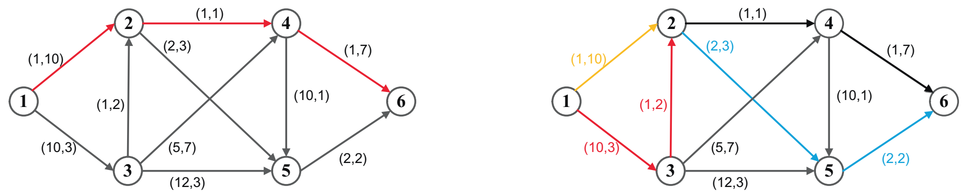
Then the network Poisson equation for the pressure can be obtained from Eqs. (8)–(10), as follows:

$$\sum_i \frac{D_{ij}}{L_{ij}}(p_i - p_j) = \begin{cases} -1 & \text{for} \quad j = 1, \\ +1 & \text{for} \quad j = 2, \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

By setting $p_2 = 0$ as a basic pressure level, all p_i can be determined by solving Eq. 13 and Q_{ij} can also be obtained.

Since f is monotonically increasing continuous function satisfying $f(0) = 0$ in Eq. (12), $f(Q) = |Q|$ is used in this paper and the reason lies that it has been certified in Tero et al. (2007) that $f(Q) = |Q|$ is able to find the shortest path with rapid speed when compared with other functions. With the flux calculated, the conductivity can be derived, where Eq. (14) is used instead of Eq. (12), adopting the functional form $f(Q) = |Q|$.

$$\frac{D_{ij}^{n+1} - D_{ij}^{n}}{\delta t} = |Q| - D_{ij}^{n+1} \tag{14}$$



(a) constrained shortest path problem (red lines shows the shortest path for $\lambda = 0$)

(b) modified cost $c_{ij} + \lambda t_{ij}$ with $\lambda = 2$ (red lines denote the shortest path)

Fig. 4. The constrained shortest path problem.

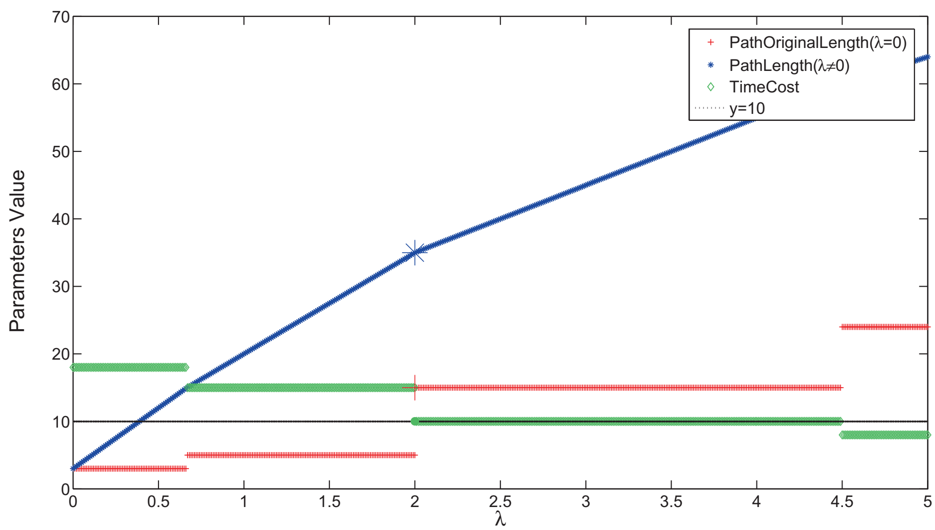


Fig. 5. The constrained shortest path solution changes with λ. "PathLength" refers to the total length changes with λ when $c_{ij} + \lambda t_{ij}$ $(\lambda \neq 0)$ is used as path length. "PathOriginalLength" means that the total path length changes with λ when c_{ij} $(\lambda = 0)$ is calculated as path length. "TimeCost" shows the cost time of the found shortest path when different λ is used. "y = 10" limits the total time to 10 units.

3. Proposed method

With experiments of path finding process of the mathematical model described in Section 2.3, it can be seen that the bio-inspired model can handle the shortest path problem. However, it is observed that the original amoeba algorithm can only find the shortest path in the undirected network. In this section, the proposed method is detailed to solve the CSP problem in a directed network. As a consequence, there are two issues that need to be solved. One is how to solve shortest path problem in the directed network, the other is how to solve CSP problem. As a result, the proposed method is composed of two parts.

3.1. Shortest path problem in directed network

Let $G = (N, E, W)$ be a directed network, where N denotes a set of n nodes, E denotes an edge set with m directed edges, and L denotes a weight set for E. Assume that there is only one directed edge between two nodes in G. Given a source node s and a target node t, the directed shortest path problem can be defined as how to find a path from s to t, which only consists of directed edges of E, with the minimum sum of weights on the edges.

In the undirected networks, every edge is bidirectional. In original amoeba model, the arc starts from the node with higher pressure and ends in the node with lower pressure. There is a positive feedback mechanism: the higher the pressure, the more the flux. The more the flux, the higher the pressure. This positive feedback make the shortest path fade in when the iteration continues. However, in a directed network, only considering the pressure of each node is not enough. The directivity of each arc should be taken into consideration. In the proposed method, we adopt a novel method to solve this issue.

The flow chart of the proposed method for finding the directed shortest path from s to t in G is shown in Fig. 2.

- Step 1: Initialize tube length and conductivity for G. In G, each tube has two attributes, its length $L_{i,j}$ and conductivity $D_{i,j}$. The length $L_{i,j}$ is initialized according to the weight of the edge $L_{i,j}$ in G, while the conductivity is assigned equally as 0.5. p_i states the pressure of node i. At the same time, the basic information of the directed network is constructed.
- Step 2: Remove the edges with conductivity equal to 0 in order to reduce the computing time.
- Step 3: Calculate the pressure of each node using its current conductivity and length according to Eq. (13).
- Step 4: Calculate the pressure and conductivity of each node according to Eq. (8) and Eq. (14) during the next iteration.
- Step 5: On the basis of the information of the directed network, it can be determined that whether there is a path from node i to node j. If it exists, R_{ij} in the mark matrix will be 1. Otherwise, R_{ij} is 0. During each iteration, the pressure of each node can be obtained. If there is a path from node i to node j $(R_{ij} = 1)$, check whether the pressure of node i is higher than that of node j in

The procedures of the proposed method

// L is the distance matrix, L_{ij} represents the edge length from node i to node j.

// s is the start node, e is the end node

$D_{ij} \leftarrow (0,1]\ (\forall i, j = 1,2,\ldots,N \wedge i \neq j)$

$R:\ if\ L_{ij} = 1, R_{ij} \leftarrow 1;\ else\ R_{ij} \leftarrow 0$

$Q_{ij} \leftarrow 0\ (\forall i, j = 1,2,\ldots.N \wedge i \neq j)$

$p_i \leftarrow 0 \quad \forall i = 1,2,\ldots,N)$

C_{ij} is the value of constraint factor of each edge

$\lambda = 0, \lambda_{\max}$ (λ *is Lagrange multiplier*, $\lambda_{\max}$ *is a predefined upper bound of* λ)

while $\lambda \leq \lambda_{\max}$

  $L_{ij} = L_{ij} + \lambda * C_{ij}$

  repeat

    $p_e \leftarrow 0$ // pressure at the end node e

    Calculate the pressure p_i of each node using Eq. (13)

$$\sum_i \frac{D_{ij}}{L_{ij}}(p_i - p_j) = \begin{cases} -1 & for\ j = s, \\ +1 & for\ j = e, \\ 0 & otherwise \end{cases}$$

    $Q_{ij} \leftarrow \frac{D_{ij}}{L_{ij}} * (p_i - p_j)$ // using Eq. (8)

    $D_{ij} \leftarrow f(|Q_{ij}|) + D_{ij}$ // using Eq. (14)

    for $(i = 1)$ to (N)

      for $(j = 1)$ to (N)

        if $R_{ij} == 1$

          if $p_i < p_j$

            $D_{ij} = 0; Q_{ij} = eps; D_{ji} = 0; Q_{ji} = eps$

          end

        end

      end

    end

  until a termination criterion is met

  $\lambda = \lambda + 0.5$ (0.5 *is the step size*);

  Calculate the corresponding value of the constraints.

end

Fig. 6. The procedures of the proposed method.

the model of *Physarum polycephalum*. If p_i is smaller than p_j, the following operations are processed: change the conductivity D_{ij} value of edge from node i to node j to be 0, the flux Q_{ij} to be $\varepsilon \to 0^+$. Otherwise, the original data such as D_{ij}, Q_{ij} is kept.

- Step 6: Check whether termination criterion is met or not. The termination criterion is set as the conductivity of each tube remaining unchanged. If it is satisfied, tubes with conductivity approximating 1 compose the directed shortest path. Otherwise, go to Step 2 and repeat until convergence.

3.2. Constrained shortest paths

Consider the network shown in Fig. 3. Each arc has two attributes: cost c_{ij} and traversal time t_{ij}. The numbers along each arc shows its cost and traversal time respectively. For example, the numbers (1,10) along the arc between node 1 and node 2 means the cost along the arc is 1, and the traversal time is 10. Assume that we wish to find the shortest path from source node 1 to sink node

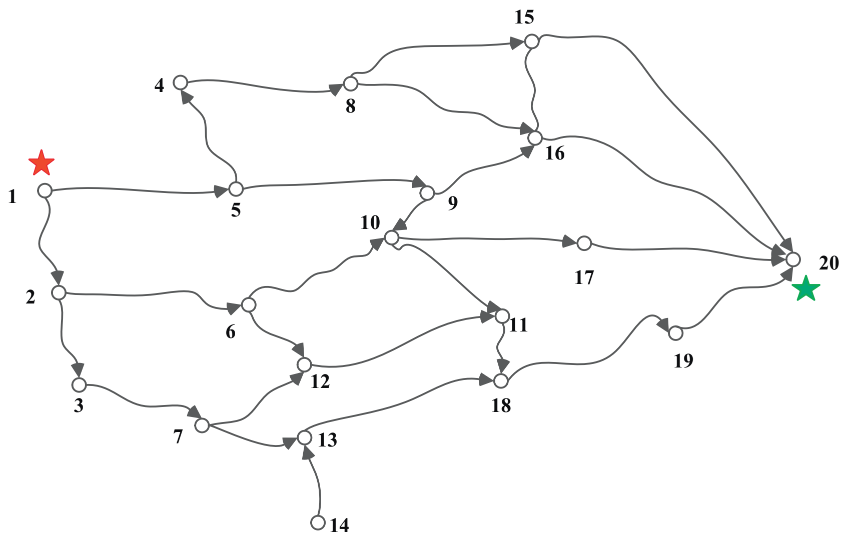


Fig. 7. A directed transportation network with 20 nodes.

Table 1
Attribute value of each edge.

Edge	Length	Toll	Edge	Length	Toll
1 → 2	120	60	9 → 10	20	60
1 → 5	90	50	9 → 16	90	40
2 → 3	100	40	10 → 11	90	50
2 → 6	90	60	10 → 17	70	40
3 → 7	90	50	11 → 18	70	60
4 → 8	70	40	12 → 11	50	50
5 → 4	120	60	13 → 18	140	40
5 → 9	80	50	14 → 13	100	60
6 → 10	40	20	15 → 20	150	50
6 → 12	60	40	15 → 16	40	40
7 → 12	80	60	16 → 20	80	60
7 → 13	60	50	17 → 20	60	50
8 → 15	70	40	18 → 19	40	40
8 → 16	100	40	19 → 20	50	40

6, but we wish to restrict our choice of paths to those that require no more than $T = 10$ time units for traversal.

This is a typical CSP problem. In this paper, we adopt an indirect approach by combining time and cost into a single generalized cost instead of solving this problem directly. Instead of setting a limit on the total time, a "cost" is set on each arc, which is proportional to the time that it takes to pass through that arc. Thus, we solve the CSP problem with the modified cost $c_{ij} + \lambda t_{ij}$. For instance, if $\lambda = 0$, this problem will become the shortest path problem only considering the original costs c_{ij} and the shortest path is 1–2–4–6 as shown in Fig. 4(a). The total length of the path is 3 while its cost time is 18. If we assume that $\lambda = 2$ and solve the constrained shortest path problem, Fig. 4(b) displays the shortest path with the modified costs $c_{ij} + 2 * t_{ij}$. The shortest path is 1–2–5–6 and 1–3–2–5–6. Their path length is 35 with respect to the modified costs $c_{ij} + 2 * t_{ij}$. However, the cost time of path 1–2–5–6 is 15 while the cost time of 1–3–2–5–6 is 10. The total length, the cost time

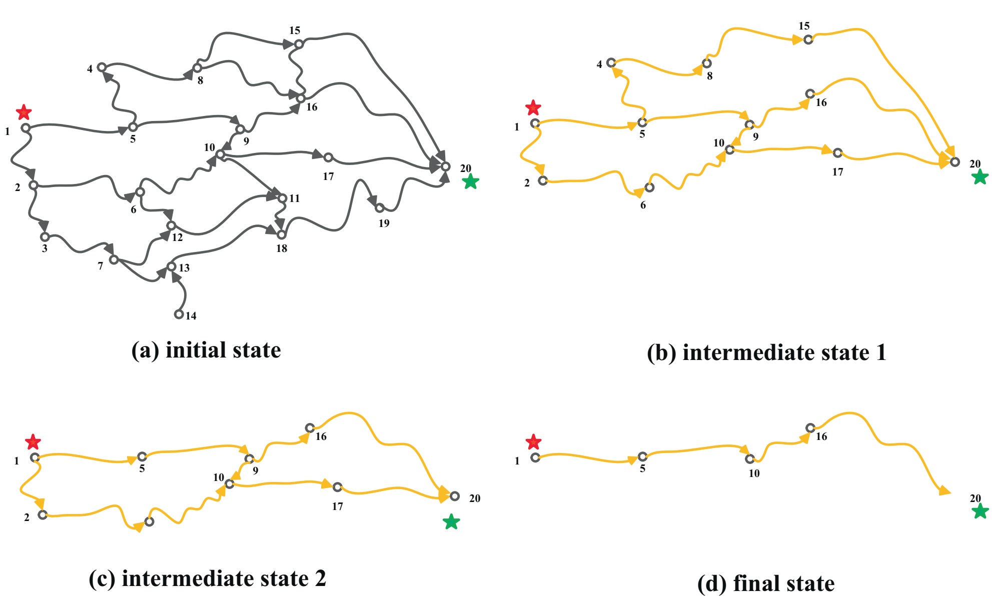


Fig. 8. Panels (a)–(d) illustrate the amoeba model path finding process: (a) initial state; (b) intermediate state 1; (c) intermediate state 2; (d) final state. Starting and ending nodes are indicated by the red pentagram and green pentagram respectively.

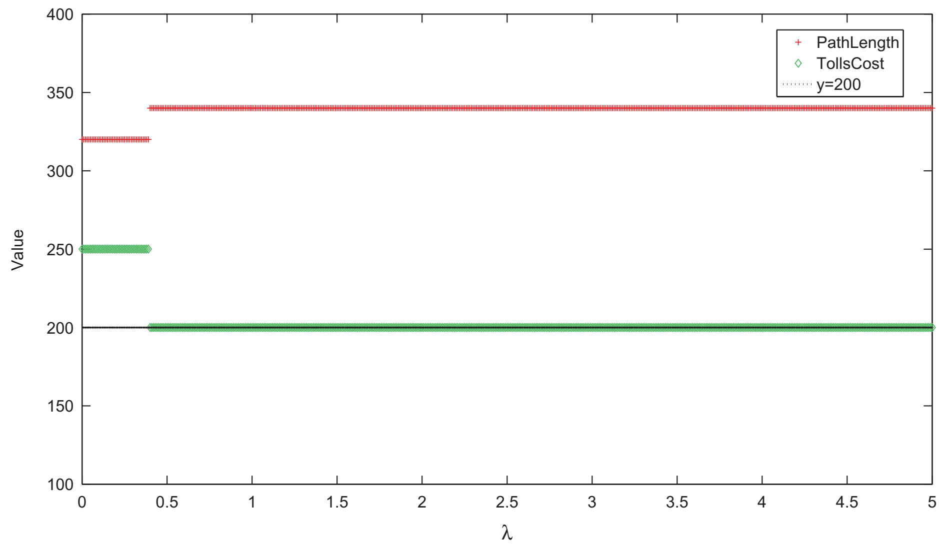


Fig. 9. The constrained shortest path problem changes with the variable λ. "PathLength" displays the path length of the corresponding shortest path. "Toll Cost" shows the total tolls of each shortest path. "y = 200" shows the constraint.

Table 2
The generated cost value of the random network.

Edge	Cost	Edge	Cost	Edge	Cost
1 → 3	10.7	3 → 21	9.64	7 → 28	9.86
2 → 3	5.24	11 → 21	11.2	9 → 28	12.3
1 → 4	10.4	14 → 21	11.5	24 → 28	11.3
4 → 5	15.4	17 → 21	11.2	25 → 28	14.4
3 → 6	8.89	2 → 22	2.68	2 → 29	10.7
4 → 6	5.11	21 → 22	7.75	7 → 29	14.1
1 → 7	15.5	2 → 23	9.60	9 → 29	9.09
5 → 7	8.32	15 → 23	8.14	11 → 29	3.92
2 → 8	13.6	16 → 23	9.79	22 → 29	5.58
3 → 8	13.4	4 → 24	6.28	25 → 29	9.48
5 → 8	8.52	7 → 24	11.3	5 → 30	6.05
7 → 8	6.22	8 → 24	6.11	9 → 30	11.9
3 → 9	8.25	10 → 24	8.85	18 → 30	15.0
5 → 9	10.9	12 → 24	11.5	1 → 31	6.52
8 → 9	8.93	13 → 24	7.74	3 → 31	13.3
4 → 10	12.1	15 → 24	10.2	7 → 31	10.3
1 → 11	10.2	21 → 24	12.3	8 → 31	10.3
10 → 12	8.24	10 → 25	11.5	11 → 31	11.1
11 → 13	11.6	16 → 25	13.2	13 → 31	8.71
12 → 13	9.74	19 → 25	6.17	14 → 31	9.80
10 → 14	9.49	22 → 25	12.6	19 → 31	6.35
12 → 15	9.55	2 → 26	8.65	29 → 31	11.1
13 → 15	9.20	3 → 26	16.5	3 → 32	12.9
1 → 16	5.87	7 → 26	11.9	4 → 32	10.9
3 → 16	13.1	8 → 26	15.3	13 → 32	11.8
7 → 17	9.16	11 → 26	8.19	15 → 32	8.71
15 → 17	8.09	14 → 26	6.99	18 → 32	6.43
5 → 18	3.04	15 → 26	10.4	23 → 32	7.32
9 → 18	10.0	17 → 26	17.3	28 → 32	11.3
10 → 18	4.56	23 → 26	8.92	3 → 33	12.3
2 → 19	8.34	2 → 27	9.13	6 → 33	7.94
3 → 19	9.04	6 → 27	5.99	9 → 33	7.01
7 → 19	8.62	8 → 27	11.8	10 → 33	13.1
2 → 20	8.16	9 → 27	5.32	12 → 33	12.9
3 → 20	9.49	14 → 27	14.3	19 → 33	5.89
6 → 20	7.05	19 → 27	6.20	20 → 33	7.05
8 → 20	4.45	20 → 27	13.8	23 → 33	11.5
2 → 21	10.9	6 → 28	7.51		

will change with λ. Fig. 5 illustrates how the constrained shortest path solution changes with λ.

As can be seen in Fig. 5, the solution to the constrained shortest path problem can be easily obtained. This figure displays how the different variables change with λ. It can be seen that there is a solution to the above CSP problem when $\lambda = 2$. In this way, the constrained shortest path can be obtained as 1–3–2–5–6.

3.3. The main steps of the proposed method

The main steps of the proposed method can be summarized as follows. First of all, given the constrained shortest path problem, initialize every variable, such as the initialized conductivity matrix D, the length of each edge L_{ij}, the original flux matrix Q and so on. Secondly, taking the constraint factor into account, we convert the CSP problem into a shortest path problem by *Lagrangian relaxation* method. Next, the check procedure is processed after each iteration so as the amoeba model can be applied to directed network. Finally, the procedure is over when the termination criterion is met. Fig. 6 shows the procedures of the proposed method.

4. Examples

Two examples are contained in order to illustrate the proposed method.

Example 4.1. Fig. 7 shows a directed transportation network with 20 nodes. Each edge has 2 attributes: length (L), toll (T). Table 1 displays the attribute value of every edge. Suppose that the shortest path between node 1 and node 20 needs to be found under the constraint that the total toll is less than 200.

Based on the proposed method, we solve this problem with the modified costs $L_{ij} + \lambda T_{ij}$, where $\lambda \geqslant 0$. The initialized conductivity matrix is set 0.5, the pressure of each node is 0, the original flux of each node Q is set with equal to 0. The process of the Physarum algorithm is depicted in Fig. 8 when $\lambda = 1$. Fig. 8(a) represents the initial state of the network, before the implementation of the algorithm. The next two figures, Fig. 8(b) and Fig. 8(c) show the intermediate states of the network when some edges has been cut off by the algorithm. The final path between two specified nodes is derived in Fig. 8(d).

The results are different for different values of λ as shown in Fig. 9.

From Fig. 9, the constrained shortest path can be obtained easily. The solution to this CSP problem is 1–5–9–16–20. The tolls of the path is 200 and the length of the path is 340.

Table 3
The generated delay value of the random network.

Edge	Delay	Edge	Delay	Edge	Delay
1 → 3	15.50	3 → 21	11.4	7 → 28	7.60
2 → 3	12.58	11 → 21	9.41	9 → 28	9.60
1 → 4	6.076	14 → 21	10.8	24 → 28	14.0
4 → 5	11.02	17 → 21	14.7	25 → 28	8.23
3 → 6	18.30	2 → 22	12.0	2 → 29	7.45
4 → 6	19.10	21 → 22	9.26	7 → 29	17.5
1 → 7	9.810	2 → 23	6.50	9 → 29	10.9
5 → 7	9.385	15 → 23	10.3	11 → 29	7.40
2 → 8	14.46	16 → 23	17.7	22 → 29	12.3
3 → 8	14.25	4 → 24	10.5	25 → 29	3.01
5 → 8	6.377	7 → 24	4.20	5 → 30	11.0
7 → 8	14.89	8 → 24	4.61	9 → 30	11.3
3 → 9	13.10	10 → 24	7.33	18 → 30	10.5
5 → 9	9.089	12 → 24	8.36	1 → 31	12.5
8 → 9	7.638	13 → 24	8.19	3 → 31	11.3
4 → 10	6.558	15 → 24	12.2	7 → 31	8.99
1 → 11	7.571	21 → 24	9.41	8 → 31	13.1
10 → 12	14.31	10 → 25	7.48	11 → 31	13.7
11→13	7.735	16 → 25	6.78	13 → 31	9.79
12 → 13	4.865	19 → 25	10.3	14 → 31	9.34
10 → 14	9.275	22 → 25	4.11	19 → 31	10.0
12 → 15	10.93	2 → 26	6.37	29 → 31	12.4
13 → 15	9.909	3 → 26	12.4	3 → 32	11.4
1 → 16	11.88	7 → 26	6.82	4 → 32	11.8
3 → 16	13.32	8 → 26	9.18	13 → 32	6.91
7 → 17	10.23	11 → 26	9.16	15 → 32	10.9
15 → 17	6.659	14 → 26	3.84	18 → 32	11.5
5 → 18	14.59	15 → 26	7.52	23 → 32	7.17
9 → 18	11.11	17 → 26	11.5	28 → 32	9.56
10 → 18	13.35	23 → 26	10.1	3 → 33	15.0
2 → 19	10.09	2 → 27	13.3	6 → 33	8.54
3 → 19	13.30	6 → 27	9.10	9 → 33	6.47
7 → 19	10.25	8 → 27	9.21	10 → 33	9.17
2 → 20	7.773	9 → 27	9.14	12 → 33	9.25
3 → 20	17.05	14 → 27	7.06	19 → 33	14.8
6 → 20	12.24	19 → 27	8.39	20 → 33	9.31
8 → 20	12.66	20 → 27	12.8	23 → 33	8.66
2 → 21	5.793	6 → 28	9.93		

Example 4.2. The Delay Constrained Least Cost (DCLC) problem is to find the least cost path in a graph while keeping the path delay below a specified value. This problem has been widely used in routing in computer networks (Jia & Varaiya, 2006; Kun, 2005; Leela, Thanulekshmi, & Selvakumar, 2011; Rocha, Sousa, Cortez, & Rio, 2011; Zhengying, Bingxin, & Erdun, 2001). The DCLC problem can be characterized as a directed, connected graph $G = (V, E)$, where V is the set of nodes and $E \subset V \times V$. Corresponding to each link/edge (i,j) connecting nodes i and j, there are nonnegative numbers c_{ij} representing the cost of the link/edge and another nonnegative delays d_{ij} representing transmission delay from node i to node j.

Given a source node $s \in V$, an ending node $e \in V$, and a transmission delay constraint Δ_{delay}, the following equation can be obtained:

$$\min_{p \in P(s,e)} c(p) \text{ such that } d(p) \leqslant \Delta_{delay} \tag{15}$$

where $P(s,e)$ is the set of all possible paths from node s to node e. As a consequence, DCLC is a typical CSP problem.

In this paper, a random directed network is generated using the Waxman model (Waxman, 1988), and the link probability between node i and node j is defined as follows:

$$P(i,j) = \gamma e^{-d(i,j)/(\beta L_{\max})}, 0 \ < \gamma, \quad \beta \leqslant 1 \tag{16}$$

where $d(i,j)$ expresses the Euclidean distance between node i and node j, and $L_{\max}$ is the maximum distance between any two nodes. γ and β are two parameters controlling the link probability. In this example, γ and β are 0.3 and 0.9 respectively. The edges's costs and delays are randomly generated using normal distribution with mean value equal to 10 and standard deviation equal to 3. As the test networks are randomly generated (not only topology but also the edge costs and delays), it is necessary to set an appropriate delay constraint (threshold) according to the delay values of the network. In order to prevent the DCLC problem from becoming trivial, the delay constraint is chosen as (Jia & Varaiya, 2006):

$$\Delta_{delay} = 0.75 * p_{ld} + 0.25 * p_{lc} \tag{17}$$

where p_{ld} is the delay of the least delay path, and p_{lc} is the cost of the least cost path. The path with the least cost and the path with the least delay can be found using the amoeba algorithm.

The data of the generated network is shown in Tables 2 and 3. Based on the data, it can be determined that the least cost path is $1 \to 3 \to 33$ while the least delay path is $1 \to 4 \to 10 \to 33$. According to Eq. (17), the constraint is $\Delta_{delay} = 0.75 * 21.8135 + 0.25 * 23.13 = 22.1438$.

As can be seen in Fig. 10, the solution to this DCLC problem can be achieved. The answer is $1 \to 4 \to 10 \to 33$ and the transmission delay is 21.8135, which is less than 22.1438.

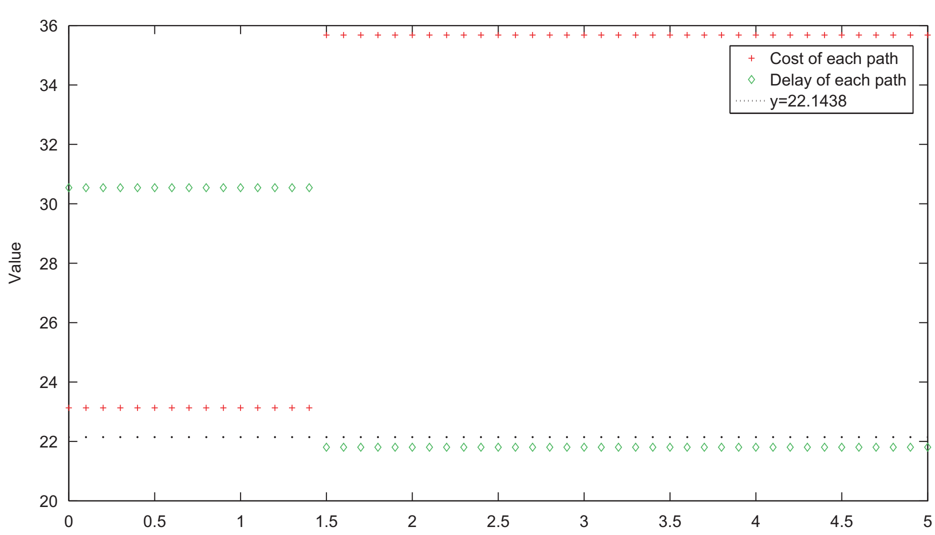


Fig. 10. The Delay Constrained Least Cost (DCLC) problem changes with the variable λ. "Cost of each path" denotes the cost of corresponding path. "Delay of each path" represents the transmission delay of selected path. "y = 22.1438" displays the constraint Δ_{delay}.

5. Conclusion

The constrained shortest path (CSP) problem plays an important role in many network problems. In this paper, an adaptive amoeba algorithm named *Physarum polycephalum* is firstly modified to solve the shortest path problem in a directed network. Then the modified model is combined with *Lagrangian relaxation* method to solve the CSP problem. To the best of our knowledge, this is the first time that the adaptive amoeba algorithm is applied to this optimization problem. Examples in route selection in transportation and computer networks are used to illustrate the proposed method. The results demonstrate that the proposed method is able to deal with CSP problem.

Acknowledgement

The Ph.D candidates of the corresponding author in Shanghai Jiao Tong University, Xiaoyan Su and Peida Xu, provide valuable discussions on bio-inspired optimization. The work is partially supported by National Natural Science Foundation of China, Grant Nos. 61174022 and 71271061, Chongqing Natural Science Foundation (for Distinguished Young Scholars), Grant No. CSCT, 2010BA2003, National High Technology Research and Development Program of China (863 Program) (No. 2012AA041101), Science and Technology Planning Project of Guangdong Province, China (2010B010600034, 2012B091100192), Business Intelligence Key Team of Guangdong University of Foreign Studies (TD1202), Doctor Funding of Southwest University Grant No. SWU110021, China State Key Laboratory of Virtual Reality Technology and Systems, the Fundamental Research Funds for the Central Universities Grant No. XDJK2013D010.

References

Baumgarten, W., Ueda, T., & Hauser, M. (2010). Plasmodial vein networks of the slime mold Physarum polycephalum form regular graphs. *Physical Review E, 82*(4), 046113.

Beheshti, Z., Shamsuddin, S. M. H., & Hasan, S. (2013). MPSO: Median-oriented Particle Swarm Optimization. *Applied Mathematics and Computation, 219*(11), 5817–5836.

Bonifaci, V., Mehlhorn, K., & Varma, G. (2012). Physarum can compute shortest paths. *Journal of Theoretical Biology, 309*(21), 121–133.

Cai, K., Zhang, J., Zhou, C., Cao, X., & Tang, K. (2012). Using computational intelligence for large scale air route networks design. *Applied Soft Computing, 12*(9), 2790–2800.

Cappanera, P., & Scaparra, M. (2011). Optimal allocation of protective resources in shortest-path networks. *Transportation Science, 45*(1), 64–80.

Carlyle, W., Royset, J., & Kevin Wood, R. (2008). Lagrangian relaxation and enumeration for solving constrained shortest-path problems. *Networks, 52*(4), 256–270.

Chen, S., Deng, Y., & Wu, J. (2013). Fuzzy sensor fusion based on evidence theory and its application. *Applied Artificial Intelligence, 27*(3), 235–248.

Das, K. N., & Mishra, R. (2013). Chemo-inspired genetic algorithm for function optimization. *Applied Mathematics and Computation, 220*, 394–404.

Deng, Y., Chan, F., Wu, Y., & Wang, D. (2011a). A new linguistic MCDM method based on multiple-criterion data fusion. *Expert Systems with Applications, 38*(6), 6985–6993.

Deng, Y., Chen, Y., Zhang, Y., & Mahadevan, S. (2012). Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Applied Soft Computing, 12*(3), 1231–1237.

Deng, Y., Jiang, W., & Sadiq, R. (2011b). Modeling contaminant intrusion in water distribution networks: A New similarity-based DST method. *Expert Systems with Applications, 38*(1), 571–578.

Deng, X., Liu, Q., Hu, Y., & Deng, Y. (2013). TOPPER: Topology prediction of transmembrane protein based on evidential reasoning. *The Scientific World Journal*.

Dudek, G. (2013). Genetic algorithm with binary representation of generating unit start-up and shut-down times for the unit commitment problem. *Expert Systems with Applications, 40*(15), 6080–6086.

Dumitrescu, I., & Boland, N. (2003). Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem. *Networks, 42*(3), 135–153.

Gao, C., Lan, X., Zhang, X., & Deng, Y. (2013). A bio-inspired methodology of identifying influential nodes in complex networks. *PLOS ONE, 8*(6), e66732.

Gómez-Gasquet, P., Andrés, C., & Lario, F. (2012). An agent-based genetic algorithm for hybrid flowshops with sequence dependent setup times to minimise makespan. *Expert Systems with Applications, 39*(9), 8095–8107.

Hester, P., Adams, K., & Mahadevan, S. (2010). Examining metrics and methods for determining critical facility system effectiveness. *International Journal of Critical Infrastructures, 6*(3), 211–224.

Jia, Z., & Varaiya, P. (2006). Heuristic methods for delay constrained least cost routing using κ-shortest-paths. *IEEE Transactions on Automatic Control, 51*(4), 707–712.

Junior, L. S., Nedjah, N., & de Macedo Mourelle, L. (2013). Routing for applications in NoC using ACO-based algorithms. *Applied Soft Computing, 13*(6), 2224–2231.

Kang, B., Deng, Y., Sadiq, R., & Mahadevan, S. (2012). Evidential cognitive maps. *Knowledge-Based Systems, 35*, 77–86.

Kun, Z., Heng, W., & Feng-Yu, L. (2005). Distributed multicast routing for delay and delay variation-bounded Steiner tree using simulated annealing. *Computer Communications, 28*(11), 1356–1370.

Lee, H.-L., Lai, T.-H., Chen, W.-L., & Yang, Y.-C. (2013). An inverse hyperbolic heat conduction problem in estimating surface heat flux of a living skin tissue. *Applied Mathematical Modelling, 37*(5), 2630–2643.

Leela, R., Thanulekshmi, N., & Selvakumar, S. (2011). Multi-constraint Qos unicast routing using genetic algorithm (MURUGA). *Applied Soft Computing, 11*(2), 1753–1761.

Lefebvre, M., Puget, J., Vilím, P. (2011). Route finder: efficiently finding *k* shortest paths using constraint programming. Principles and Practice of Constraint Programming–CP (vol. 2011, pp. 42–53).

Liu, J., Chan, F. T., Li, Y., Zhang, Y., & Deng, Y. (2012). A new optimal consensus method with minimum cost in fuzzy group decision. *Knowledge-Based Systems, 35*, 357–360.

Liu, W., Zheng, Z., & Cai, K.-Y. (2013). Bi-level programming based real-time path planning for unmanned aerial vehicles. *Knowledge-Based Systems, 44*, 34–47.

Mahadevan, S., & Overstreet, J. (2012). Use of warranty and reliability data to inform call center staffing. *Quality Engineering, 24*(3), 386–399.

Mahdavi, I., Nourifar, R., Heidarzade, A., & Amiri, N. (2009). A dynamic programming approach for finding shortest chains in a fuzzy network. *Applied Soft Computing, 9*(2), 503–511.

Marzouk, M., & Ali, H. (2013). Modeling safety considerations and space limitations in piling operations using agent based simulation. *Expert Systems with Applications, 40*(12), 4848–4857.

Mendes, J., Gonçalves, J., & Resende, M. (2009). A random key based genetic algorithm for the resource constrained project scheduling problem. *Computers & Operations Research, 36*(1), 92–109.

Miyaji, T., & Ohnishi, I. (2008). Physarum can solve the shortest path problem on Riemannian surface mathematically rigorously. *International Journal of Pure and Applied Mathematics, 47*(3), 353–369.

Mohemmed, A., Sahoo, N., & Geok, T. (2008). Solving shortest path problem using particle swarm optimization. *Applied Soft Computing, 8*(4), 1643–1653.

Mousavi, S. M., Hajipour, V., Niaki, S. T. A., & Alikar, N. (2013). Optimizing multi-item multi-period inventory control system with discounted cash flow and inflation: Two calibrated meta-heuristic algorithms. *Applied Mathematical Modelling, 37*(4), 2241–2256.

Nakagaki, T., Iima, M., Ueda, T., Nishiura, Y., Saigusa, T., Tero, A., Kobayashi, R., & Showalter, K. (2007). Minimum-risk path finding by an adaptive amoebal network. *Physical Review Letters, 99*(6), 1–4.

Nakagaki, T., Iima, M., Ueda, T., Nishiura, Y., Saigusa, T., Tero, A., Kobayashi, R., & Showalter, K. (2007). Minimum-risk path finding by an adaptive amoebal network. *Physical Review Letters, 99*(6), 68104.

Nakagaki, T., Yamada, H., & Toth, A. (2001). Path finding by tube morphogenesis in an amoeboid organism. *Biophysical Chemistry, 92*(1–2), 47–52.

Nakagaki, T., Yamada, H., & Tth, A. (2000). Maze-solving by an amoeboid organism. *Nature, 407*(6803) (470–470).

Nakagaki, T., Yamada, H., & Tth, A. (2001). Path finding by tube morphogenesis in an amoeboid organism. *Biophysical Chemistry, 92*(1–2).

Rocha, M., Sousa, P., Cortez, P., & Rio, M. (2011). Quality of service constrained routing optimization using evolutionary computation. *Applied Soft Computing, 11*(1), 356–364.

Royset, J., Carlyle, W., & Wood, R. (2009). Routing military aircraft with a constrained shortest-path algorithm. *Military Operations Research, 14*(3), 31–52.

Sura, V., & Mahadevan, S. (2011). Modeling of vertical split rim cracking in railroad wheels. *Engineering Failure Analysis, 18*(4), 1171–1183.

Tero, A., Kobayashi, R., & Nakagaki, T. (2006). Physarum solver: A biologically inspired method of road-network navigation. *Physica A, 363*(1), 115–119.

Tero, A., Kobayashi, R., & Nakagaki, T. (2007). A mathematical model for adaptive transport network in path finding by true slime mold. *Journal of Theoretical Biology, 244*(4), 553–564.

Tero, A., Takagi, S., Saigusa, T., Ito, K., Bebber, D. P., Fricker, M. D., Yumiki, K., Kobayashi, R., & Nakagaki, T. (2010). Rules for biologically inspired adaptive network design. *Science, 327*(5964), 439–442.

Waxman, B. (1988). Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications, 6*(9), 1617–1622.

Zhang, H., Deng, Y., Chan, F. T., & Zhang, X. (2013c). A modified multi-criterion optimization genetic algorithm for order distribution in collaborative supply chain. *Applied Mathematical Modelling, 37*(14–15), 7855–7864.

Zhang, X., Deng, Y., Chan, F. T. S., Xu, P., Mahadevan, S., & Hu, Y. IFSJSP: A novel methodology for the Job-Shop Scheduling Problem based on intuitionistic fuzzy sets. *International Journal of Production Research*. http://dx.doi.org/10.1080/00207543.2013.793425.

Zhang, X., Huang, S., Hu, Y., Zhang, Y., Mahadevan, S., & Deng, Y. (2013). Solving 0–1 knapsack problems based on amoeboid organism algorithm. *Applied Mathematics and Computation, 219*(19), 9959–9970.

Zhang, W., Hu, T., Chen, J., & Shen, L. (2012). BioDKM: Bio-inspired domain knowledge modeling method for humanoid delivery robots planning. *Expert Systems with Applications, 39*(1), 663–672.

Zhang, L., & Wang, N. (2013). A modified DNA genetic algorithm for parameter estimation of the 2-Chlorophenol oxidation in supercritical water. *Applied Mathematical Modelling, 37*(3), 1137–1146.

Zhang, Y., Zhang, Z., Deng, Y., & Mahadevan, S. (2013a). A biologically inspired solution for fuzzy shortest path problems. *Applied Soft Computing, 13*(5), 2356–2363.

Zhang, X., Zhang, Z., Zhang, Y., Wei, D., & Deng, Y. (2013). Route selection for emergency logistics management: A bio-inspired algorithm. *Safety Science, 54*, 87–91.

Zhengying, W., Bingxin, S., & Erdun, Z. (2001). Bandwidth-delay-constrained least-cost multicast routing based on heuristic genetic algorithm. *Computer Communications, 24*(7), 685–692.

Zhu, Y., Zhang, T., Song, J., & Li, X. (2011). A new hybrid navigation algorithm for mobile robots in environments with incomplete knowledge. *Knowledge-Based Systems, 27*, 302–313.