Petri Nets Based Max-flow/Min-cut Modeling and Analyzing

Zheng Zou[1,2,3], Shi-Jian Liu[4], Jeng-Shyang Pan[5,6]

[1] College of Mathematics and Informatics, Fujian Normal University, China
[2] Fujian Provincial Engineering Technology Research Center for Public Service Big Data Mining and Application, Fujian Normal University, China
[3] Digital Fujian Institute of Big Data Security Technology, Fujian Normal University, China
[4] Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, China
[5] College of Computer Science and Engineering, Shandong University of Science and Technology, China
[6] Department of Information Management, Chaoyang University of Technology, Taiwan
zouzheng84@sina.com, {liusj2003, jengshyangpan}@gmail.com

Abstract

Max-flow/min-cut is named by the dual problem of finding a flow with maximum value in a given network and looking for a cut with minimum capacity overall cuts of the network. Petri Nets (PNs) is an effective modeling tool which has been widely used for the description of distributed systems in terms of both intuitive graphical representations and primitives well-defined by mathematics. Most cited references related to the max-flow/min-cut focus on either the improvement of the solving algorithm or its applications. Whereas in this paper, PNs are adopted to model and analyze the max-flow/min-cut. Firstly, algorithms for generating models from a given flow network and its residual network are introduced based on the PNs theory. Then, the models are combined to simulate the classic Ford-Fulkerson for solving the max-flow/min-cut and finding flow distributions when the max-flow achieves. Finally, simulation results and proofs are presented to show that our method is an intuitive and effective way of understanding and presenting the max-flow/min-cut theory and to ensure its validity. The proposed PNs based method is extensible because the idea can easily be applied to other graph problems similar to the Max-flow/min-cut.

Keywords: Petri nets, Max-flow/min-cut, Ford-Fulkerson method, Modeling, Simulation

1 Introduction

The dual problem of maximum flow and minimum cut in graph theory is often referred to as the max-flow/min-cut, which is one of the most explored problems in terms of combinatorial optimization. Researches related to max-flow/min-cut nowadays focus on either the fundamental principles or its applications. Regarding the fundamentals, Han et al. [1] explored the maximum flow problem regarding to uncertain network. Given a network and a set of source and destination pairs, Bonami et al. [2] discussed the problem of maximizing the sum of the flow under proportional delay constraints. Zhu and Gleich [3] presented a parallel algorithm for the undirected min-cut problem with floating-point valued edge weights. A strong version of the max-flow/min-cut theorem was studied by Aharoni et al. [4] for countable networks. Regarding to the applications, as a powerful tool, max-flow/min-cut also has been widely adopted in scenarios such as road networks evaluation [5-6], online semi-supervised learning [7], invulnerability analysis of power grids [8], and energy minimization in vision [9]. For example, the idea of employing min-cut/max-flow to minimize certain energy functions was firstly introduced by Greig et al. [10] for the purpose of image restoration. Later, Boykov et al. [11] extended the model and proposed a famous framework named Graph Cuts, which has been popularly used for image & mesh segmentation [12-16], stereo with occlusions [17], multi-camera scene reconstruction [18] and so on.

The max-flow/min-cut has played fundamental and crucial roles in many kinds of research which can be formalized by the graph, however, their principles and solutions are usually very abstract and difficult for understanding and presentation, which becomes the major motivation of our work, and we find a Petri Nets (PNs) based approach as the solution. As mentioned by Serral et al. [19], Petri nets are a mathematical and graphical modeling language with powerful analysis techniques. In other words, PNs have animated graphical notations which make them intuitive for describing abstract knowledge and modeling procedures such as choice, iteration and concurrent of a system. Besides, simulations of PNs can be defined mathematically, which makes the PNs based analysis more convincible. For example, Ivasic-Kos et al. [20]

*Corresponding Author: Jeng-Shyang Pan; E-mail: jengshyangpan@gmail.com
DOI: 10.3966/160792642020072104002

represented knowledge about concepts that can appear in an image by a fuzzy PNs-based formalism. Thanks to the fuzzy-knowledge representation scheme, the obtained image interpretation is enriched with new, more general and abstract concepts that are close to concepts people use to interpret these images. Yuan et al. [21] proposed a Petri net model for autonomous tracking of chemical plumes developed in both diffusive and turbulent airflow environments with relatively high source localization rates under the two cases. Formal aspects of the methodology to construct an expert system or decision support model based on the associative Petri net was introduced by Chiang et al. [22] for associative Petri nets are a powerful method to represent knowledge in the domain of decision support systems and can be translated into rule-based systems easily. The PNs also can be used to verify the cross-organisational workflows [23].

The aim of this paper is to use PNs to formalize the max-flow/min-cut problem and depict its solutions. To achieve the object, after the descriptions of some basic concepts and notations in Section 2, modeling algorithms for both flow networks and the corresponding residual networks are introduced in Section 3 according to the PNs theory. Then PNs models of the flow network and residual network are combined to simulate the Ford-Fulkerson method for solving the max-flow/min-cut and finding different flow distributions when the max-flow achieves (see Section 4). In additional, simulation results and proofs are presented in Section 4 and Section 5 respectively to show that our method is an intuitive and effective way of understanding and presenting the max-flow/min-cut theory and to ensure the validity. Section 6 concludes the paper.

Main contributions of this work are as follows:

- PN models for flow networks and residual networks are introduced in this paper, which is intuitive.
- Based on the proposed models, the proposed simulation methods can be used to solve and illustrate the max-flow/min-cut of a given flow network.
- A back-flow strategy also is presented to identify different flow distributions when the max-flow achieves.
- Formal analyses and proofs are made to validate the proposed method.

2 Problem Statement and Preliminaries

In this Section, problem statement and preliminaries about max-flow/min-cut and Petri Nets are given.

2.1 Max-flow/ Min-cut

Max-flow problem concerns about the flow in a network. Definition 1 formally defines the problem (please refer to [24] for the definitions of flow-networks, the flow and its value).

Definition 1 (Maximum flow) [24].

Given a flow network $G=(V,E)$ with source s and sink t, the maximum flow problem wishes to find a flow of maximum value.

Sometimes, acquiring the value of maximum flow in G is the aim of an application, but more commonly, finding out the value of flow for each edge is preferred when a maximum flow achieves [25]. As for a solution, our method is closely related to the classic Ford-Fulkerson method which relays on residual networks. Therefore, the definition of residual networks is given in Definition 2 as well.

Definition 2 (Residual networks) [24].

Given a flow network $G=(V,E)$ and a flow f, the residual network of G induced by f is $G_f=(V,E_f)$, where $E_f=\{(u,v)\in V\times V : c_f(u,v)>0\}$, and $c_f(u,v)$ is the residual capacity which meets the Equation 1.

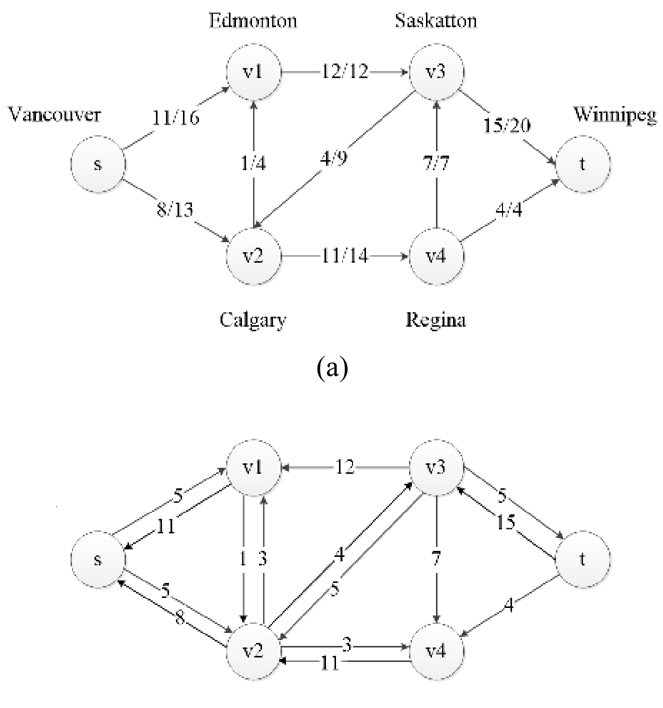
$$c_f(u,v)=\begin{cases} c(u,v)-f(u,v), & if\,(u,v)\in E, \\ f(v,u), & if\,(v,u)\in E, \\ 0, & otherwise. \end{cases} \tag{1}$$

An example of a flow network is shown in Figure 1(a). It represents a case presented in [24] that a company named Lucky Puck has a factory (i.e., source s) in Vancouver that manufactures pucks, and it has a warehouse (i.e., sink t) in Winnipeg that stocks them. The pucks are shipped from s to t through intermediate cities. Only $c(u,v)$ crates per day can go from city u to city v at most, while an actual amount is $f(u,v)$. In Figure 1(a), each edge (u,v) in G is labeled by $f(u,v)/c(u,v)$, which are the flow and capacity respectively, and the value of flow $|f|$ is 19. The aim of the Lucky Puck Company's trucking problem is to determine the largest number of crates per day that they can ship and should produce, namely the max-flow problem for the given G shown in Figure 1(a), whose residual network is illustrated in Figure 1(b).

2.2 Petri Nets

Petri nets, firstly introduced by Dr Carl Adam Petri in 1939, is one of the most famous mathematical modeling tools for the description of distributed systems. It was also known as a Place/Transition net, which consists of basic elements such as Places (e.g, conditions, represented by circles in Figure 2) and Transitions (e.g, events, represented by bars in Figure 2). Tokens (e.g., resources, represented by spots in Figure 2) can be transmitted from one Place to another through Transition between them following some rules.

We use $|P|$ to denote the number of tokens within a Place P. A formal definition of PNs is presented in Definition 3.



(a)

(b)

Figure 1. Demonstration of a flow network (a) and its corresponding residual network (b)

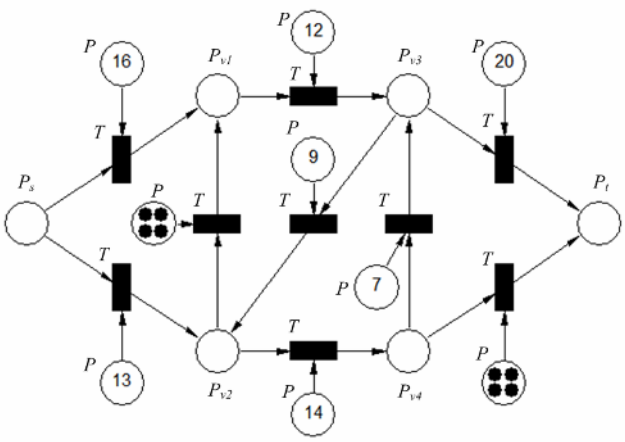


Figure 2. A Petri net system designed according to the graph shown in Figure 1(a)

Definition 3 (Petri Nets) [26].

A Petri net is a triple $N=(S,T;F)$, *satisfying*

(1) $S \bigcup T \neq \varnothing$

(2) $S \bigcap T = \varnothing$

(3) $F \subseteq (S \times T) \bigcup (T \times S)$

(4) $dom(F) \bigcup cod(F) = S \bigcup T$

where dom *means domain of* F *and meets* $dom(F)=\{x \mid \exists y:(x,y)\in F\}$, cod *means codomain of* F *and meets* $cod(F)=\{y \mid \exists x:(x,y)\in F\}$. S *and* T *denote the set of Place and Transition respectively, and* F *denotes the flow relation between them.*

2.3 Ford-Fulkerson Method

As a classic solution of max-flow/min-cut, the Ford-Fulkerson method is proposed by L. R. Ford and D. R. Fulkerson in 1962. As shown in Algorithm 1, it acquires the maximum flow in a flow network by finding an augmenting path in the corresponding residual network and augment the flow iteratively.

Algorithm 1. Ford-Fulkerson method [24]

Input: Flow network G with source s and sink t

Output: A maximum flow f

initialize flow f to 0;
while there exists an augmenting path p in the residual network G_f **do**
    augment flow f along p;
end
return f;

Theorem 1 (Max-flow/min-cut theorem) [24] If f is a flow in a flow network $G=(V,E)$ with source s and sink t, then the following conditions are equivalent:

(1) f is a maximum flow in G.

(2) The residual network G_f contains no augmenting paths.

(3) $|f| = c(C_S, C_T)$, where $C_S = \{v \in V :$ there exists a path from s to v in $G_f\}$, $C_T = V - C_S$, and $c(C_S, C_T)$ is the minimum capacity of cut (C_S, C_T) overall cuts of the network G (i.e., a minimum cut).

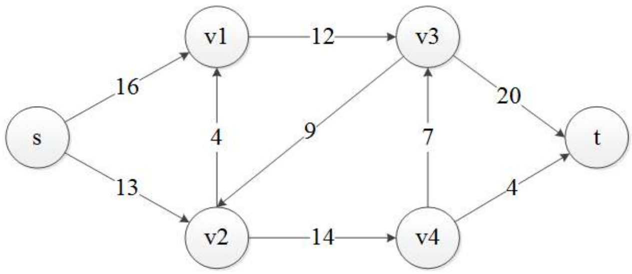
3 The Proposed Modeling Method

In this Section, PNs based modeling methods for (1) flow network and (2) its corresponding residual network are proposed.

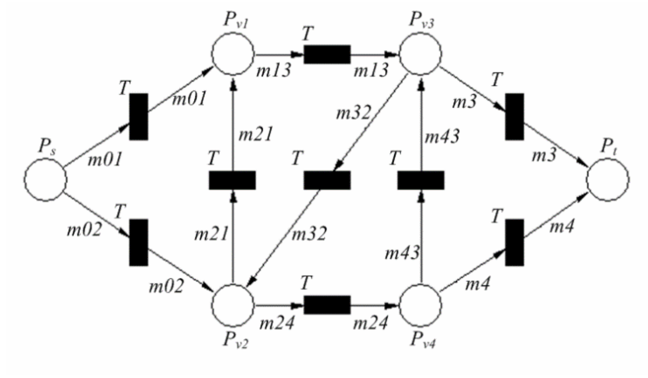
3.1 Flow Network Modeling

A primary issue when we use PNs for modeling is to determine the meaning of each Place and Transition in PNs. As for the modeling of flow network $G=(V,E)$ shown in Figure 3(a), the Places and Transitions stand for vertexes V and edges E respectively in our method. The formal description of the flow network modeling method is introduced in Algorithm 2.

Given a flow network $G=(V,E)$ shown in Figure 3(a) as input, Figure 3(b) shows the result of Algorithm 2. The parameter m_{uv} in N_G is a variable which stands for flow $f(u,v)$ between $u,v \in V$ which need to be set.



(a)



(b)

Figure 3. A flow network (a) and its corresponding PNs model according to Algorithm 2 (b)

Algorithm 2. PNs based flow network modeling

Input: A flow network $G=(V,E)$

Output: Petri net model N_G of G

1. **foreach** $v \in V$ **do**
2. add a Place P_v into N_G corresponding to v in G ;
3. **end**
4. **foreach** $(u,v) \in E$, where $u,v \in V$ **do**
5. add a Transition T into N_G between P_u and P_v ;
6. add two directed arcs into N_G pointing both from P_u to T and from T to P_v with parameter m_{uv} ;
7. **end**
8. **return** N_G ;

3.2 Residual Network Modeling

Let G_f be the residual network of a given flow network $G=(V,E)$, since G_f contains not only the edges in G but also some reversed edges, we modify the PNs model shown in Figure 2 and propose a residual network modeling method in Algorithm 3.

Algorithm 3. PNs based residual network modeling

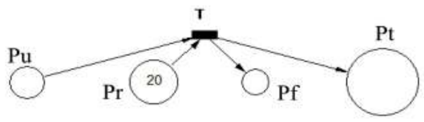
Input: Flow network $G=(V,E)$ with source s and sink t

Output: Petri net model N_{Gf} of residual network G_f corresponding to G

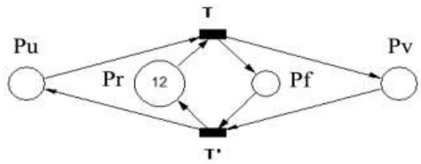
1. **foreach** $v \in V$ **do**
2. add a Place P_v into N_{Gf} corresponding to v in G ;
3. **end**
4. **foreach** $(u,v) \in E$, where $u,v \in V$ **do**
5. add a Transition T into N_{Gf} between P_u and P_v ;
6. **end**
7. **foreach** $(u,v) \in E$ and its associated Transition T **do**
8. add a foreword Place P_f into N_{Gf} , s.t. $P_f \in T^{\bullet}$ and $|P_f| = f(u,v)$, where $T^{\bullet}$ is the post-set of T ;
9. add a revise Place P_r into N_{Gf} , s.t. $P_r \in {}^{\bullet}T$ and $|P_r| = c(u,v) - f(u,v)$, where ${}^{\bullet}T$ is the pre-set of T ;
10. **if** $v \neq t$ **then**
11. add a Transition T' into N_{Gf} , s.t. ${}^{\bullet}T' = \{P_v, P_f\}$ and $T'^{\bullet} = \{P_u, P_r\}$;
12. **end**
13. **end**
14. **return** N_{Gf} ;

Figure 4(a) and Figure 4(b) demonstrates two examples for Algorithm 3 running from step 7 to 13. Namely, for $(u,v) \in E$, if $f(u,v)=0$ and $c(u,v)=20$, then P_f and P_r will be added into N_{Gf} with the number of tokens equals 0 and 20 respectively (see Figure 4(a)). If $v \neq t$ where t denotes the sink, then T' will further be added as shown in Figure 4(b). The reason for treating t differently is that adding a reverse edge for the sink in the residual network is meaningless, resources should not leave the sink after they have arrived their destination. Figure 4(c) depicts the result of Algorithm 3 given flow network shown in Figure 3(a) as input.

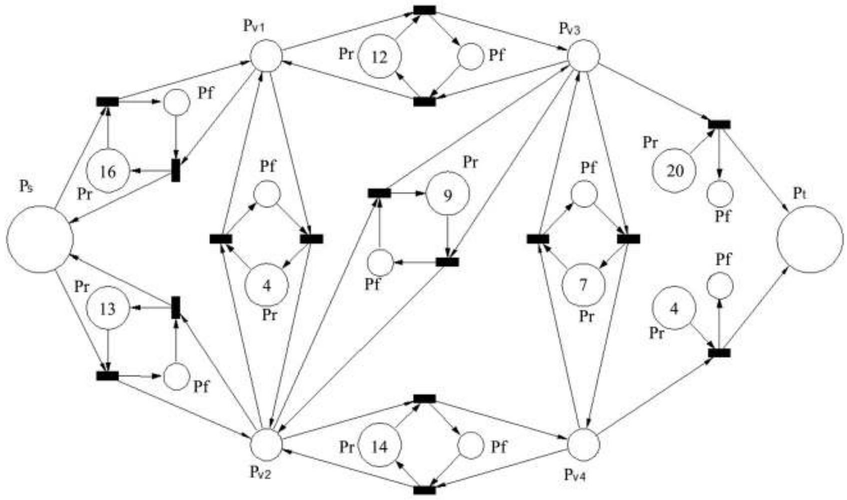
It is worth to point out that the P_f and P_r described in Algorithm 3 are referred to as forward Place and reverse Place in this paper, because according to PNs rules, if (u,v) is a forward edge of flow network G , then the number of tokens out of P_u and into P_f will be exactly the same. That is to say, $|P_f|$ will always equal to $f(u,v)$. Similarly, the number of tokens out of P_v



(a) An instance for Algorithm 3 running from step 7 to 13 for $(u,v)\in E\wedge v=t$, where forward Place P_f and revise Place P_r with a specified number of tokens are added into the model



(b) An instance for Algorithm 3 running from step 7 to 13 for $(u,v)\in E\wedge v\neq t$, where forward Place P_f, revise Place P_r and Transition T' are added into the model



(c) The result of Algorithm 3 given flow network shown in Figure 3(a) as input

Figure 4. Demonstration of PNs based residual network modeling method given a flow network $G=(V,E)$

and into P_r will be exactly the same. So, $|P_r|$ will always be $c(u,v)-f(u,v)$, namely, the revise flow $f(v,u)$ if $v\neq t$. Additionally, P_f and P_r are complementary Places (or S-complement), which are important elements in PNs theory [26], they make sure that no contact will be added into the system. Further discussions about the S-complement can be found in our previous work [27]. The useful properties of forward and reverse Places are summarized in Property 1.

Property 1. *Given a flow network* $G=(V,E)$, *for* $\forall(u,v)\in E$, *the value of flow* $f(u,v)$ *equals to the number of tokens in its forward Place* P_f, *while the value of revise flow* $f(v,u)$ *in the corresponding residual network* G_f *equals to the number of tokens in its reverse Place* P_r.

4 Simulation and Analysis

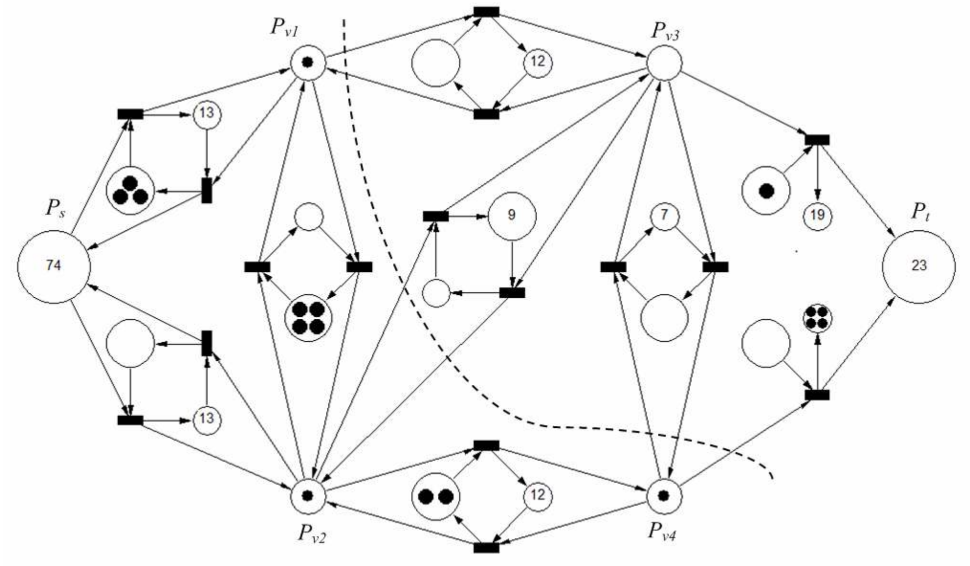
A major fact distinguished our method from others is that though not designed to find a max-flow/min-cut solution, our method can illustrate a solution intuitively by the simulation of PNs based modeling systems proposed in Section 3, and provide some analyses.

4.1 Max-flow/Min-cut Solution

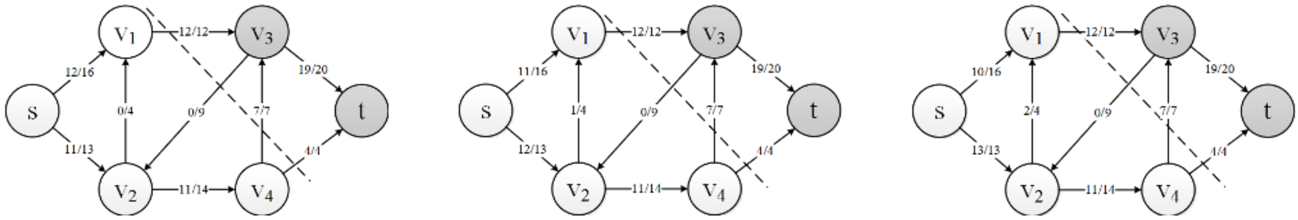
When solving the max-flow/min-cut, the core idea of our method is that, take the N_{Gf} shown in Figure 4(c) for example, if we initialize the source Place P_s with enough tokens (e.g., 100 tokens in our experiments) and let the system run under the rules of PNs automatically until it reaches a status that there exists no path for a token flow from source Place P_s to sink Place P_t as shown in Figure 5(a), then a max-flow achieves. At the same time, a min-cut can be found as two sets of vertexes divided by dashed lines demonstrated in Figure 5. Namely, the line separates V of flow network $G(V,E)$ into two categories, which are denoted as C_S and C_T in this paper. C_S contains vertexes corresponding to Places in the biggest live sub-net of N_{Gf} (e.g., P_s, P_{v1}, P_{v2} and P_{v4} in Figure 5(a)), C_T contains the rests. The concepts of liveness about PNs can be found in [26].

4.2 Max-flow Distribution

Our method can not only find the max-flow but also acquire a flow value of each edge when maximum flow achieves. Specifically, we find the flow values by a



(a) A max-flow has achieved when there exists no path for a token flow from P_s to P_t



(b) are three different max-flow distributions when different back-flow paths of N_{Gf} shown in (a) are chosen

(c) are three different max-flow distributions when different back-flow paths of N_{Gf} shown in (a) are chosen

(d) are three different max-flow distributions when different back-flow paths of N_{Gf} shown in (a) are chosen

Figure 5. Simulation results of a PNs based residual network N_{Gf}

back-flow strategy. In other words, we let the tokens flow back to P_s, after that, the number of tokens reserved in the forward Places P_f shall be the flow value $f(u,v)$ of edge (u,v). Furthermore, it is worth to point out that different back-flow paths may lead to different flow distributions of the max-flow. For example, there are three options for tokens in P_{v1}, P_{v2} and P_{v4} flow back to P_s, which are (1) one from P_{v4} to P_{v2}, two from P_{v2} to s, and one from P_{v1} to P_s; (2) one from P_{v4} to P_{v2}, one from P_{v2} to P_s, one from P_{v2} to P_{v1}, and two from P_{v1} to P_s; (3) one from P_{v4} to P_{v2}, two from P_{v2} to P_{v1}, and three from P_{v1} to P_s. Three of the cases correspond to three flow distributions shown in Figure 5(b), Figure 5(c) and Figure 5(d) respectively.

4.3 System Integration

Using the back-flow strategy, an integrated system N_I for max-flow/min-cut is proposed as demonstrated in Figure 6. The top part of the N_I contains a flow network model as that depicted in Figure 4(c), while the bottom part contains a residual network model as that depicted in Figure 3(b). Differently, Places and Transitions such as T_1, P_1 and P_t are added.

As we can see, there are two phases for our system to run if the source Place s_f is initialized with enough resources. In the first phase, tokens will flow in the top part, which represents the residual network N_{Gf}. Once a max-flow achieves, the flow value of the flow network equals to $|P_t|$. During the first phase, T_1 is used to realize a back-flow process. The parameter ms works as a timer because it makes sure only when $|P_1|$ equals to the number of resources initialized in s_f (e.g., $ms = 100$ in this case) can the system enters into the second phase.

In the second phase, tokens will flow in the bottom part, which represents the flow network N_G. Parameters of arcs such as $m01, m02, \ldots$ in the model are associated with forward Places in N_{Gf}, which indicate the value of the flow of each edge in the flow network. Finally, T_2 is used to trigger a reset of the flow, just as the shipments in another day for the Lucky Puck Company's trucking problem described in [24].

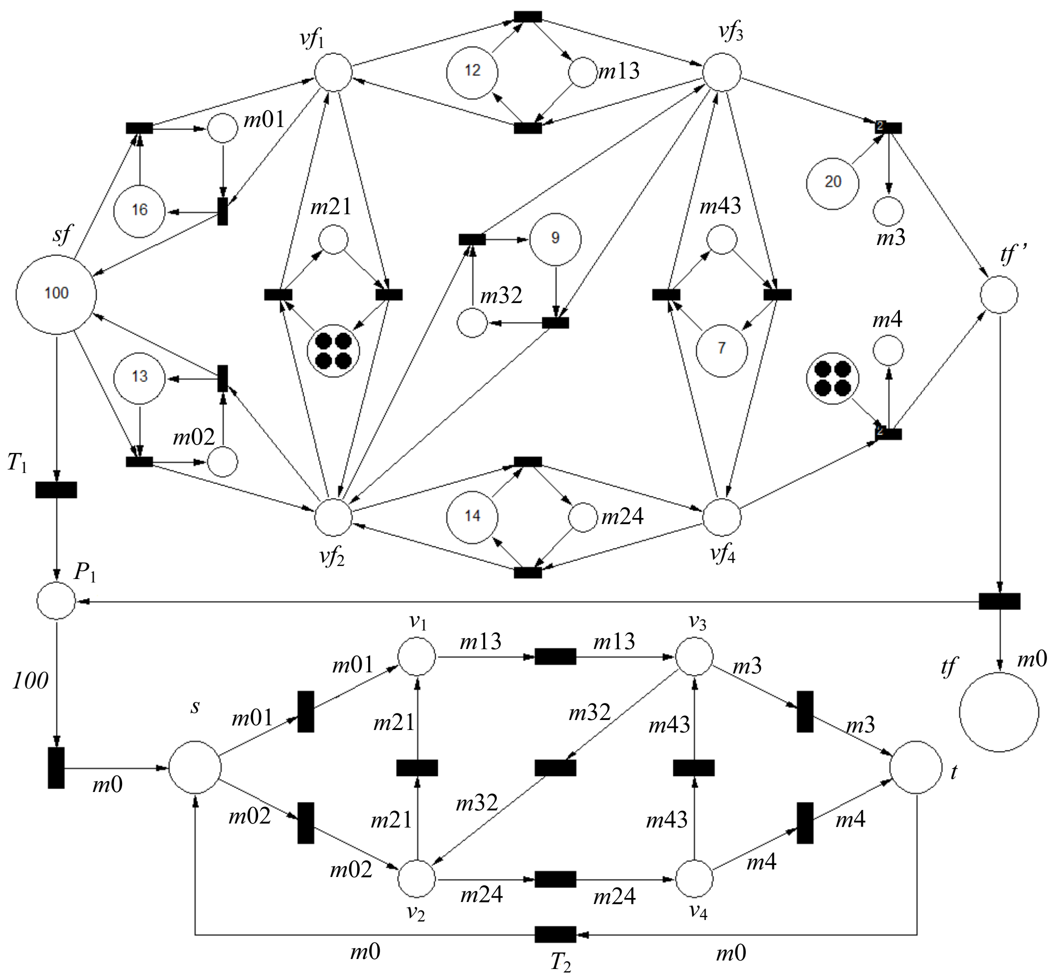


Figure 6. An integrated PNs system for solving and demonstrating the max-flow/min-cut

5 Validation

In the integrated model shown in Figure 6, we find the max-flow/min-cut by the proposed residual network N_{Gf} part, while the flow network N_G part is mainly used for demonstration. Therefore, we focus on the validity of N_{Gf}.

We validate the proposed model by proving it is equivalent to the Ford-Fulkerson method (i.e., Algorithm 1), whose validity is acknowledged. The description and proof of some lemmas and theories are given below.

Lemma 1. *In N_{Gf}, if a token transmits from source Place P_s to sink Place P_t, the flowing path p is an augmenting path in G_f.*

Proof. For a given path p in N_{Gf}, we can always find a simple path p' in G_f which is equivalent to p as follows.

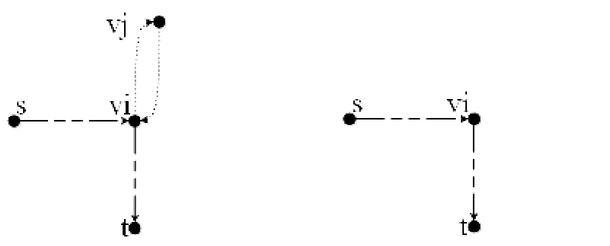
If p is a simple path, then let $p' = p$;

If there is a loop in p, let

$$p = s \rightarrow v_i \rightarrow v_j \rightarrow v_i \rightarrow t \ \ (v_i \neq t)$$

As Figure 7(a) illustrated, we can delete the loop and form the simple path $p' = s \rightarrow v_i \rightarrow t$ as shown in Figure 7(b) because the number of tokens out of v_i and into it are the same in N_{Gf}.

If there exist multiple loops in p, a simple path p' will be acquired after deleting all loops as described above.



(a) A path with a loop

(b) A simple path after removing the loop shown in (a)

Figure 7. Demonstration of loop removal for path in a PN based residual network model

Lemma 2. *In* N_{Gf} *, a unit of token reaches* P_t *from* P_s *is equivalent to a unit of flow is pushed along an augmenting path in* G_f .

Since an augmenting path is a simple path in residual network from s to t, then it can be the path p' which is equivalent to p.

Proof. When a unit of token reaches P_t in N_{Gf} going through p, we can transform p into a simple path p' as did in the proof of Lemma 1. For each edge e in p', if e is an edge in G_f, then the number of tokens in its forward Place will increase one; if e is a revise edge in G_f, then the number of tokens in its revise Place will increase one. According to Property 1, it means a unit of flow increased for each e, namely along p', which is equivalent to an augmenting path in G_f according to Lemma 1.

Theorem 2. *When no path can be found for a token transmitted from* P_s *to* P_t *in the proposed PNs based residual network* N_{Gf} *, a max-flow* f *achieves and* $|f|$ *equals to the number of tokens in* P_t .

Proof. According to Theorem 1, after the flow f is initialized to 0, if there exists an augmenting path p in G_f, augment flow along p. While, in our method, we can find a path p' in N_{Gf} which is equivalent to p according to Lemma 1, and transmit a unit of token along p', which is equivalent to push a unit of flow along p according to Lemma 2. Therefore, our method is equivalent to the Ford-Fulkerson method. The difference between the two methods is that we augment a unit of tokens once a time concurrently in PNs, other than a specified flow in the Ford-Fulkerson method.

According to Algorithm 1, a max-flow achieves when there exists no augmenting path in the residual network. While the equivalent case in our method is that no path can be found for a token transmitted from P_s to P_t.

Furthermore, according to Equation 1, $|f|$ equals to the total flow into the sink t, which is the $|P_t|$.

Theorem 3. When a max-flow in flow network $G=(V,E)$ achieves, the set C_S consists of vertexes corresponding to Places in the biggest live sub-net of N_{Gf} and the set $C_T = V - C_S$ form a min-cut (C_S, C_T) of G.

Proof. According to Theorem 1, when a max-flow in flow network $G=(V,E)$ achieves, let $C_S = \{v \in V$: there exists a path from s to t in $G_f\}$ and $C_T = V - C_S$, the cut (C_S, C_T) is a min-cut. Similarly, in our PNs based residual network N_{Gf}, when no path can be found for a token transmitted from source Place P_s to sink Place P_t, a max-flow achieves according to Theorem 2. And we can also treat vertexes which correspond to Places in the biggest live sub-net of N_{Gf} as the set C_S, and the others as C_T which form a min-cut because living system means every Transition in it has a chance to be enabled and transmit tokens from its pre-set Places to its post-set Places. In other words, there exists a path for tokens to transmit from P_s to each Place whose corresponding vertex belongs to C_S.

6 Conclusion and Future Work

In this paper, a PNs based approach is proposed for modeling and analyzing the max-flow/min-cut, which is a basic graph theory used in many kinds of research. Most cited references related to the max-flow/min-cut focus on either the improvements of algorithm which solves the problem or some applications, whereas, in this paper, we aim to formalize the max-flow/min-cut problem and depict its solution. The advantages of using PNs for modeling and analyzing are that the PNs have animated graphical notations which make them intuitive for describing abstract knowledge and modeling procedures such as choice, iteration and concurrent of a system. In additional, simulations of PNs can be defined mathematically, which makes the PNs based analysis convincible.

To fulfil the aims, firstly, novel modeling methods for both flow network and the corresponding residual network are introduced according to the PNs theory. Then the PNs models are combined to simulate a classic max-flow/min-cut solution, and we can come to the conclusions that, in the PNs based modeling system, (1) a max-flow achieves when there exists no path for a token transmit from source Place to sink Place, (2) a min-cut can be determined by liveness analysis of the system, and (3) different flow distributions depend on the choice of different back-flow paths. Proofs of the simulation results also are given to validate the feasibility of the proposed method.

Focusing on the modeling and analyzing of max-flow/min-cut and its solution, the proposed method is an intuitive and effective way of understanding and presenting the max-flow/min-cut theory. In the future, we would like to extend this work for other problems in graph theory. In addition, it would be interested to apply PNs for state-of-the-art intelligent algorithms such as those proposed in [28-29].

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61872085), Project of Fujian Education Department Funds (No. JK2017029), Scientific Research Project of Fujian Education Department “The microstructure extraction of the root ranal on CBCT" (No. JAT190069) and Scientific Research Project of Fujian University of Technology (No. XF-X19017).

The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] S. Han, Z. Peng, S. Wang, The Maximum Flow Problem of Uncertain Network, *Information Sciences*, Vol. 265, pp. 167-175, May, 2014.

[2] P. Bonami, D. Mazauric, Y. Vaxès, Maximum Flow under Proportional Delay Constraint, *Theoretical Computer Science*, Vol. 689, pp. 58-66, August, 2017.

[3] Y. Zhu, D. F. Gleich, A Parallel Min-Cut Algorithm Using Iteratively Reweighted Least Squares Targeting at Problems with Floating-Point Edge Weights, *Parallel Computing*, Vol. 59, pp. 43-59, November, 2016.

[4] R. Aharoni, E. Berger, A. Georgakopoulos, A. Perlstein, P. Sprüssel, The Max-Flow Min-Cut Theorem for Countable Networks, *Journal of Combinatorial Theory*, Series B, Vol. 101, No. 1, pp. 1-17, January, 2011.

[5] K. Otsuki, Y. Kobayashi, K. Murota, Improved Max-Flow Min-Cut Algorithms in A Circular Disk Failure Model with Application to A Road Network, *European Journal of Operational Research*, Vol. 248, No. 2, pp. 396-403, January, 2016.

[6] V. Ramesh, S. Nagarajan, S. Mukherjee, Max-Flow Min-Cut Algorithm in Spark with Application to Road Networks, *International Conference on Big Data Technologies and Applications*, Seoul, Korea, 2016, pp. 12-22.

[7] L. Zhu, S. Pang, A. Sarrafzadeh, T. Ban, D. Inoue, Incremental and Decremental Max-Flow for Online Semi-Supervised Learning, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 8, pp. 2115-2127, August, 2016.

[8] W. Fan, S. Huang, S. Mei, Invulnerability of Power Grids based on Maximum Flow Theory, *Physica A: Statistical Mechanics and Its Applications*, Vol. 462, pp. 977-985, November, 2016.

[9] Y. Boykov, V. Kolmogorov, An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision, *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, Sophia Antipolis, France, 2001, pp. 359-374.

[10] D. M. Greig, B. T. Porteous, A. H. Seheult, Exact Maximum A Posteriori Estimation for Binary Images, *Journal of the Royal Statistical Society*, Vol. 51, No. 2, pp. 271-279, January, 1989.

[11] Y. Boykov, O. Veksler, R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, *Seventh IEEE International Conference on Computer Vision, Kerkyra*, Greece, 1999, pp. 377-384.

[12] Y. Y. Boykov, M.-P. Jolly, Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images, *Eighth IEEE International Conference on Computer Vision*, Vancouver, Canada, 2001, pp. 105-112.

[13] Y. Boykov, G. Funka-Lea, Graph Cuts and Efficient N-D Image Segmentation, *International Journal of Computer Vision*, Vol. 70, No. 2, pp. 109-131, November, 2006.

[14] W. Ju, D. Xiang, B. Zhang, L. Wang, I. Kopriva, X. Chen, Random Walk and Graph Cut for Co-Segmentation of Lung Tumor on PET-CT Images, *IEEE Transactions on Image Processing*, Vol. 24, No. 12, pp. 5854-5867, December, 2015.

[15] A. Sbei, K. ElBedoui, W. Barhoumi, P. Maksud, C. Maktouf, Hybrid PET/MRI Co-Segmentation Based on Joint Fuzzy Connectedness and Graph Cut, *Computer Methods and Programs in Biomedicine*, Vol. 149, pp. 29-41, October, 2017.

[16] L. Fan, L. Liu, K. Liu, Paint Mesh Cutting, Computer Graphics Forum, Vol. 30, No. 2, pp. 603-612, April, 2011.

[17] V. Kolmogorov, R. Zabih, Computing Visual Correspondence with Occlusions Using Graph Cuts, *Eighth IEEE International Conference on Computer Vision*, Vancouver, Canada, 2001, pp. 508-515.

[18] V. Kolmogorov, R. Zabih, Multi-Camera Scene Reconstruction via Graph Cuts, *European Conference on Computer Vision*, Copenhagen, Denmark, 2002, pp. 82-96.

[19] E. Serral, J. De Smedt, M. Snoeck, J. Vanthienen, Context-Adaptive Petri Nets: Supporting Adaptation for The Execution Context, *Expert Systems with Applications*, Vol. 42, No. 23, pp. 9307-9317, December, 2015.

[20] M. Ivasic-Kos, I. Ipsic, S. Ribaric, A Knowledge-Based Multi-Layered Image Annotation System, *Expert Systems with Applications*, Vol. 42, No. 24, pp. 9539-9553, December, 2015.

[21] J. Yuan, D. Oswald, W. Li, Autonomous Tracking of Chemical Plumes Developed in both Diffusive and Turbulent Airflow Environments Using Petri Nets, *Expert Systems with Applications*, Vol. 42, No. 1, pp. 527-538, January, 2015.

[22] H. S. Chiang, D. H. Shih, M. H. Shih, J. M. Chang, APN Model Construction for Malicious Email Detection, *Expert Systems with Applications*, Vol. 42, No. 13, pp. 5696-5704, August, 2015.

[23] F. S. Corrêa Da Silva, M. L. Fernández Venero, D. M. David, M. Saleem, P. W. H. Chung, Interaction Protocols for Cross-Organisational Workflows, *Knowledge-Based Systems*, Vol. 37, pp. 121-136, January, 2013.

[24] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, Third Edition, The MIT Press, 2009.

[25] R. Sedgewick, *Algorithms in C, Part 5: Graph Algorithms*, Third Edition, Addison-Wesley Professional, 2001.

[26] T. Murata, Petri Nets: Properties, Analysis and Applications, *Proceedings of the IEEE*, Vol. 77, No. 4, pp. 541-580, April, 1989.

[27] S. J. Liu, X. B. Le, Z. Zou, Additional Theorem Proving and Analysis of S-Complement in The Petri Net, *Journal of System Simulation*, Vol. 20, No. S2, pp. 1-5, August, 2008.

[28] J. S. Pan, L. P. Kong, T. W. Sung, P. W. Tsai, V. Snasel, α-Fraction First Strategy for Hierarchical Model in Wireless Sensor Networks, *Journal of Internet Technology*, Vol. 19, No. 6, pp. 1717-1726, November, 2018.

[29] J. S. Pan, C. Y. Lee, A. Sghaier, M. Zeghid, J. Xie, Novel Systolization of Subquadratic Space Complexity Multipliers Based on Toeplitz Matrix–Vector Product Approach, *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 27, No. 7, pp. 1614-1622, July, 2019.

Biographies

Zheng Zou received the B.S. degree and M.S. degree both from Changsha University of Science and Technology, Changsha, China in 2006 and 2010 respectively, and the Ph.D. degree in computer science from Central South University, Changsha, China in 2017. She is currently a lecturer in the Colleague of Mathematics and Informatics, Fujian Normal University, Fuzhou, China. Her research interests include Petri nets and biomedical image processing.

Shi-Jian Liu received the B.S. degree in mathematics from Xiangtan University, Xiangtan, China, in 2006, the M.S. degree in computer science from Changsha University of Science and Technology, Changsha, China, in 2010, and the Ph.D. degree in computer science from Central South University, Changsha, China, in 2015. He is currently an Associate Professor in the School of Information Science and Engineering, Fujian University of Technology, Fuzhou, China. His research interests include Petri nets, mesh/biomedical image processing, and information security.

Jeng-Shyang Pan received the B.S. degree in Electronic Engineering from the National Taiwan University of Science and Technology in 1986, the M. S. degree in Communication Engineering from the National Chiao Tung University, Taiwan in 1988, and the Ph.D. degree in Electrical Engineering from the University of Edinburgh, U.K. in 1996. Currently, he is a Dean for College of Information Science and Engineering, Fujian Universityof Technology and a director of Innovative Information Industry Research Center, Harbin Institute of Technology Shenzhen Graduate School, China. He jointed the editorial board of LNCS Transactions on Data Hiding and Multimedia Security, Journal of Computers, Journal of Information Hiding and Multimedia Signal Processing etc. His current research interests include soft computing, information security and signal processing.