Research on Matching and Vertex Cover Problems in Bipartite Graphs using Simplex Method

Cheuk Hei Josh Yiu*

Dulwich College Shanghai Pudong, Shanghai, 201206, China

*Corresponding author: 15060140424@xs.hnit.edu.cn

Abstract. This paper considers a bipartite graph where a perfect matching does not necessarily exist. Linear programming is used in this paper as a special case of the linear program is the assignment problem, which is another name for the weighted maximum matching problem. The objective is to show that linear programming, in particular the simplex algorithm, can be used to calculate maximum weight matchings and minimum weighted vertex covers. This study also calculates and shows the equivalence of the maximum matching and minimum vertex cover cardinalities, and uses linear programming duality to present its relevance to König's theorem. Finally, Hall's marriage theorem is used to explicitly prove the absence of a perfect matching in particular graphs. There exist many algorithms developed over the years that are designed to solve maximum matching problems in polynomial time, but the simplex method is one of the oldest and simplest algorithms to understand in solving these problems.

Keywords: Bipartite graphs; linear programming; matchings; vertex covers; duality.

1. Introduction

This paper focusses on the mathematical topic of graph theory, specifically a type of graph called a bipartite graph, where all the vertices can be split into two disjoint sets of vertices such that each vertex in any set are only connected to vertices of the opposite set. Bipartite graphs are useful as they have applications in advertising and e-commerce for ranking. They are also used to predict preferences and cancer detection [1].

Matching and vertex cover problems are among the most significant problems within graph theory, and they have many applications within optimization problems and modelling real life situations. Maximum matching problems are used for assignment with maximized scores [2]. Other applications for matchings include scheduling in packet switches, max flow problems and the travelling salesman problem [3]. There are applications for vertex covers in biochemistry, and solving computer security problems or SNP assembly problem [4].

The minimum vertex cover problem is a known NP-hard problem, while the maximum matching problem is also NP-hard in hypergraphs [5, 6]. This paper will focus on the maximum matching and minimum vertex cover problems for bipartite graphs specifically, as both of them are solvable in polynomial time. In a study by Hung, the assignment problem can be solved with at most $n^3 ln\Delta$ pivots, where Δ is the difference between the optimal point and the initial point [7]. There are also several theorems regarding bipartite graphs, two of which are stated and studied in this paper. The purpose of this paper is to demonstrate the correctness of König's and Hall's Theorem.

This paper utilizes the simplex algorithm to solve these linear programs. From studies conducted by Tanaka, it is possible to use the incidence matrix of the bipartite graph to formulate a linear program [8]. However, this is not by any way the fastest algorithm to find a solution. For example, one of the earliest algorithms used is the Hungarian algorithm [9], which has a worse case time complexity of $O(n^3)$. Newer algorithms include deleting constraints, performing a Lagrange relaxation to reduce the problem into a 0-1 knapsack problem or performing a Lagrange decomposition [10]. The Ford-Fulkerson algorithm used for computing flow problems can also be implemented as the matching problem is just a particular case of the minimum flow problem.

2. Methodology

2.1. Terminology, Definitions and Theorems

This paper requires the use of some definitions and theorems that are specifically stated below with technical terminology.

Definition 1 (bipartite graph): A bipartite graph, also known as a bigraph, is a special type of graph where the vertices can be separated into two disjoint sets such that every edge connects vertices from opposite sets [11].

Definition 2 (matching, vertex cover): A matching in a graph is a subset of edges, such that no two edges connect to the same vertex. A vertex cover in a graph is a subset of vertices that touches every edge collectively [12].

In this paper, we are less concerned with the rigorous proofs of the theorems stated below, instead, we use strong duality of linear programming to show the validity of said theorems. Proofs of the two theorems can be found in [13, 14].

Theorem 1 (König): Consider any bipartite graph $G = (V, E)$ where V is the vertex set of the graph and E is the edge set of the graph. Then the maximum matching and minimum vertex cover of G have equal size [2, 13].

Theorem 2 (Hall): Let $G = (V, E)$ be a bipartite graph where the vertex set V is divided into two disjoint sets X and Y. For any subset $T \subseteq X$, the neighbourhood $N(T) \subseteq Y$ of T is defined as the set $N(T) = \{w \in Y : \{v, w\} \in E \; for\; some\; v \in T\}$. If for every $T \subseteq X$, $|N(T)| \geq |T|$ holds, then G has a matching that encompasses all of X's vertices. [2, 14].

Hall's theorem first defines the neighbourhood of any subset T as the subset of vertices within Y that are connected to the vertices in T. Then if for every possible subset of vertices in X, the size of the neighbourhood is larger than the original subset T, then there exists a perfect matching in the original bipartite graph.

2.2. Maximum Weighted Matching

Consider any general bipartite graph. The ultimate goal of a maximum weighted matching is to arrive at the best assignment that maximizes efficiency, which is measured as the greatest sum of edge weights that is included within the matching. An application for a matching in such a graph may be for assignment, where people have to be assigned to positions, and their suitability for the specific positions is denoted by the weights of the edges.

Thus, in the linear program, the variables would be the edges: whether the edge would be included within our assignment or not (the objective function (3)), and the constraints would be the vertices, as every vertex can only accommodate a maximum of 1 edge (constraint matrix (4)).

Technical terminology must be introduced to construct this problem into a linear program. There exists a bipartite graph G with vertex set $V = X \cup Y$ and edge set E. For each edge i in E, let w_i denote its weight. The subset of edges in the matching will be denoted as M. Introduce variables x_i for every edge i defined as 1 if edge i is included in this matching and 0 otherwise. This is shown in equation (1).

$$x_i = \begin{cases} 1 \; if \; i \in M \\ 0 \; if \; i \notin M \end{cases} \tag{1}$$

The objective function is to maximize the total weights of all the edges that we include in M, shown in equation (2). With this, it is then possible to write the whole integer linear program.

$$Maximize \;\; z = \sum_{i \in E} w_i x_i \tag{2}$$

Where w = the weights of all edges i in edge set E.

$$Subject\ to \qquad \sum_{i \in j} x_i \leq 1, for\ all\ j \in V \tag{3}$$

$$Bounds: \qquad x_i \in \{0,1\} \tag{4}$$

The simplex method for solving linear programs can only be used where variables can take all real values within a certain interval, so we must perform a LP relaxation, by allowing x_i to take all real values between 0 and 1. Because every solution to the integer program is also a solution to the LP relaxation, the answer to this LP relaxation is always an upper bound on the original integer program. In fact, because of the integral flow theorem, if the LP relaxation has a feasible solution, then there must exist an integral optimal solution [8]. The purpose of this work does not extend to the proofs of this theorem.

The LP relaxation program for a general bipartite graph would use the same old objective function (2) and constraints (3), but relaxing the bounds to $0 \le x_i \le 1$ instead.

Finding the size of a maximum weightless matching would use the same constraints in equation (3), but ignoring the weights in the objective function, using equation (5) as the new objective function.

$$\text{Maximize } z = \sum_{i \in E} x_i \tag{5}$$

2.3. Minimum weighted vertex cover

The minimum weighted vertex cover problem follows a similar procedure, by first taking a general bipartite graph. The goal is to identify the smallest group of vertices that collectively touch every edge, which can be quantified as the smallest sum of weights in the vertex cover. It is then possible to formulate this problem as a linear program, by letting the vertices be the variables and letting the edges be the constraints, as every edge must touch at least 1 chosen vertex.

Again, technical terminology is introduced to define the integer program. There exists a bipartite graph G with vertex set $V = X \cup Y$ and edge set E. The subset of vertices in the vertex cover is denoted as N. For every vertex i in V, let w_i denote its weight. Introduce variables y_i defined as 1 if vertex i is included in our vertex cover and 0 otherwise, shown in equation (5).

$$y_i = \begin{cases} 1 \; if \; i \in M \\ 0 \; if \; i \notin M \end{cases} \tag{6}$$

The objective function (6) is to minimize the total weights of vertices that we include in N. The whole integer linear program is stated below.

$$Minimize\ z' = \sum_{i \in V} w_i y_i \tag{7}$$

$$Subject\ to \qquad \sum_{i \in j} y_i \ge 1, for\ all\ j \in E \tag{8}$$

$$Bounds: \qquad y_i \in \{0,1\} \tag{9}$$

where w = the weights of all vertices i in vertex set V.

Again, a LP relaxation is required for the same reasons stated in 2.2, yielding the general linear program with same objective function (7), same constraints (8) but bounds $0 \le y_i \le 1$. Finding the minimum weightless vertex cover would use the same constraints in equation (8), but again ignoring the weights in the objective function, using equation (10) as the new objective function.

$$Maximize\ z = \sum_{i \in V} y_i \tag{10}$$

2.4. Neighborhoods

Hall’s theorem defines the neighbourhood $N(T) \subseteq Y$ of T for a set $T \subseteq X$, as the set $N(T) = \{w \in Y : \{v, w\} \in E$ for some $v \in T\}$. It is feasible to use an algorithm to go through all potential subsets $T \subseteq X$ for relatively small graphs and calculate the size of the subsets T and N(T), even if the algorithm had exponential time complexity. The overview of the algorithm is shown in the flowchart in Figure 1.

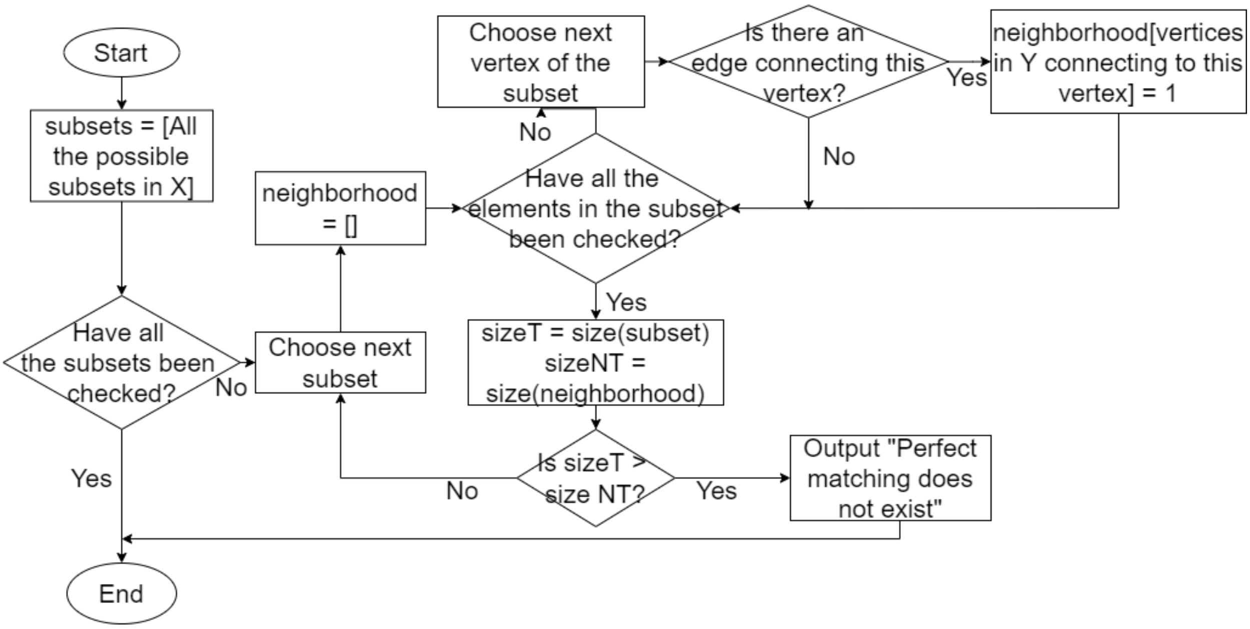

Fig. 1 Flowchart of Hall's theorem algorithm

3. Results and Discussion

3.1. Linear Programming

Because all the constraints are linear, vertex cover and matching problems in bipartite graphs may be resolved using linear programming: in the matching problem, the constraints are that every vertex can accommodate for a maximum of one edge; and every edge must touch at least one vertex for the vertex cover problem.

This original program is an integer linear program, which is a NP-complete problem. Performing a LP relaxation does help reduce the integer program into a normal linear program, but there exist optimal solutions to this linear program that include fractional values for the variables, which makes it inapplicable to the integer program. However, this issue may be overcome by demonstrating the incidence matrix of the bipartite graph's total unimodularity.

Definition 3 (totally unimodular): Every square submatrix of a matrix with determinant 0, 1, or -1 is said to be totally unimodular.

This total unimodularity can be proved by induction. Consider a $n \times n$ submatrix of the original matrix. When n = 1, the determinant is trivially 1 or 0. When n > 1, every column has at most 2 ones, as every edge connects with exactly 2 vertices. Therefore, we can split this problem into 3 cases.

Firstly, there may exist a column with 0 ones. Then the determinant of this submatrix is 0. Secondly, there may exist a column with 1 one. Then by performing the Laplace expansion, We discover that the $(n-1) * (n-1)$ submatrix's determinant and this submatrix's determinant share the same absolute value, which is 0, 1 or -1 by induction. Lastly, the final possibility is that every column has 2 ones. As every row in the incidence matrix corresponds to an edge connecting with exactly 2 vertices, It implies that adding up all the rows corresponding to vertices in X and in Y are identical, equal to $\{1,1,\ldots,1\}$. A similar argument can be said for the columns in the incidence matrix. Therefore, there exists linear dependency in the incidence matrix, so the determinant of the matrix is 0.

By Cramer's rule, if the incidence matrix of the bipartite graph is totally unimodular, then there exists an integral optimal feasible solution. A similar conclusion can be derived from the integral flow theorem.

3.2. Maximum weighted matching of a specific graph

The bipartite graph in Fig. 2 is considered, where vertices are split into two groups X and Y (shown by top and bottom), and connected by some set of edges E with various weights w_i.

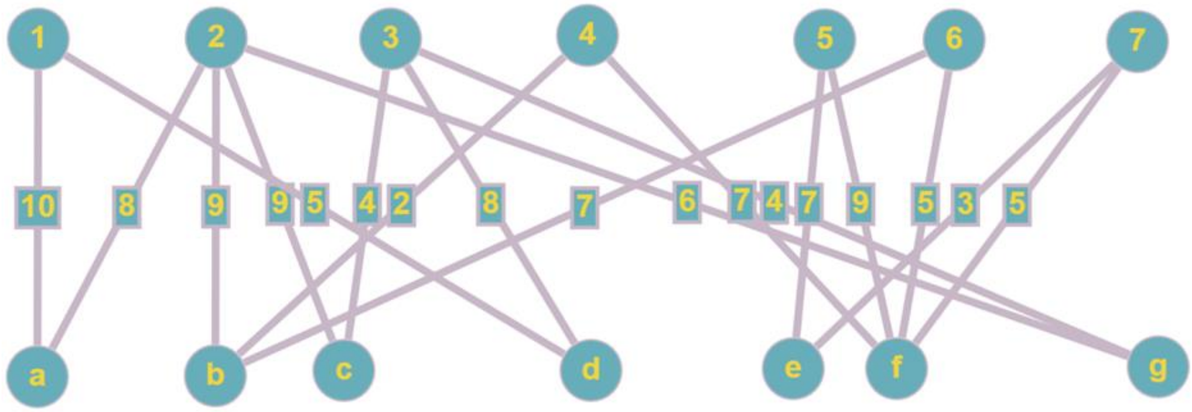

Fig. 2 Initial Bipartite graph

To find the maximum weighted matching of this particular graph, the notation and method in 2.2 is used to formulate the integer program. By enumerating all the edges from left to right order $i \in \{1,2, \dots 17\}$, it is possible to expand the objective function and constraints.

$$Maximize\ \ z = \sum_{i \in E} w_i x_i \tag{11}$$

$$Where\ w = [10, 5, 8, 9, 9, 6, 4, 8, 4, 2, 7, 7, 9, 7, 5, 3, 5].$$

$$\begin{array}{lll} Subject\ to & x_1 + x_2 \leq 1 & x_1 + x_3 \leq 1 \\ & x_3 + x_4 + x_5 + x_6 \leq 1 & x_4 + x_{10} + x_{14} \leq 1 \\ & x_7 + x_8 + x_9 \leq 1 & x_5 + x_7 \leq 1 \\ & x_{10} + x_{11} \leq 1 & x_2 + x_8 \leq 1 \\ & x_{12} + x_{13} \leq 1 & x_{12} + x_{16} \leq 1 \\ & x_{14} + x_{15} \leq 1 & x_{11} + x_{13} + x_{15} + x_{17} \leq 1 \\ & x_{16} + x_{17} \leq 1 & x_6 + x_9 \leq 1 \end{array} \tag{12}$$

$$Bounds: \qquad x_i \in \{0,1\} \tag{13}$$

Using the simplex method on the LP relaxation program with objective function (11) and constraints (12) but with bounds $0 \leq x_i \leq 1$ returns the solution $x_1, x_5, x_8, x_{11}, x_{12}, x_{14} = 1$ and setting every other variable to 0, with the objective function maximized at 48. This optimal solution is presented visually in Figure 3.

Finding the cardinality of the maximum matching (weightless) follows a similar procedure, by solving the same program while ignoring the weights of the edges (modifying the objective to $\sum_{i \in E} x_i$ as shown in equation (5)). In both cases, the cardinality of the maximum matching is 6.

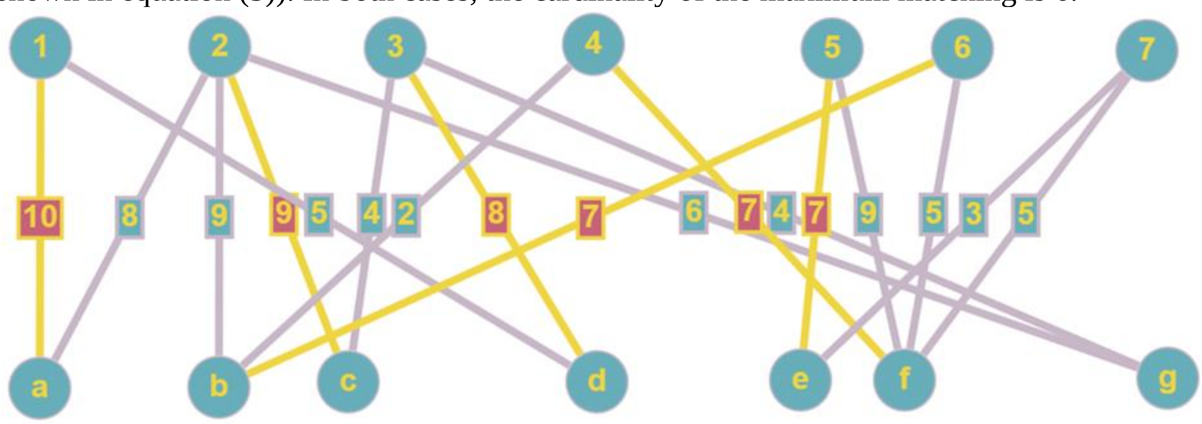

Fig. 3 Maximum weighted matching calculated by simplex method

3.3. Minimum weighted vertex cover of a specific graph

The exact same graph is used for the minimum weighted vertex cover problem, except weights are added to all of the vertices, as shown in Figure 4.

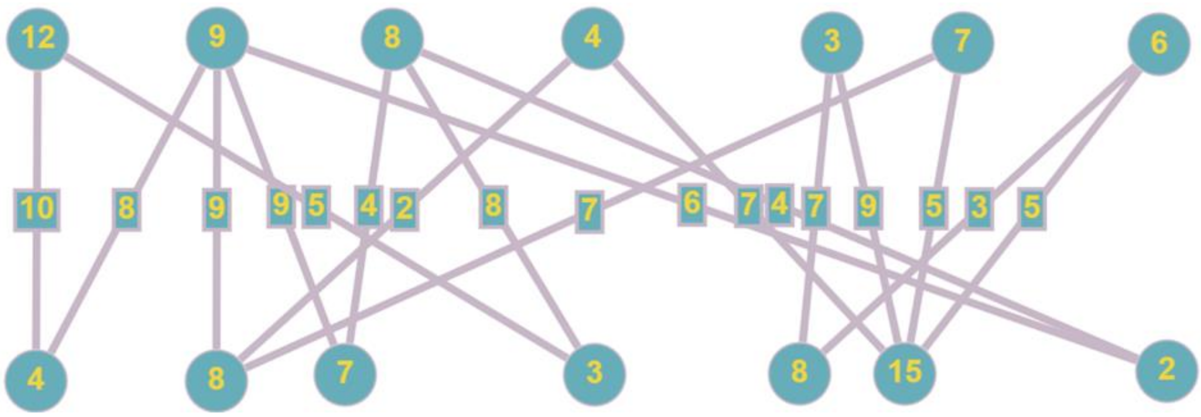

Fig. 4 Bipartite graph with weighted vertices

To determine this graph's smallest weighted vertex cover, the notation and method in 2.3 is used to formulate the integer program. By enumerating all the vertices from left to right order ($i \in \{1,2,\dots 14\}$, it is possible to expand the objective function and constraints.

$$Minimize\ z' = \sum_{i \in V} w_i y_i \tag{14}$$

$$Where\ w = [12, 9, 8, 4, 3, 7, 6, 4, 8, 7, 3, 8, 15, 2]$$

$$\begin{array}{lll} Subject\ to & y_1 + y_8 \geq 1 & y_4 + y_9 \geq 1 \\ & y_1 + y_{11} \geq 1 & y_4 + y_{13} \geq 1 \\ & y_2 + y_8 \geq 1 & y_5 + y_{12} \geq 1 \\ & y_2 + y_9 \geq 1 & y_5 + y_{13} \geq 1 \\ & y_2 + y_{10} \geq 1 & y_6 + y_9 \geq 1 \\ & y_2 + y_{14} \geq 1 & y_6 + y_{13} \geq 1 \\ & y_3 + y_{10} \geq 1 & y_7 + y_{12} \geq 1 \\ & y_3 + y_{11} \geq 1 & y_7 + y_{13} \geq 1 \\ & y_3 + y_{14} \geq 1 & \end{array} \tag{15}$$

$$Bounds: \quad y_i \in \{0,1\} \tag{16}$$

Again, a LP relaxation is required to use the simplex method. Thus, solving the linear program with objective function (14) and constraints (15) but with modified bounds $0 \leq y_i \leq 1$ using the simplex method returns the solution $y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}, y_{14} = 1$, and all other variables set to 0, with the objective function minimized at 44. This solution found by the simplex method is shown visually in Figure 5.

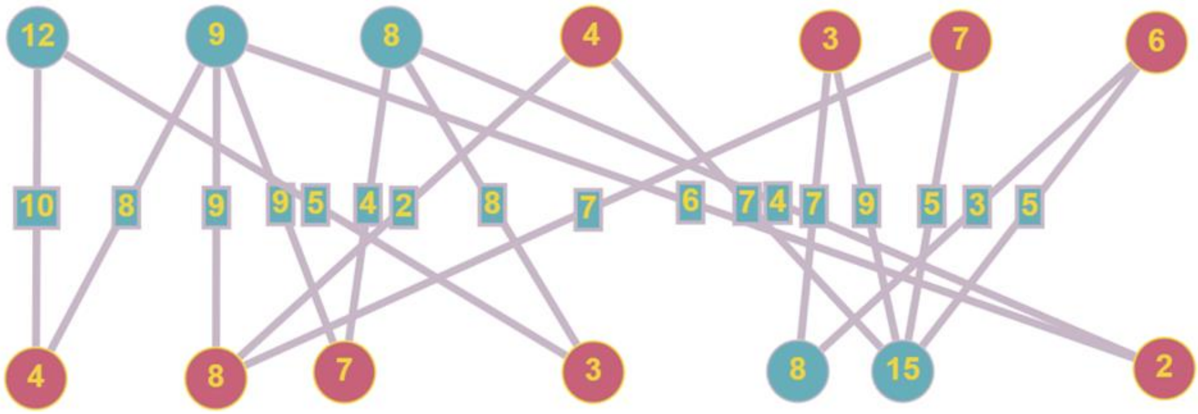

Fig.5 Minimum weighted vertex cover

It is evident that the cardinality of this vertex cover (9) is much larger than what was expected from the previous matching problem, considering the same graph has been used. In fact, if the restriction of weights were removed (by only considering $\sum_{i \in V} y_i$ as the objective function), the simplex algorithm returns a much simpler vertex cover, of size 6, as shown in Figure 6.

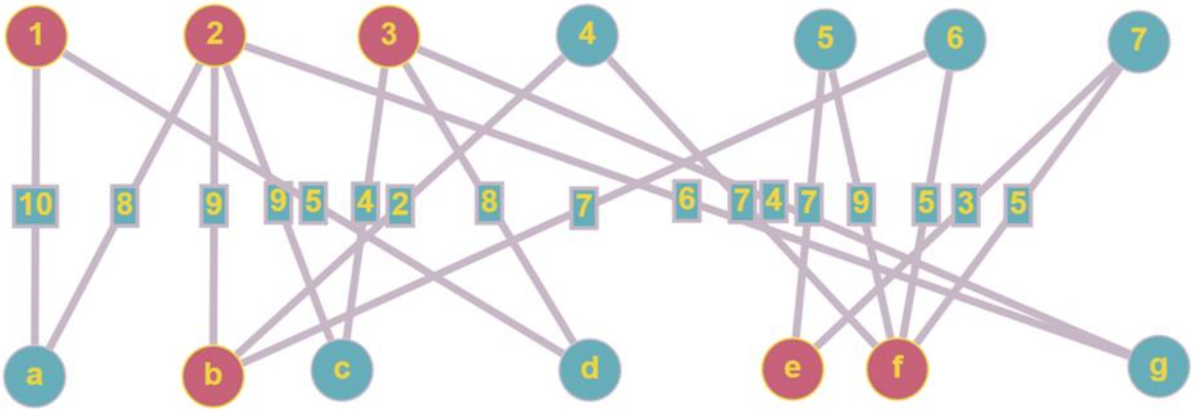

Fig. 6 Minimum weightless vertex cover

The optimal solution for the minimum weightless vertex cover is $y_1, y_2, y_3, y_9, y_{12}, y_{13} = 1$, every other variable being equal to 0 and the objective is minimized at 6.

3.4. König's theorem

By using linear programming, we found that the cardinality of both the maximum matching and minimum vertex cover (unweighted) of the graph in Figure 1 is 6. This corresponds with König's theorem as these sizes are actually equal.

It was established in earlier sections that resolving the matching or vertex cover problem's LP relaxation is sufficient to yield an optimum solution to the initial integer program. We see that after dropping the integrality constraints on the two integer programs, the programs are dual to each other. Therefore, we can use strong duality.

Definition 4 (Strong duality in linear programming): If two programs are dual to each other, exactly one of four possibilities will occur.

Firstly, both programs don't have feasible solutions. Secondly, the primal program is unbounded and the dual program is infeasible. Thirdly, the primal program is infeasible and the dual program is unbounded. Finally, both programs are feasible. If this is the case, then they both have an optimal solution which are equal to each other. By strong duality, we can then deduce that if the programs have an optimum, then the optimal solutions are equal to each other.

3.5. Hall's Theorem

Through using code from any programming language to run the algorithm visually presented in Figure 6 (from section 2.4), the result of a perfect matching not existing in the original bipartite graph in Figure 1 will be returned. This is because the subset {4, 5, 6, 7} in X only corresponds to a neighbourhood {b, e, f}. Therefore, because of the existence of a subset T which has size (4) greater than its neighbourhood N(T) (3), Hall's theorem states that there does not exist a perfect matching within this graph. Figure 7 presents the case where the neighbourhood of T is smaller than the subset T.

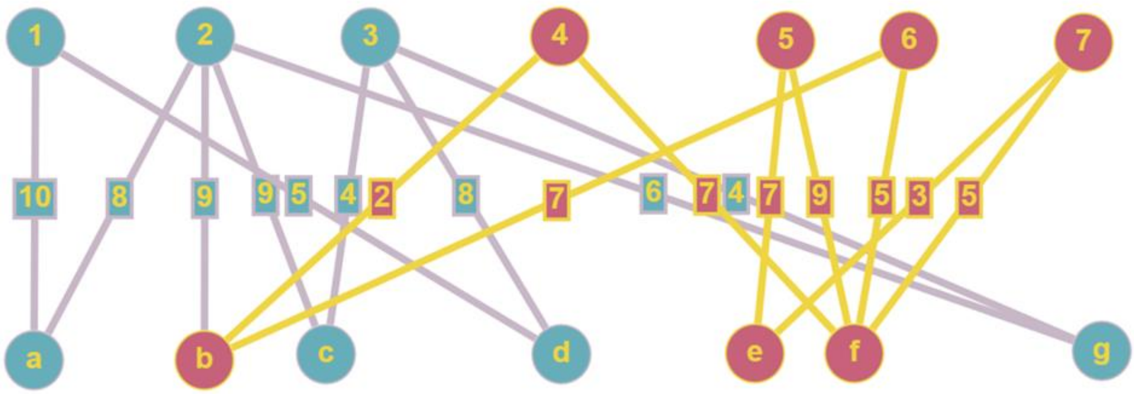

Fig. 7 Visualisation of T and N (T)

4. Conclusion

In conclusion, maximum matching and minimum vertex cover problems in bipartite graphs may be directly solved using linear programming, especially with the simplex method, in polynomial time. The results gathered from these direct computations also show that König's and Hall's theorem in fact do hold, with regards to a specific balanced bipartite graph with 14 vertices.

With regards to practical uses, the assignment and vertex cover problems have very specific applications, but the simplex method may be too inefficient to deal with bipartite graphs are larger size. Moreover, the methods discussed in this paper cannot be applied to general graphs. Therefore, future studies can investigate other more efficient methods stated in the introduction to reach faster results, and also study applying these methods to general graphs.

References

[1] Kench R. Bipartite Graphs. [online] Study.com. Available at: https://study.com/learn/lesson/bipartite-graph-applications-examples.html, [Accessed 18 August 2022]. 2022.

[2] Matousek J. and GärtnerB. Understanding and using linear programming. Springer Science & Business Media. 2006.

[3] Turner J. Applications of Matchings. [online] Available at: https://www.arl.wustl.edu/~jon.turner/cse/542/text/sec13.pdf, [Accessed 18 August 2022]. 2013.

[4] Dharwadker A. and Pirzada S. Applications of Graph Theory. [online] Available at: https://www.dharwadker.org/pirzada/applications/, [Accessed 18 August 2022]. 2007.

[5] Dinur I. and Safra S. On the hardness of approximating minimum vertex cover. Annals of mathematics, 2005, pp.439-485.

[6] M, P. and user53923. Is maximum matching problem equivalent to maximum independent set problem in its dual graph? [online] Computer Science Stack Exchange. Available at: <https://cs.stackexchange.com/questions/69846/is-maximum-matching-problem-equivalent-to-maximum-independent-set-problem-in-its> [Accessed 18 August 2022]. 2017.

[7] Hung M.S. A polynomial simplex method for the assignment problem. Operations Research, 1983, 31(3), pp.595-600.

[8] Tanaka Y. Using the simplex method for a type of allocation problems. American Journal of Computational Mathematics, 2019, 9(2), pp.25-31.

[9] Kuhn H. W. The Hungarian method for the assignment problem. Naval research logistics quarterly, 1955, 2, pp.83-97.

[10] Cattrysse D. G. and Van Wassenhove L. N. A survey of algorithms for the generalized assignment problem. European journal of operational research, 1992, 60(3), pp.260-272.

[11] Bipartite Graph. [online] Available at: <https://mathworld.wolfram.com/BipartiteGraph.html> [Accessed 18 August 2022]. 2022.

[12] Ahmadi A. Bipartite matching and vertex covers. [online] Available at: <https://www.princeton.edu/~aaa/Public/Teaching/ORF523/ORF523_Lec6.pdf.> [Accessed 18 August 2022].

[13] König D. Theorie der endlichen und unendlichen Graphen, Akad. Verlagsgeselschaft, Leipzig. 1936.

[14] Hall P. On representatives of subsets. Classic Papers in Combinatorics, 1987, pp.58-62.

[15] Metatrix et al. Python recursive function to display all subsets of given set. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/26332412/python-recursive-function-to-display-all-subsets-of-given-set.> [Accessed 18 August 2022]. 2014.