Solving the Maximum Flow Problem by a Modified Adaptive Amoeba Algorithm

Zhixuan Wang, Daijun Wei*, *Member, IAENG,*

***Abstract*—Maximum flow problem is one of the most fundamental network optimization problems. Recently, experimental observations showed that an amoeboid organism, *Physarum polycephalum*, contains a tube network by means of nutrients and signals circulating through the body. The tube network can sense and adopt to local shear stress difference in its own body until the shortest tubes of connecting two food sources (placed at two exits of a maze) keep alive while longer tubes vanish eventually. Unlike to the global optimization algorithm *Physarum solver*, we develop a mathematical model of a dynamical system which capture the local control behavior of *Physarum* when searching for foods. With this new model, a novel adaptive amoeba algorithm for maximum flow problem is proposed, which can be used to solve shortest path problem as *Physarum solver*. Furthermore,we firsts apply this model to solve maximum flow problem, where its convergence is proved in this paper as well. Additionally, numerical results demonstrate the validity and efficiency of the proposed algorithm to solve maximum flow problem.**

***Index Terms*—maximum flow problem, bio-inspired algorithm, network optimization, *physarum solver*.**

I. Introduction

NETWORK flow problems form a large class of optimization problems and are central problems in operations research, computer science, civil engineering and combinatorial optimization [1], [2], [3], [4], [5]. Among them, one of the most famous problems is maximum flow problem, which has many applications in transportation, logistics, telecommunications, and scheduling etc. The maximum flow problem was first formulated by Ted Harris and F. S. Ross when they studied a simplified model of Soviet railway traffic flow in 1954 [6] as follows. Consider a rail network connecting two cities by way of a number of intermediate cities, where each link of the network has a number assigned to it representing its capacity. Assuming a steady state condition, the flow is maximal from one given city to other this moment.

To the best of our knowledge, the existing algorithms for maximum flow problem can be classified in two major classes. One is the classical augmenting path method. Ford and Fulkerson devised the first algorithm of feasible flow by working with augmenting path incrementing the flow at every iteration [7]. This algorithm is based on the fact that a flow is maximum if and only if there is no augmenting path. It repeatedly finds an augmenting path and augments along it until no augmenting path exists anymore. The other is the preflow algorithm. A preflow is a flow that seems violate the restriction that the incoming flow and outgoing flow should be balanced. The push-relabel algorithm is a representative of this type which is efficient both theoretically and empirically [8]. A comprehensive discussion of such algorithms and applications can be found in [9]. Note that max flow problem is a fundamental problem. Numerous efficient algorithms for this problem have been proposed to reduce the runtime or time complexity. However, for general cases, only small improvements have been made. In this paper, we try our best to solve maximum flow problem from other perspective, that is, is there a different type of method for maximum flow problem other than the the two classical algorithms?

Recently, studies of *Physarum polycephalum* (*P. polycephalum*) have attracted huge attention. The plasmodium of *P. polycephalum* is a large amoeba-like organism with great intelligence. Studies have shown that it has the ability to solve a complex maze [10], [11] and other graph theoretical problems [12], [13], [14], [15]. When food sources (FSs) placed at two exits are presented to a starved *P. polycephalum*, it covers FSs to absorb nutrients and constructs a tube network by means of which nutrients and signals circulate through the body. Later, only a few short tubes remain which means effectiveness for transportation. Tero *et al.* develop a mathematical model *Physarum solver* to describe the above adaptation process of the tube network [16]. The insight essence of *Physarum solver* is rhythmic oscillation which is exactly described by positive feedback mechanism between the thickness of each tube and internal protoplasmic flow. *Physarum solver* has been applied in many fields [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27]. In the dissertation [28], Wang is the first to apply physarum solver to solve maximum flow problem. Wang claims that her developed algorithm is easy to implement, and more importantly, it can obtain maximum flow and minimum cut sets simultaneously.

Even though significant progress has been made, it remains challenging to achieve novel methods to solve maximum flow problem. In this paper, we are motivated to develop a biologically inspired algorithm for maximum flow problem based on *Physarum solver*. As we know, it is the first time that *Physarum solver* is developed to solve maximum flow problem. It is very interesting to see how this new amoeba model works when solving a classical network flow problem. Second, we also note that the adaptation principle of *Physarum solver* follows the property of collaborating simultaneously and globally. However, it is observed in experiment that *P. polycephalum* adapts tube diameters in

Manuscript received Feb. 17, 2017; revised June 6, 2017. We thank Dr. Qing Wang at Southwest University and PHD Cai Gao at University at Buffalo for helpful discussions. The work is partially supported by National Natural Science Foundation of China (No. 61364030 and 61763009), the Scientific Research Foundation of Educational Commission of Hubei Province of China (No. D20151902),the Training Programs of Innovation and Entrepreneurship for Undergraduates of Hubei Province (No. 201410517018), The Training Programs of innovation and entrepreneurship for undergraduates of Hubei University for Nationalities (No. 2016CX033).

D. Wei is with the School of Science, Hubei University for Nationalities, Enshi,445000, China e-mail: prof.wei@hotmail.com.

Z. Wang is with Hubei University for Nationalities.

response to wall shear stress [16], [18]. Based on these facts, we simulate the adaptation process of *P. polycephalum* and develop a similar but novel mathematical model where a dynamical system works quite like *Physarum* but responses to local information. Third, the proposed model is proved to efficiently solve maximum flow problem by testing on instances with relatively large size when compared with *Physarum solver* [16].

The rest of the paper is organized as follows. In Section 2, some definitions and preliminary descriptions are introduced. Section 3 describes the proposed mathematical model and applies it to solve maximum flow problem. Section 4 gives the short proof of the adaptive amoeba algorithm's correctness. We use a simple example to show how the algorithm works and several large-size instances to test the efficiency of the proposed algorithm, in Section5. Finally, concluding remarks are given in Section 6.

II. Preliminaries

In this section we introduce some basic knowledge about maximum flow problem and the well-known theorem, "Max-flow Min-cut Theorem".

A. Maximum Flow Problem

The maximum flow problem is defined on a capacitated directed network $G = (N, M, L, C)$ with a set of n nodes and m directed edges. Specifically, M_{ij} denotes the edge with a direction from node i to node j. Its length is written as L_{ij}. C_{ij} denotes the capacity of edge M_{ij}, which is always nonnegative. For edge M_{ij}, the lower bound of the feasible flow is zero and an upper bound on flow is C_{ij}. In such a network, the maximum flow problem is to send maximum possible flow from a source s to a sink t. Let f represents the amount of flow in the network. Then, the maximum flow problem may be expressed as follows,

$$maximize\ f \tag{1}$$

subjecting to

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = \begin{cases} f & if\ i = s, \\ 0 & if\ i \neq s\ or\ t, \\ -f & if\ i = t, \end{cases} \tag{2}$$

where $0 \le x_{ij} \le C_{ij},\ i, j = 1, 2, ..., n$. The sums and inequalities are taken over all edges in the network. Every feasible flow f must satisfy the above capacity constraint and flow conservation constraint.

B. Max-flow Min-cut Theorem

Let two mutually exclusive subsets S, T in N, that is, $S, T \subset N$, $S \cup T = N$ and $S \cap T = \emptyset$. Considering any two nodes $p \in S$ and $q \in T$, the cut $[S, T]$ is defined as

$$[S, T] = \{e_{pq} \in M | p \in S, q \in T\} \tag{3}$$

and the capacity of cut $[S, T]$ is defined as

$$C(S, T) = \sum_{e \in M} C_e. \tag{4}$$

The following is the well-known max-flow min-cut theorem. We introduce this theorem here for later use when proving the correctness of the proposed algorithm.

Theorem 2.1: The maximum value of $|f|$ is equal to the minimum capacity of cut $[S, T]$, that is, $|f| = C(S, T)$.

III. An Adaptive Amoeba Algorithm

Given a resistance network G, a particle is assumed with one unit of energy which enters at the starting node s. This particle traverses through the network and leaves from the sink node t. Each node has an ability of storing particles temporarily. The number of particles stored in a node indicates the energy level of this node's current state. These particles flow to nodes with lower energy levels and leave from the sink node t. Furthermore, we define energy flow E_{ij} transferred from node i to node j as

$$E_{ij} = \frac{\Phi_i - \Phi_j}{L_{ij}/D_{ij}} = \frac{D_{ij}}{L_{ij}}(\Phi_i - \Phi_j), \tag{5}$$

where Φ_i is the energy level of node i, L_{ij} length of edge M_{ij}, and D_{ij} is the conductivity of edge M_{ij}. L_{ij}/D_{ij} can be regarded as the energy resistance. Due to non-negativity of both L_{ij} and D_{ij}, the direction of the energy flow E_{ij} is determined by the sign of $\Phi_i - \Phi_j$. If $\Phi_i > \Phi_j$, the energy flows from node i to node j, otherwise it goes to the opposite direction.

For the source s, particles enter at a rate of I_0 and disappear at the sink node with equal rate. The total particle flow I_0 is a fixed constant in our model. In order to describe the positive feedback behavior of the system, the conductivity D_{ij} varies in response to energy drop with time according to the following *evolution equation*,

$$\frac{d}{dt}D_{ij} = f(E_{ij}) - D_{ij} \tag{6}$$

where $f(E_{ij})$ is the driving power that incurs in energy propagation and satisfies $f(0) = 0$. In this paper, we set $f(E) = E$. Then, the semi-implicit scheme of *evolution equation* can be expressed as

$$\frac{\Phi_i^{t+1} - \Phi_j^t}{\delta t} = E_{ij}^t - D_{ij}^{t+1} \tag{7}$$

where δt is a time mesh size and the upper index t denotes a time step. As energy inflow and outflow incur in each time step, the amount of energy stored in each node is updated as

$$\Phi_i^{t+1} = \Phi_i^t + \sum_{e \in M_i} E_e^t \tag{8}$$

where M_i is the set of adjacent edges of node i.

In a word, the network full of particles can be viewed as a system of energy propagation on the basis of aforementioned defined rules. So far, we describe a mathematical model for a dynamical system in response to local information (node energy difference). Similar to *Physarum solver*, the proposed model can solve shortest path problem [26]. However, our model adopts to energy level difference between each pair of nodes while the flows in *Physarum solver* are computed by solving a systems of linear equation. The time complexity of solving it is $O(n^3)$, but our model computes the flows in $O(m)$ at each iteration.

IV. Solving Maximum Flow Problem

In this section, we apply the above mathematical model to solve maximum flow problem. First, a new network G' is constructed by adding a dummy node v and two virtual edge M_{sv} and M_{vt}. Such additional path $s-v-t$ is longer than any other possible path between s and t by setting $L_{sv} = L_{vt} =$

$n\max(L)/2$, where $max(L)$ is the length of the longest edge among M. The capacities of edges M_{sv} and M_{vt} are $C_{sv} = C_{vt} = \max(\sum_{e\in M_s} C_e)$. The total inflow I_0 is equal to C_{sv} and C_{vt}. On the basis of the positive feedback between the energy flow E_{ij} and the energy conductivity D_{ij}, we want to the flow converges to the edges in G as much as possible by complying with some certain constraint conditions below. For an unsaturated edge, the flow should follow Eq. (5). The maximum flow of a saturated edge should be lower than C_{ij}. As a consequence, the energy flow Eq. (5) is rewritten as

$$E_{ij} = \begin{cases} \frac{D_{ij}}{L_{ij}}(\Phi_i - \Phi_j), & M_{ij} < C_{ij}, \\ C_{ij}, & M_{ij} = C_{ij}. \end{cases} \tag{9}$$

Consequently, the conductivity of edge M_{ij} evolves according to the equation as

$$\frac{dD_{ij}}{dt} = \begin{cases} E_{ij} - D_{ij}, & 0 \le E_{ij} \le C_{ij}, \\ C_{ij} - D_{ij}, & E_{ij} > C_{ij}. \end{cases} \tag{10}$$

In order to ensure flow tracing the virtual edges mostly at the first iteration, the conductivity $D_{ij}(0)$ of each edge in G' initialize as

$$D_{ij}(0) = \begin{cases} \frac{2L_{sv}C_{sv}}{\min\limits_{e\in M}(L_e C_e)}, & i = v \ or \ j = v, \\ 1, & otherwise, \end{cases} \tag{11}$$

where $\min\limits_{e\in M}(L_e C_e)$ is the minimum value of the length of each edges in M times its capacity, respectively. Later, the flow will converge to the path from s to t in the network G as much as possible. When the flow in the G' is steady, the maximum flow is obtained.

V. Proof

In this section, we prove that the flow in the network G is the maximum flow when the dynamical system is stable.

Let τ be the stabilization time. The stabilization of the system means $\forall t > \tau, dD_{ij}^t/dt = 0$.

Lemma 5.1: In an equilibrium point, if $E_{ij} = 0$, then $\Phi_i - \Phi_j < L_{ij}$; if $0 < E_{ij} < C_{ij}$, then $\Phi_i - \Phi_j = L_{ij}$; if $E_{ij} = C_{ij}$, then $\Phi_i - \Phi_j \ge L_{ij}$.

Proof: If $\forall t > \tau, E_{ij}^t = 0$ and $D_{ij}^t \ne 0$, by Eq. (10), D_{ij}^t decreases with time, which is inconsistent with the default assumption. As a consequence, $dD_{ij}^t/dt < 0$. According to Eqs. (9) and (10), we get $dD_{ij}^t/dt = E_{ij}^t - D_{ij}^t = ((\Phi_i^t - \Phi_j^t)/L_{ij} - 1)D_{ij}^t < 0$. Thus, we get $\Phi_i^t - \Phi_j^t < L_{ij}$. If $0 < E_{ij}^t < C_{ij}$, there is $dD_{ij}^t/dt = 0$. Based on Eqs. (10) and (9), it can be seen that $E_{ij}^t = D_{ij}^t$. Then we get $\Phi_i^t - \Phi_j^t = L_{ij}$. By solving the lower differential equation of (10) when $E_{ij}^t = C_{ij}$, we have $\lim\limits_{t\to\infty} D_{ij}^t = \lim\limits_{t\to\infty}(C_{ij} + (D_{ij}^0 - C_{ij})\exp(-t)) = C_{ij}$. By Eq. (9), there is $\Phi_i^t - \Phi_j^t > L_{ij}$. ∎

Lemma 5.2: In an equilibrium point, $\Phi_s - \Phi_t = L_{sv} + L_{vt}$.

Proof: According to Lemma 5.1, if the total flow I_0 is large enough to ensure the flow on the virtual edges satisfy $0 < E_{ij} < C_{ij}$, we will have $\Phi_s - \Phi_t = (\Phi_s - \Phi_v) + (\Phi_v - \Phi_t) = L_{sv} + L_{vt}$. Generally speaking, the maximum flow in the network G is lower than the value of total incoming flow I_0 we set above. ∎

Lemma 5.3: Assume a path from s to t $(s, N_{k_1}, N_{k_2}, ...N_{k_i}, t)$ in the G in an equilibrium point. Then, there exists at least one saturated edge.

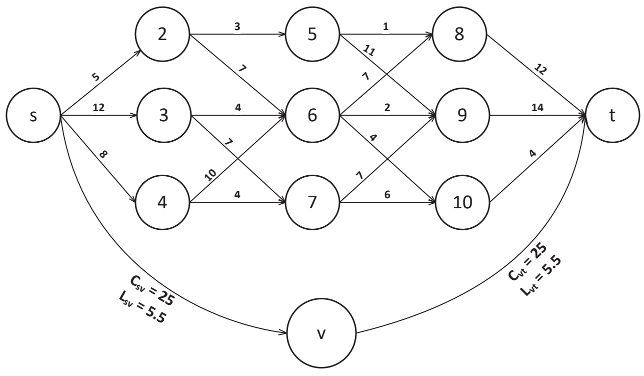

Fig. 1. Given a directed network with 11 nodes and 19 edges, $L_{ij} = 1$, for all i, j. The upper capacity bounds of edges are listed beside, i.e., $C_{12} = 5$. By connecting a dummy node v with both s and t, we have a new network.

Proof: According to Lemma 5.2, we have $\Phi_s - \Phi_t = (\Phi_s - \Phi_{k_1}) + (\Phi_{k_1} - \Phi_{k_2}) + ... + (\Phi_{k_i} - \Phi_t) = L_{sv} + L_{vt}$. Due to $L_{sv} + L_{vt}$ is larger than any path from s to t in G, so $\Phi_s - \Phi_t > L_{sk_1} + L_{k_1k_2} + ... + L_{k_it}$. Therefore, there must be at least one edge that satisfies $\Phi_i - \Phi_j > L_{ij}$. It means that there is a saturated edge M_{ij}, that is, $E_{ij} = C_{ij}$. For edge M_{ji}, in turn, there is $\Phi_j - \Phi_i < -L_{ji}$. By Eq. (10), we get $E_{ji} = 0$. ∎

Let $[S, T]$ is a cut of G from a source node s to a sink node t. Then, we have

$$[S,T] = \{e_{ij} | e_{ij} \in M, i \in S, j \in T\},$$

$$[T,S] = \{e_{ij} | e_{ij} \in M, i \in T, j \in S\}.$$

Theorem 5.4: In an equilibrium point, for $e_{ij} \in [S, T]$, each edge is saturated; for $e_{ij} \in [T, S]$, e_{ij} is empty.

Proof: According to Lemma 5.3, both saturated edges and empty ones can be cut off. So these edges include a cut of G. Furthermore, all saturated edges are from S to T and all the empty edges are in the opposite direction. ∎

According to Theorem 2.1, the flow in the network G is obviously equal to the maximum flow when the dynamical system is at equilibrium state.

VI. Simulation Experiment

A. A simple example

For Fig. 1, node 1 and node 11 are set as the source s and the sink t, respectively. The dummy node v is added to squeeze the flow into the original network. The total input flow is set as $I_0 = C_{1v} = C_{v11} = 25$. and the length of the virtual edge is $L_{sv} = L_{vt} = 5.5$. As illustrated in Fig. 1(b), most part of the flow traces among the virtual path $s - v - t$ at the beginning. Afterwards, the value of flow on virtual path decreases and converges to 2 in the equilibrium point. Flows on edges $M_{s2}, M_{s3} and M_{s4}$ are increasing and converge to some fixed value after a certain amount of time. As a result, the value of maximum flow in Fig. 1(a) is obtained $I_0 - 2 = 23$, which is the correct maximum flow that the network could have.

B. Large-scale Instances

In order to test the efficiency of the proposed algorithm, we adopt large max flow problems instances. The instances were generated by G. Skorobohatyi using the program RMFGEN

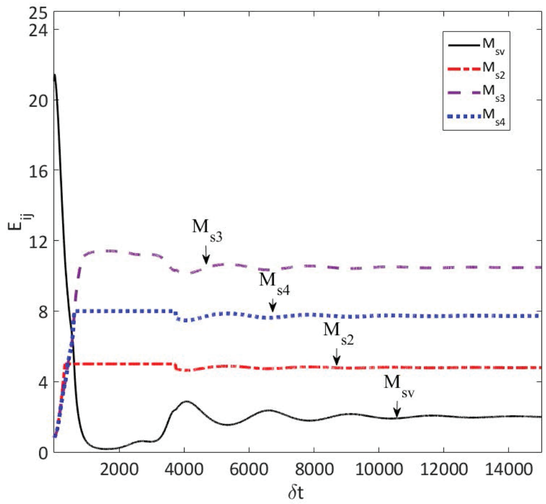

Fig. 2. Flows change of edges from beginning until convergence.

and containing instances of grids-on-pipe graphs [29]. The details about the datasets are shown in the following table I.

TABLE I
THE BASIC TOPOLOGICAL FEATURES OF THE SIX REAL NETWORKS. n AND m ARE THE TOTAL NUMBERS OF NODES AND LINKS, RESPECTIVELY. s AND t DENOTE THE ENTERING NODE AND LEAVING NODE. C IS THE CAP CITY OF EDGES OR ARCS

Networks	n	m	s	t	C
elist96	96	187	1	96	-
elist96d	96	528	1	96	-
elist160	160	285	1	160	-
elist160d	160	912	1	160	-
elist200	200	483	1	200	-
elist200d	200	1340	1	200	-
elist500	500	1040	1	500	-
elist500d	500	3975	1	500	-
elist640	640	3037	1	640	-
elist640d	640	12608	1	640	-

We next present computational results for the iterative solutions method discussed in the above. The computational experiments are performed on a PC with Intel Core I7-4720HQ and 16 GB of memory, running Windows 10. The algorithms are implemented in Matlab programming language of version 2015a. To make a comparison, we also revise the original physarum solver model to solve max flow problems. And furthermore, we also compare our algorithm with the famous linear programming method, “simplex method”. We call it LP in the following parts

In table II, the computational results are listed. Clearly, for instances elist96 and elist96d, the PS algorithm reach to suboptimal solution with a relatively large gap to the optimal solution. However, our proposed algorithm AA can always reach the optimal solution without any gap with even less computational time. Note that, instances with same number of nodes but different number of edges could lead to an significant different computation time. More edges could affect the performance of the PS in a large extent. From the other instances' computational results, we can generally say that more edges lead to more runtime. This is because the particles traverses the networks and need more time to reach the end. FF and LP methods perform much better than PS and AA in the computational speed. We are not surprised by the

TABLE II
COMPARISON BETWEEN PHYSARUM SOLVER(PS), THE PROPOSED ADAPTIVE AMOEBA (AA) ALGORITHM, SIMPLEX METHOD BY LINEAR PROGRAMMING (LP) AND FORD-FULKERSON ALGORITHM (FF). NOTE THAT THE LAST THREE ALGORITHMS' GAP ARE ALL ZEROS. SO WE DO NOT LIST THEM FOR WASTING SPACE.

	PS		AA	LP	FF
Instances	Gap	$Time$	$Time$	$Time$	$Time$
elist96	8.2	49.2	35.3	2.1	0.3
elist96d	8.4	55.5	36.1	1.3	0.2
elist160	9.5	70.0	40.0	4.4	0.6
elist160d	9.3	54.8	31.6	3.5	1.2
elist200	11.5	102.4	75.3	5.2	1.3
elist200d	10.2	120.6	54.2	3.4	1.6
elist500	11.1	200.3	80.9	6.6	2.6
elist500d	9.6	180.7	99.3	7.3	2.7
elist640	12.3	331.6	140.6	8.1	3.2
elist640d	12.8	340.3	157.9	8.3	3.4

longer runtime of PS because solving the flow conservation constraint is truly to solve a system of linear equations. It can be solved by $O(n^3)$ by using Gaussian elimination method.

VII. CONCLUSIONS

In this paper, we have proposed a novel adaptive amoeba algorithm for maximum flow problem. The insight essence of this algorithm is the positive feedback mechanism between the particle flow E and the conductivity D: greater conductivity results in great flow, and this increases conductivity in turn. Distinct from preflow algorithms, the algorithm observes the restriction on the balance of the incoming flow and the outgoing flow into each internal node (other than source and sink nodes). In addition, this model has computational complexity $O(m)$, lower than $O(n^3)$ in *Physarum Solver* [16]. In terms of space complexity, it takes $O(m)$ in our model, also lower than $O(n^2)$ in *Physarum Solver* [16]. We will do research on the time complexity of the iterations in the near future.

As the numerical results illustrate, the proposed algorithm obtains the maximum flow in a continuous manner, which is quite consistent with actual situation. When working with dynamically changing environment, i.e., the cost of transferring goods may change dynamically because of weather or other unexpected factors. The continuous process allow the model to quickly adapt to external variation and recompute the experimental results. In a word, the algorithm is flexible and quite effective.

When comparing with the other two traditional methods for maximum flow problem, the runtime is quite longer. That's because *Physarum solver* is modeled based on the solving a systems of linear equations on every iteration. It takes definitely more time to solve. But as we mention from beginning, the adaptivity of the proposed algorithm has never been shown or developed to solve maximum flow problem. Maybe parallel computing would help accelerate the algorithm.

ACKNOWLEDGMENT

We thank Dr. Qing Wang at Southwest University and PHD Cai Gao at University at Buffalo for helpful conversations. The work is partially supported by National Natural Science Foundation of China (No. 61364030 and 61763009), the Scientific Research Foundation of Educational Commission of Hubei Province of China (No. D20151902),the Training Programs of Innovation and Entrepreneurship for Undergraduates of Hubei Province (No. 201410517018), The Training Programs of innovation and entrepreneurship for undergraduates of Hubei University for Nationalities (No. 2016CX033).

REFERENCES

[1] M. M. A. El-Kareem, "Comparative study between linear and graphical methods in solving optimization problems," *IAENG International Journal of Computer Science*, Vol. 37, No. 1, pp. 110-117, 2010.

[2] H. Hong and J. Kamruzzaman, "A Modified Immune Network Optimization Algorithm," *IAENG International Journal of Computer Science*, Vol. 41, No. 4, pp. 231-236, 2014.

[3] C.-F. Wang, and Y.-H. Zhang, "An improved artificial bee colony algorithm for solving optimization problems," *IAENG International Journal of Computer Science*, Vol. 43, No. 3, pp. 336-343, 2016.

[4] X. Zhang and C. Wu, "Energy cost minimization of a compressor station by modified genetic algorithms," *Engineering Letters*, Vol. 23, No. 4, pp. 258-268, 2015.

[5] A. Song, M. Li, X. Ding, W. Cao and K. Pu, "Community Detection Using Discrete Bat Algorithm," *IAENG International Journal of Computer Science*, Vol. 43, No. 1, pp. 37-43, 2016.

[6] T. E. Harris and F. S. Ross, "Fundamentals of a method for evaluating rail net capacities," *RAND CORP SANTA MONICA CA*, 1955.

[7] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, Vol. 8, No. 3, pp. 399-404, 1956.

[8] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," *Journal of the ACM*, Vol. 35, No. 4, pp. 921-940, 1988.

[9] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, "Network flows: theory, algorithms, and applications," *Prentice Hall, Englewood Cliffs, NJ*, 1993.

[10] T. Nakagaki, H. Yamada, and A. Tóth, "Intelligence: maze-solving by an amoeboid organism," *Nature*, Vol. 407, No. 6803, pp. 470, 2000.

[11] A. Adamatzky, "Slime mold solves maze in one pass, assisted by gradient of chemo-attractants," *IEEE Transactions on Nanobioscience*, Vol. 11, No. 2, pp. 131-134, 2012.

[12] A. Tero, S. Takagi, T. Saigusa, K. Ito, D.P. Bebber, M.D. Fricker, K. Yumiki, R. Kobayashi and T. Nakagaki, "Rules for biologically inspired adaptive network design," *Science*, Vol. 327, No. 5964, pp. 439-442, 2010.

[13] Y.-P. Gunji, T. Shirakawa, T. Niizato, M. Yamachiyo and I. Tani, "An adaptive and robust biological network based on the vacant-particle transportation model," *Journal of Theoretical Biology*, Vol. 272, No. 1, pp. 187-200, 2011.

[14] A. Adamatzky, "If BZ medium did spanning trees these would be the same trees as physarum built," *Physics Letters A*, Vol. 373, No. 10, pp. 952-956, 2009.

[15] J. Jones, "The emergence and dynamical evolution of complex transport networks from simple low-level behaviours," *International Journal of Unconventional Computing*, Vol. 6, No. 2, pp. 125-144, 2010.

[16] A. Tero, R. Kobayashi and T. Nakagaki, "A mathematical model for adaptive transport network in path finding by true slime mould," *Journal of Theoretical Biology*, Vol. 244, No. 4, pp.553-564, 2007.

[17] C. Gao, C. Yan, Z. Zhang,Y. Hu, S. Mahadevan and Y. Deng, "An amoeboid algorithm for solving linear transportation problem," *Physica A: Statistical Mechanics and Its Applications*, Vol. 398, No. 398, pp. 179-186, 2014.

[18] D. Hu and D. Cai, "Adaptation and optimization of biological transport networks," *Physical review letters*, Vol. 111, No. 13, pp. 138701, 2013.

[19] Q. Ma, A. Johansson, A. Tero, T. Nakagaki and D. J. T. Sumpter, "Current-reinforced random walks for constructing transport networks," *Journal of the Royal Society Interface*, Vol. 10, No. 80, pp. 20120864, 2013.

[20] X. Zhang, Z. Zhang, Y. Zhang, D. Wei and Y. Deng, "Route selection for emergency logistics management: A bio-inspired algorithm," *Safety Science*, Vol. 54, No. 2, pp. 87-91, 2013.

[21] X. Zhang, Q. Wang, F. T. S. Chan, S. Mahadevan and Y. Deng, "A physarum polycephalum optimization algorithm for the bi-objective shortest path problem," *International Journal of Unconventional Computing*, Vol. 10, No. 1-2, pp. 143-162, 2014.

[22] X. Zhang, S. Huang, Y. Hu, Y. Zhang, S. Mahadevan and Y. Deng, "Solving 0-1 knapsack problems based on amoeboid organism algorithm," *Applied Mathematics and Computation*, Vol. 219, No. 19, pp. 9959-9970, 2013.

[23] X. Zhang, S. Mahadevan and X. Deng, "Reliability analysis with linguistic data: An evidential network approach," *Reliability Engineering & System Safety*, Vol. 162, pp. 111-121, 2017.

[24] X. Zhang, Y. Zhang, Y. Hu, Y. Deng and S. Mahadevan, "Adaptive amoeba algorithm for constrained shortest paths," *Expert Systems with Applications*, Vol. 40, No. 18, pp. 7607-7616, 2013.

[25] S. Watanabe, A. Tero, A. Takamatsu and T. Nakagaki, "Traffic optimization in railroad networks using an algorithm mimicking an amoeba-like organism, Physarum plasmodium," *BioSystems*, Vol. 105, No. 3, pp. 225-232, 2011.

[26] C. Gao, C. Yan, A. Adamatzky and Y. Deng, "A bio-inspired algorithm for route selection in wireless sensor networks," *IEEE Communications Letters*, Vol. 18, No. 11, pp. 2019-2022, 2014.

[27] X. Cai, "A bio-inspired optimization algorithm for the maximum flow problem," *ICIC Express Letters*, Vol. 11, pp. 3031-3038, 2015.

[28] Q. Wang, "Research on the key issues of graph theory based on physarum polycephalum," *Chongqing, Southwest University*, 2015.

[29] D. Goldfarb and M. D. Grigoriadis, "A computational comparison of the Dinic and network simplex methods for maximum flow," *Annals of Operations Research*, Vol. 13, No. 1, pp. 81-123, 1988.

Dr. Daijun Wei (M'17) is working as professor, School of Science, Hubei University for Nationalities. He received the M.S. and Ph.D. degrees in 2008 and 2014, respectively, from Central China Normal University and Southwest University in China. He has published more than 30 papers in international journals and conferences. His research interests include fuzzy system, civil and engineering, optimizing control system and complex networks.