КИБЕРБЕЗОПАСНОСТЬ

Enhancing Cyber Security with Maximum Coupling in Bipartite Graphs

Sargsyan Garegin V.
Candidate of Sciences in Physics and Mathematics
Yerevan State University (Yerevan, RA)
garegin.sargsyan.v@gmail.com
Matevosyan Artur V.
Master of Electronic Design Automation (EDA), 2nd year
National Polytechnic University of Armenia (Yerevan, RA)
garegin.sargsyan.v@gmail.com, artur.matevosyan99@gmail.com
Avetyan David R.
Master of Electronic Design Automation (EDA), 2nd year
National Polytechnic University of Armenia (Yerevan, RA)
avidavetyan15@gmail.com

UDC: 519.174.7; **EDN:** JKCZSU;
DOI: 10.58587/18292437-2023.2-95
Keywords: Network security, Edge, Bipartite graph, C++, Matrix, Security policy, Maximum coupling, Net, QT, signal-slot

Կիբերանվտանգության բարձրացում երկկողմանի գրաֆում առավելագույն զուգակցման մեթոդով

Սարգսյան Գարեգին Վ.
ֆ.մ.գ.թ., Երևանի պետական համալսարան (Երևան, ՀՀ)
garegin.sargsyan.v@gmail.com
Մաթևոսյան Արթուր Վ.
Էլեկտրոնային ավտոմատացված համակարգեր (EDA) մագիստրոս, 2րդ կուրս
Հայաստանի ազգային պոլիտեխնիկական համալսարան (Երևան, ՀՀ)
artur.matevosyan99@gmail.com
Ավետյան Դավիթ Ռ.
Էլեկտրոնային ավտոմատացված համակարգերի (EDA) մագիստրոս, 2րդ կուրս
Հայաստանի ազգային պոլիտեխնիկական համալսարան (Երևան, ՀՀ)
davidavetyan15@gmail.com

Ամփոփագիր. Երկկողմանի կշռված գրաֆում առավելագույն զուգկցման խնդիրը կարևոր գործնական կիրառություն ունի պաշտպանության և անվտանգության ոլորտում: Օրինակ, սահմանում անվտանգությունն ապահովելու ժամանակ խնդիր է ծագում պոտենցիալ սպառնալիքների դիտարկման և դրանց կանխարգելման համար ռեսուրսների բաշխման մեջ: Որտեղ եզրերի կշիռները կարող են ներկայացնել տարբեր ռեսուրսների արդյունավետությունը կոնկրետ սպառնալիքների հայտնաբերման կամ մեղմացման համար: Նման գրաֆում առավելագույն զուգկցումն գտնելը կարող է ապահովել, որ ամենաարդյունավետ ռեսուրսները հատկացվում են ամենակարևոր տարածքներին՝ դրանով իսկ բարելավելով սահմանների անվտանգությունը:
Բացի ռազմական կիրառություններից, երկկողմանի կշռված գրաֆում առավելագույն զուգկցման խնդիրը կարող է նաև ուժեղացնել կապի անվտանգությունը: Ցանցը մոդելավորելով որպես երկկողմանի գրաֆ ՝ որտեղ գագաթները ցանցի բաղադրիչներն են իսկ կողերը դրանց կապերը, առավելագույն զուգակցման որոնման մեթոդը կարող է օգտագործվել անվտանգության հնարավոր խախտումները հայտնաբերելու համար: Գրաֆի ցանկացած անհամապատասխան հանգույց հուշում է, որ բաղադրիչը կարող է վտանգված լինել կամ խոցելի լինել հարձակման համար: Այսպիսով առավելագույն զուգակցման որոնման մեթոդը դարձնում է գործիքը արդյունավետ՝ ցանցի անվտանգության առաջնահերթությունները պարզելու և ցանցը բարելավելու համար:
Այս հոդվածը ներկայացնում է երկկողմանի կշռված գրաֆում առավելագույն զուգակցում արդյունավետորեն գտնելու նոր մեթոդ: Առաջարկվող մեթոդը օգտագործում է կրկնվող մոտեցում, որը հիմնված է ավանդական ուղի մեծացնող ալգորիթմի վրա, բայց ներառում է լրացուցիչ էվրիստիկ և կրճատման տեխնիկա՝ հաշվողական բարդությունը նվազեցնելու համար: Փորձարարական արդյունքները ցույց են տալիս, որ առաջարկվող մեթոդը գերազանցում է գոյություն ունեցող ալգորիթմներին և՛ ճշգրտության, և՛ արագության առումով: Մեթոդի արդյունավետությունն ու ճշգրտությունը դարձնում են այն արժեքավոր գործիք տարբեր ծրագրերում պաշտպանությունն ու անվտանգությունը բարձրացնելու համար, ներառյալ ռազմական գործողությունները և ցանցային անվտանգությունը:

Հանգուցաբառերը՝ ցանցային անվտանգություն, կողմ, երկկողմանի գրաֆ, c++, մատրիցա, անվտանգության ապահովագիր, առավելագույն զուգակցում, ցանց, QT, ազդանշան-հանգույց

Повышение кибербезопасности с максимальной связью в двудольных графах

Саргсян Гарегин В.
Кандидат физико-математических наук
Ереванский государственный университет (Ереван, РА)
garegin.sargsyan.v@gmail.com
Матевосян Артур В.
Магистр автоматизации электронного проектирования (EDA), 2 курс
Национальный политехнический университет Армении (Ереван, РА)
artur.matevosyan99@gmail.com
Аветян Давид Р.
Магистр автоматизации электронного проектирования (EDA), 2 курс
Национальный политехнический университет Армении (Ереван, РА)
davidavetyan15@gmail.com

Аннотация. Проблема максимальной связи в двудольных взвешенных графах имеет важные практические приложения в области обороны и безопасности. Например, в контексте пограничной безопасности проблема возникает при распределении ресурсов для мониторинга и реагирования на потенциальные угрозы, где веса на краях представляют эффективность различных ресурсов для обнаружения или смягчения конкретных угроз. Нахождение максимальной связи в таком графе может гарантировать, что наиболее эффективные ресурсы будут выделены наиболее важным областям, тем самым улучшив безопасность границ.

Помимо военных приложений, проблема максимальной связи в двудольных взвешенных графах также может повысить безопасность сети. Путем моделирования сети в виде двудольного графа с узлами, представляющими компоненты сети, и ребрами, представляющими их соединения, можно использовать метод поиска максимальной связи для выявления потенциальных нарушений безопасности. Любой несопоставленный узел на графике предполагает, что компонент может быть скомпрометирован или уязвим для атаки, что делает метод поиска максимальной связи эффективным инструментом для определения приоритетов и повышения безопасности сети.

В этой статье представлен новый метод эффективного нахождения максимальной связи в двудольном взвешенном графе. В предлагаемом методе используется итеративный подход, основанный на традиционном алгоритме увеличивающего пути, но включающий дополнительные эвристики и методы сокращения для уменьшения вычислительной сложности. Экспериментальные результаты показывают, что предложенный метод превосходит существующие алгоритмы как по точности, так и по скорости. Эффективность и действенность метода делают его ценным инструментом для усиления защиты и безопасности в различных приложениях, включая военные операции и сетевую безопасность.

Ключевые слова: сетевая безопасность, ребро, двусторонний граф, c++, матрица, политика безопасности, максимальная связь, сеть, QT, сигнал-слот

Introduction

Develop and implement a software tool that, upon inputting orthogonal bipartite graphs with the corresponding file description or a graphical interface, will construct their corresponding bipartite graphs, find the maximum matching of the graph, and represent them in the software's graphical environment. The Hungarian (Munkres) algorithm [**1**] is used to find the maximum matching in the bipartite graph.

Let $G(V_1, V_2, E)$ be the given bipartite graph. Consider the steps of the algorithm:

1. The bipartite graph corresponding matrix A is constructed with the following principle: the number of rows is equal to $m = |V_1|$, the number of columns is $n = |V_2|$, $A_j^i = f\left((v_i, u_j)\right), i = 1..m, j = 1..n$, where f is the graph's matching function, and (v_i, u_j) the edge connecting vertices $v_i \in V_1, u_j \in V_2$ is. If these vertices are not adjacent, then $A_j^i = 0$.

2. If the obtained matrix is not square, i.e., m≠n, then zero rows (columns) are added to the matrix to make it square.

3. As the algorithm searches for the maximum matching, all elements of the matrix are multiplied by -1, and their sum is added to the largest value of the initial matrix elements.

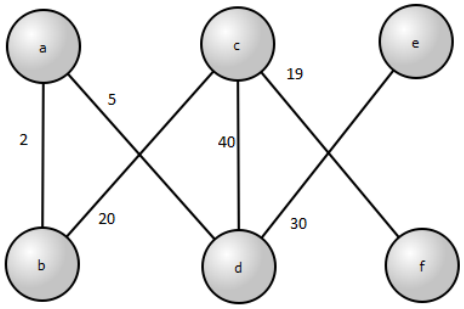
4. For each row, the smallest element is found and subtracted from all the elements of that row.

5. For each column, the smallest element is found and subtracted from all the elements of that column.

6. The minimal number of horizontal (row) lines is determined to cover all the zeros in the matrix. If this number is less than n, the optimal coverage has not been found, and the algorithm

continues with the next step. If this number is equal to n, the optimal coverage has been found, and it is determined according to the rows where the zeros are placed without duplicating the columns.

Figure 1: *Bipartite graph*



7. We are looking for the smallest uncovered element from the matrix, which is subtracted from the uncovered elements, and added to those elements that are located at the intersection points of the horizontal and vertical lines composing the matrix cover. Proceeding to the 6th step of the algorithm. The time complexity of the algorithm is Ь $O(n^2m)$ [**3**]: Let's consider an example of the algorithm's operation on the following bipartite graph:

For graph in Fig 1, $V_1 = \{a, c, e\}, V_2 = \{b, d, f\}$:

1. The matrix will have the following form:
$$\begin{pmatrix} 2 & 5 & 0 \\ 20 & 40 & 19 \\ 0 & 30 & 0 \end{pmatrix}$$
2. The matrix is square, so no action will be performed at this step.
$$\begin{pmatrix} 2 & 5 & 0 \\ 20 & 40 & 19 \\ 0 & 30 & 0 \end{pmatrix}$$
3. After performing the actions of the 3rd step, we get the following matrix:
$$\begin{pmatrix} 38 & 35 & 40 \\ 20 & 0 & 21 \\ 40 & 10 & 40 \end{pmatrix}$$
4. Subtracting the smallest element from each row, we get:
$$\begin{pmatrix} 3 & 0 & 5 \\ 20 & 0 & 21 \\ 30 & 0 & 30 \end{pmatrix}$$
5. Subtracting the smallest element from each column, we get:
$$\begin{pmatrix} 0 & 0 & 0 \\ 17 & 0 & 16 \\ 27 & 0 & 25 \end{pmatrix}$$
6. To cover this matrix, 2 lines are needed, which is less than 3, so we proceed to the next step of the algorithm:
$$\begin{pmatrix} 0 & 0 & 0 \\ 17 & 0 & 16 \\ 27 & 0 & 25 \end{pmatrix}$$
7. The smallest uncovered element is 16, so as a result of the 7th step, we get the following matrix:
$$\begin{pmatrix} 0 & 16 & 0 \\ 1 & 0 & 0 \\ 11 & 0 & 9 \end{pmatrix}$$
8. Repeating the 6th step of the algorithm, we find that the zeros are covered in 3 lines, so the optimal cover is found
$$\begin{pmatrix} 0 & 16 & 0 \\ 1 & 0 & 0 \\ 11 & 0 & 9 \end{pmatrix}$$

According to the lines and columns covering zeros, the following will be:
$$\begin{pmatrix} 0 & 16 & 0 \\ 1 & 0 & 0 \\ 11 & 0 & 9 \end{pmatrix}$$

In the initial matrix:
$$\begin{pmatrix} 2 & 5 & 0 \\ 20 & 40 & 19 \\ 0 & 30 & 0 \end{pmatrix}$$

As a result, we find that the maximum matching will be the following: {(a,b),(c,f),(e,d)}.

Software Implementation

During the execution of the work, a graphical user interface (GUI) for mapping orthogonal line-segment [**2**] domains has been developed, and the algorithm for finding the maximum matching in a bipartite graph generated from the input domains has been implemented. The main components of the software are:

• Implementation of the algorithm

• Graphical interface for displaying orthogonal domains and the corresponding partitions

• Graphical component for representing the partition graph and its maximum matching

During the operation of the software, orthogonal domains are read from an input file or entered by the user, which are stored in a specific data structure and graphically represented. Afterward, when the user presses the "show graph" button, the graph corresponding to the domains is created and its maximum matching is found.

The graphical interface receives the data structure, which describes the bipartite graph and the maximum matching, which are displayed in the graphical environment.

Description of the input file

The file describing the orthogonal directionality domains can have any extension. Below, let's examine an example of an input file describing the domains:

Figure 2: *Example of input file*

test.txt - Notepad

File Edit Format View Help

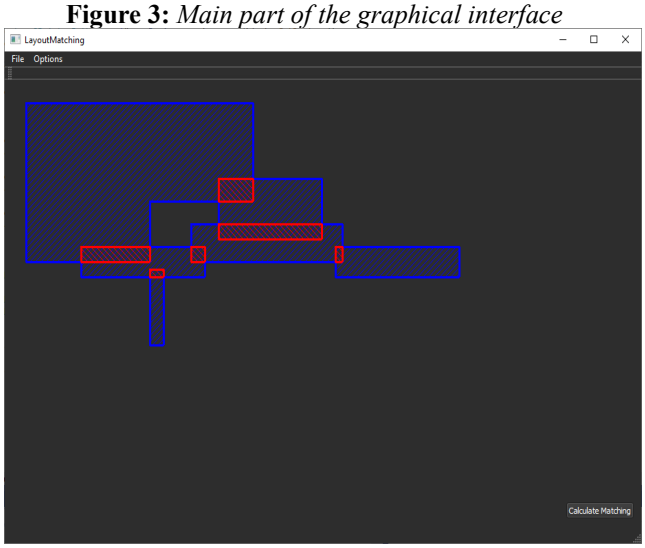
```
{
{(20,20)(20,50)(40,50)}
{(20,20)(20,50)(40,50)(40,20)}
{(100,25)(100,45)(150,45)(150,25)}
{(120,30)(120,60)(140,60)(140,30)}
{(240,240)(240,250)(270,250)(270,240)}
}
{
{(200,200)(200,300)(300,300)(300,200)}
{(210,210)(210,290)(290,290)(290,210)}
{(210,210)(210,280)(280,280)(280,210)}
{(210,210)(210,220)(220,220)}
}
{
{(20,20)(20,50)(40,50)(40,20)}
{(25,25)(25,45)(35,45)(35,25)}
}
{
{(10,25)(10,45)(35,45)(35,25)}
}
{
{(27,30)(27,40)(32,40)(32,30)}
}
{
{(45,25)(45,45)(65,45)(65,25)}
}
```

Structure of the graphical interface

The graphical interface consists of a main part, where the orthogonal directionality domains and their divisions are depicted, and a button that launches the main algorithm [**4**]. Figure 3 presents the appearance of the graphical environment.

The user can load the orthogonal directionality domains described in the input file using the File->Open button. When the user presses the "Calculate Matching" button, a separate window opens, displaying the graph corresponding to the domains and the maximum matching in that graph.

Figure 3: *Main part of the graphical interface*

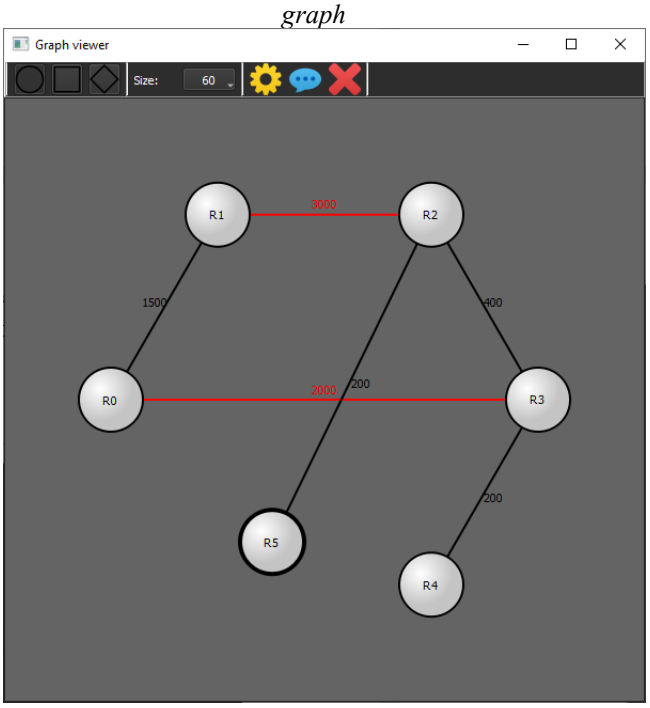


The part for displaying the orthogonal directionality domains consists of the following sections:

• LayoutMatching - a main window inherited from QmainWindow that represents a template containing the basic elements of the graphical interface (QWidget, QPushButton, QStatusBar, etc.). Using Qt's signal-slot mechanism [5], it connects the different parts of the interface, specifically when the "Calculate Matching" button is pressed to calculate the maximum matching, this class's function is called, which, with the help of another part of the program, obtains the graph constructed from the orthogonal directionality domains, passes it to the algorithm, and, as a result, obtaining the maximum matching, opens the window displaying the graph, transferring the information about the graph.

• CanvasWidget – A class inherited from QWidget that is responsible for drawing rectangular areas and their divisions. The base class QWidget's paintEvent(), mousePressEvent(), mouseMoveEvent(), and mouseDoubleClickEvent() functions are overridden to allow the user to add new areas during the program's operation. To add a new area, the user clicks once at the point where the rectangular area begins, then moves the mouse to draw the side, depending on the direction - horizontal or vertical, and then clicks again to start drawing the next side. To finish, the user double-clicks at the endpoint of the last side, which closes the boundary and constructs the area.

Figure 4: *The view of the maximum matching of the graph*



The window representing the graph, by receiving the graph and a certain number of its edges as input, depicts the graph's vertices in a circular layout, while the edges belonging to the

input set of edges are highlighted with a different color. In this window, the user has the ability to change the appearance of the vertices, their sizes, as well as change the arrangement of the vertices relative to each other.

Obtained results of the software tool

By providing the file shown in Fig. 2 as input data to the software tool, the result of the work is the graph depicted in Fig. 4, where the edges forming the maximum matching are shown in a distinct color.

Conclusions

In this work, we have developed an algorithm for finding the maximum matching in a bipartite graph with bent edges, using an orthogonal directionality language to describe areas. Our implementation includes a software tool that graphically displays the input areas and calculates the maximum matching for the corresponding bipartite graph. The tool presents the resulting graph in a graphical environment, with the edges forming the maximum matching displayed in a different color. The software implementation materials are available at the following link: [**6**].

The algorithm can be used in network security to identify potential vulnerabilities or breaches. By modeling a network as a bipartite graph and searching for the maximum matching, any unmatched nodes can indicate potential points of weakness. This makes the maximum matching search algorithm a valuable tool for prioritizing and improving network security.

Overall, our algorithm and software tool offer a flexible and efficient approach to finding the maximum matching in bipartite graphs, with important practical applications in defense and security. The availability of our software implementation materials at the given link will enable others to build on our work and apply the algorithm in a variety of contexts.

References

1. **Stroustrup** - The C++ Programming Language 4th edition, - Addison-Wesley Professional, 2012. – 1040 p.
2. **Mark Allen** -Data structures and algorithm analysis in C++, 2006, 507p
3. **Kuhn H. W**., 50 Years of Integer Programming 1958-2008. Springer, Berlin, Heidelberg, 2010, p29-47
4. **Petrosyan P. A, Mkrtchyan V. V., Kamalyan R. R,** Graph Theory, YSU Publishing House, 2015.
5. **Jasmin Blanchette, Mark Summerfield** –C++ GUI programming with Qt 4 -Prentice Hall PTR, 2006. – 560 p.
6. https://github.com/davidavetyan/LayoutMatching

Сдана/Հանձնվել է՝ 03.03.2023
Рецензирована/Գրախոսվել է՝ 11.04.2023
Принята/Ընդունվել է՝ 17.04.2023