Bi-criteria Optimization of Matchings in Trees with Application to Kidney Exchange

Michal Mankowski and Mikhail Moshkov

King Abdullah University of Science and Technology (KAUST)
Computer, Electrical and Mathematical Sciences & Engineering Division
Thuwal 23955-6900, Saudi Arabia
{michal.mankowski, mikhail.moshkov}@kaust.edu.sa

Abstract. In this paper, we propose a method for bi-criteria optimization of matchings in a tree relative to a pair of different weight functions that assign natural weights to the edges of the tree. As a result, we obtain the set of Pareto optimal points for the considered optimization problem. This method can be useful in transplantology where nodes of the tree correspond to pairs (donor, recipient) and two nodes (pairs) are connected by an edge if these pairs can exchange kidneys. Weight functions can characterize the number of exchanges, the importance of exchanges, or their compatibility.

Keywords: tree, matching, weight function, bi-criteria optimization, Pareto optimal point

1 Introduction

In this paper, we consider problems of matching optimization connected with kidney paired donation [3, 11]. This is a novel alternative for living, incompatible (donor, recipient) pairs to get an organ by matching with another incompatible pair. Let G be an undirected graph which edges and nodes have natural weights. Nodes of this graph can be interpreted as pairs (donor, recipient) and two nodes $A = (a_1, a_2)$ and $B = (b_1, b_2)$ are connected by an edge if the donor a_1 can donate a kidney to the recipient b_2, and the donor b_1 can donate a kidney to the recipient a_2. The weight of a node A can be interpreted as an importance of the transplantation for the recipient from the pair A. The weight of an edge connecting nodes A and B can be interpreted as a compatibility of the exchange of kidneys between the pairs A and B.

A matching in G is a set of edges without common nodes. We consider three optimization problems connected with matchings: (c) maximization of the cardinality of a matching, (n) maximization of the sum of weights of nodes in a matching, and (e) maximization of the sum of weights of edges in a matching. The considered problems can be solved in polynomial time [2, 7]. It is well known that there exist dynamic programming algorithms for solving these problems for trees which have linear time complexity (see, for example, [4]).

Each solution of the problem (c) allows us to help to the maximum number of patients. It is known [5] that each solution of the problem (n) is also a solution of the problem (c). The situation with the problem (e) is different: a solution of the problem (e) can be not a solution of the problem (c) (see Fig. 1).

Fig. 1. Matching with maximum weight of edges (a) and matching with maximum cardinality (b)

In such a situation, it is reasonable to consider bi-criteria optimization of matchings relative to the cardinality of matching and the sum of weights of edges in matching. Another possibility is to consider the sum of weights of edges in matching and the sum of weights of nodes in matching. All the considered criteria should be maximized.

The result of bi-criteria optimization is the set of Pareto optimal points (POPs). If we consider the bi-criteria optimization relative to the cardinality of matching and the sum of weights of nodes in matching, we will have only one POP that correspond to matchings with maximum sum of weights of nodes and with maximum cardinality (it follows from results obtained in [5]). In the case of bi-criteria optimization relative to the cardinality of matching and the sum of weights of edges in matching, we can have more than one POPs. For example, for the problem presented in Fig. 1, there are exactly two POPs $(1, 3)$ and $(2, 2)$ corresponding to matchings (a) and (b) in Fig. 1, respectively. The first coordinate correspond to the cardinality of matching, and the second one – to the sum of weights of edges in matching.

The problem (c) can be formulated as the problem (e) when the weight of each edge is equal to 1. The problem (n) can be formulated as the problem (e) when the weight of each edge is equal to the sum of weights of its ends. So we can consider bi-criteria optimization of weights of edges in matchings relative to a pair of weight functions each of which assigns a natural weight to each edge of the graph G.

In this paper, we study a dynamic programming algorithm for bi-criteria optimization of matchings in trees.

The dynamic programming bi-criteria optimization approach was created initially for the decision trees and decision rules [1]. One of the main areas of applications for the approach is the rough set theory [9, 10] in which decision trees and rules are widely used. Here we consider one more its application.

In the paper [8], we studied multi-stage optimization for the same problems. We use here many constructions from this paper.

This paper consists of six sections. In Sect. 2, we discuss some tools for the study of Pareto optimal points. In Sect. 3, we consider a graph $D(G)$ corresponding to the tree G. We use this graph to describe the set of matchings in G and to study these matchings. Section 4 is devoted to the bi-criteria optimization of matchings in G relative to a pair of weight functions. In Sect. 5, we discuss an example. Section 6 contains short conclusions.

2 Tools for Study of Pareto Optimal Points

In this section, we consider some tools (statements and algorithms) which are used for the study of Pareto optimal points for bi-criteria optimization problems related to matchings.

These tools were created in our group in KAUST (details can be found, in particular, in [6]) for the study of minimization problems. To apply these tools to the problems of maximization of edge weight in matchings, it is enough to consider additive inverse of the weights, for example, instead of the weight 5, consider the weight -5.

Let $\mathbb{R}^2$ be the set of pairs of real numbers (*points*). We consider a partial order $\leq$ on the set $\mathbb{R}^2$ (on the plane): $(c, d) \leq (a, b)$ if $c \leq a$ and $d \leq b$. Two points α and β are *comparable* if $\alpha \leq \beta$ or $\beta \leq \alpha$. A subset of $\mathbb{R}^2$ in which no two different points are comparable is called an *antichain*. We will write $\alpha < \beta$ if $\alpha \leq \beta$ and $\alpha \neq \beta$. If α and β are comparable then $\min(\alpha, \beta) = \alpha$ if $\alpha \leq \beta$ and $\min(\alpha, \beta) = \beta$ if $\alpha > \beta$.

Let A be a nonempty finite subset of $\mathbb{R}^2$. A point $\alpha \in A$ is called a *Pareto optimal point (POP) for* A if there is no a point $\beta \in A$ such that $\beta < \alpha$. We denote by $Par(A)$ the set of Pareto optimal points for A. It is clear that $Par(A)$ is an antichain.

Points from $Par(A)$ can be ordered in the following way: $(a_1, b_1), \ldots, (a_t, b_t)$ where $a_1 < \ldots < a_t$. Since points from $Par(A)$ are incomparable, $b_1 > \ldots > b_t$. We will refer to the sequence $(a_1, b_1), \ldots, (a_t, b_t)$ as the *normal representation of the set* $Par(A)$.

We now describe an algorithm which, for a given nonempty finite subset A of the set $\mathbb{R}^2$, constructs the normal representation of the set $Par(A)$. We assume that A is a multiset containing, possibly, repeating elements. The cardinality $|A|$ of A is the total number of elements in A.

Algorithm $\mathcal{A}_1$ (construction of normal representation for the set of POPs).

***Input*:** A nonempty finite subset A of the set $\mathbb{R}^2$ containing, possibly, repeating elements (multiset).

***Output*:** Normal representation P of the set $Par(A)$ of Pareto optimal points for A.

1. Set P equal to the empty sequence.
2. Construct a sequence B of all points from A ordered according to the first coordinate in the ascending order.
3. If there is only one point in the sequence B, then add this point to the end of the sequence P, return P, and finish the work of the algorithm. Otherwise, choose the first $\alpha = (\alpha_1, \alpha_2)$ and the second $\beta = (\beta_1, \beta_2)$ points from B.

4. If α and β are comparable then remove α and β from B, add the point $\min(\alpha, \beta)$ to the beginning of B, and proceed to step 3.
5. If α and β are not comparable (in this case $\alpha_1 < \beta_1$ and $\alpha_2 > \beta_2$) then remove α from B, add the point α to the end of P, and proceed to step 3.

Proposition 1. [6] *Let A be a nonempty finite subset of the set $\mathbb{R}^2$ containing, possibly, repeating elements (multiset). Then the algorithm $\mathcal{A}_1$ returns the normal representation of the set $Par(A)$ of Pareto optimal points for A and makes $O(|A| \log |A|)$ comparisons.*

Lemma 1. [6] *Let $A_1, \ldots, A_k$ be nonempty finite subsets of $\mathbb{R}^2$. Then $Par(A_1 \cup \ldots \cup A_k) \subseteq Par(A_1) \cup \ldots \cup Par(A_k)$.*

Lemma 2. [6] *Let A be a nonempty finite subset of $\mathbb{R}^2$, $B \subseteq A$, and $Par(A) \subseteq B$. Then $Par(B) = Par(A)$.*

Let A, B be nonempty finite subsets of the set $\mathbb{R}^2$. We denote by $A \oplus B$ the set $\{(a + c, b + d) : (a, b) \in A, (c, d) \in B\}$.

Let $P_1, \ldots, P_t$ be nonempty finite subsets of $\mathbb{R}^2$, $Q_1 = P_1$, and, for $i = 2, \ldots, t$, $Q_i = Q_{i-1} \oplus P_i$. We assume that, for $i = 1, \ldots, t$, the sets $Par(P_1), \ldots, Par(P_t)$ are already constructed. We now describe an algorithm that constructs the sets $Par(Q_1), \ldots, Par(Q_t)$ and returns $Par(Q_t)$.

Algorithm $\mathcal{A}_2$ (fusion of sets of POPs).

***Input*:** Sets $Par(P_1), \ldots, Par(P_t)$ for some nonempty finite subsets $P_1, \ldots, P_t$ of $\mathbb{R}^2$.
***Output*:** The set $Par(Q_t)$ where $Q_1 = P_1$, and, for $i = 2, \ldots, t$, $Q_i = Q_{i-1} \oplus P_i$.

1. Set $B_1 = Par(P_1)$ and set $i = 2$.
2. Construct the multiset
$$A_i = B_{i-1} \oplus Par(P_i) = \{(a + c, b + d)) : (a, b) \in B_{i-1}, (c, d) \in Par(P_i)\}$$
– we will not remove equal pairs from the constructed set.
3. Using algorithm $\mathcal{A}_1$, construct the set $B_i = Par(A_i)$.
4. If $i = t$ then return B_i and finish the work of the algorithm. Otherwise, set $i = i + 1$ and proceed to step 2.

Proposition 2. [6] *Let $P_1, \ldots, P_t$ be nonempty finite subsets of $\mathbb{R}^2$, $Q_1 = P_1$, and, for $i = 2, \ldots, t$, $Q_i = Q_{i-1} \oplus P_i$. Then the algorithm $\mathcal{A}_2$ returns the set $Par(Q_t)$.*

Proposition 3. [6] *Let $P_1, \ldots, P_t$ be nonempty finite subsets of $\mathbb{R}^2$, $Q_1 = P_1$, and, for $i = 2, \ldots, t$, $Q_i = Q_{i-1} \oplus P_i$. Let $P_i^{(1)} = \{a : (a, b) \in P_i\}$ for $i = 1, \ldots, t$, m be a natural number, and $P_i^{(1)} \subseteq \{0, -1, \ldots, -m\}$ for $i = 1, \ldots, t$. Then, during the construction of the set $Par(Q_t)$, the algorithm $\mathcal{A}_2$ makes $O(t^2 m^2 \log(tm))$ additions and comparisons.*

3 Graph $D(G)$ Corresponding to Tree G

Let G be a tree. A matching in G is a set of edges without common nodes. We choose a node in the tree G as a root. It will be useful for us to consider G as a directed graph with the orientation of edges from the root. Now each node v in G defines a subtree $G(v)$ of G in which v is the root.

We describe now a graph $D(G)$ (forest of directed trees) which will be used to describe the set of matchings in G and to study these matchings. It contains main nodes from G and auxiliary nodes corresponding to the main ones.

Let v be a terminal node of G – see Fig. 2 (a). Then in the graph $D(G)$ there are two nodes v (main) and $v(\emptyset)$ (auxiliary) corresponding to v which are connected by an edge starting in v and entering $v(\emptyset)$ – see Fig. 2 (b).

Fig. 2. Nodes and edge in $D(G)$ (b) corresponding to terminal node v of G (a)

Let v be a nonterminal node of G which has k outgoing edges $e_1, \ldots, e_k$ entering nodes $v_1, \ldots, v_k$, respectively – see Fig. 3 (a). Then in $D(G)$ there are the main node v, $k+1$ auxiliary nodes $v(e_1), \ldots, v(e_k), v(\emptyset)$, and $k+1$ edges starting in v and entering these auxiliary nodes – see Fig. 3 (b).

Let C and D be sets elements of which are also sets. We denote $C \otimes D = \{c \cup d : c \in C, d \in D\}$.

We now correspond a set $M_{D(G)}(u)$ of matchings in G to each node u of $D(G)$. Let v be a terminal node of G. Then $M_{D(G)}(v) = M_{D(G)}(v(\emptyset)) = \{\lambda\}$ where λ is the empty matching. Let v be a nonterminal node of G which has k outgoing edges $e_1, \ldots, e_k$ entering nodes $v_1, \ldots, v_k$, respectively. Then, for $i = 1, \ldots, k$,

$$M_{D(G)}(v(\emptyset)) = \bigotimes_{j \in \{1,\ldots,k\}} M_{D(G)}(v_j), \tag{1}$$

$$M_{D(G)}(v(e_i)) = (\bigotimes_{j \in \{1,\ldots,k\} \setminus \{i\}} M_{D(G)}(v_j)) \otimes M_{D(G)}(v_i(\emptyset)) \otimes \{\{e_i\}\}, \tag{2}$$

$$M_{D(G)}(v) = \bigcup_{\sigma \in \{e_1,\ldots,e_k,\emptyset\}} M_{D(G)}(v(\sigma)). \tag{3}$$

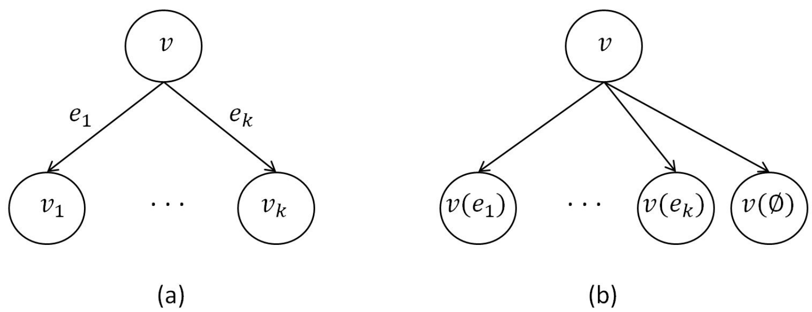

Fig. 3. Nodes and edges in $D(G)$ (b) corresponding to nonterminal node v of G (a)

One can show that, for any node v of G, $M_{D(G)}(v)$ is the set of all matchings in $G(v)$ and $M_{D(G)}(v(\emptyset))$ is the set of all matchings in $G(v)$ which do not use the node v (have no edges with the end v). For any nonterminal node v of G and for any edge e starting in v, $M_{D(G)}(v(e))$ is the set of all matchings in $G(v)$ containing the edge e.

Let G contain n nodes and, therefore, $n-1$ edges. Then the graph $D(G)$ contains $3n-1$ nodes and $2n-1$ edges. It is clear that the graph $D(G)$ can be constructed in linear time depending on n.

4 Bi-criteria Optimization of Matchings

We will consider the problems of minimization of weights of edges in matchings instead of the problems of maximization of weights. To this end, instead of natural weights we will attach to edges additive inverse of weights.

A weight function w for G assigns a negative integer $w(e)$ to each edge e of the tree G. Let μ be a matching in G. The value $w(\mu) = \sum_{e \in \mu} w(e)$ is called the weight of matching μ relative to the weight function w. If $\mu = \lambda = \emptyset$ then $w(\mu) = 0$.

Remark 1. Let w_1 and w_2 be weight functions for G. For any set A of matchings in G, we denote $P^{w_1,w_2}(A) = \{(w_1(\mu), w_2(\mu)) : \mu \in A\}$. Let C and D be sets of matchings in G such that, for any $\mu_1 \in C$ and $\mu_2 \in D$, we have $\mu_1 \cap \mu_2 = \emptyset$. One can show that $P^{w_1,w_2}(C \otimes D) = P^{w_1,w_2}(C) \oplus P^{w_1,w_2}(D)$.

Let w_1 and w_2 be weight functions for G. In the previous section, we corresponded a set $M_{D(G)}(u)$ of matchings in G to each node u of the graph $D(G)$. We now correspond to each node u of $D(G)$ the set of pairs of negative integers $P^{w_1,w_2}_{D(G)}(u) = \{(w_1(\mu), w_2(\mu)) : \mu \in M_{D(G)}(u)\}$.

Let v be a terminal node of G. Then $P^{w_1,w_2}_{D(G)}(v) = P^{w_1,w_2}_{D(G)}(v(\emptyset)) = \{(0, 0)\}$.

Let v be a nonterminal node of G which has k outgoing edges $e_1, \ldots, e_k$ entering nodes $v_1, \ldots, v_k$, respectively.

Using Remark 1 and (1)-(3), we obtain that, for $i = 1, \dots, k$,

$$P^{w_1,w_2}_{D(G)}(v(\emptyset)) = \bigoplus_{j \in \{1,\dots,k\}} P^{w_1,w_2}_{D(G)}(v_j), \tag{4}$$

$$\begin{aligned} P^{w_1,w_2}_{D(G)}(v(e_i)) = &(\bigoplus_{j \in \{1,\dots,k\} \setminus \{i\}} P^{w_1,w_2}_{D(G)}(v_j)) \oplus P^{w_1,w_2}_{D(G)}(v_i(\emptyset)) \\ &\oplus \{(w_1(e_i), w_2(e_i))\}, \end{aligned} \tag{5}$$

$$P^{w_1,w_2}_{D(G)}(v) = \bigcup_{\sigma \in \{e_1,\dots,e_k,\emptyset\}} P^{w_1,w_2}_{D(G)}(v(\sigma)). \tag{6}$$

Let G contain n nodes. We denote $E(G)$ the set of edges of G and $m = n \cdot W$ where $W = \max\{|w_1(e)| : e \in E(G)\}$. It is clear that, for any matching μ in G,

$$w_1(\mu) \in \{0, -1, \dots, -m\}. \tag{7}$$

We describe now a procedure of bi-criteria optimization of matchings in G which, for any node u of $D(G)$, constructs the set $Par(P^{w_1,w_2}_{D(G)}(u))$ of Pareto optimal points for the set of pairs $P^{w_1,w_2}_{D(G)}(u)$.

Let v be a terminal node of G. It is clear that

$$Par(P^{w_1,w_2}_{D(G)}(v)) = Par(P^{w_1,w_2}_{D(G)}(v(\emptyset))) = \{(0,0)\}.$$

So we do not need arithmetical operations to construct the sets of Pareto optimal points for the sets of pairs corresponding to v and $v(\emptyset)$.

Let v be a nonterminal node of G which has k outgoing edges $e_1, \dots, e_k$ entering nodes $v_1, \dots, v_k$, respectively. Let the nodes $v_1, \dots, v_k$ are already processed, i.e., we know the sets of Pareto optimal points for the sets of pairs corresponding to the main nodes $v_1, \dots, v_k$ and their auxiliary nodes.

We now evaluate the number of arithmetical operations (additions and comparisons) required to construct the sets of Pareto optimal points for the sets of pairs corresponding to main node v and its auxiliary nodes $v(e_1), \dots, v(e_k), v(\emptyset)$.

To construct $Par(P^{w_1,w_2}_{D(G)}(v(\emptyset)))$ we use the equality (4) and apply the algorithm $\mathcal{A}_2$ to the sets $Par(P^{w_1,w_2}_{D(G)}(v_1)), \dots, Par(P^{w_1,w_2}_{D(G)}(v_k))$. As a result, we obtain the set $Par(P^{w_1,w_2}_{D(G)}(v(\emptyset)))$ – see Proposition 2. Using (7) and Proposition 3 we obtain that the construction of the set $Par(P^{w_1,w_2}_{D(G)}(v(\emptyset)))$ requires $O(k^2 m^2 \log(km))$ additions and comparisons.

Let $i \in \{1, \dots, k\}$. To construct $Par(P^{w_1,w_2}_{D(G)}(v(e_i)))$ we use the equality (5) and apply the algorithm $\mathcal{A}_2$ to the sets $Par(P^{w_1,w_2}_{D(G)}(v_1)), \dots, Par(P^{w_1,w_2}_{D(G)}(v_{i-1}))$, $Par(P^{w_1,w_2}_{D(G)}(v_i(\emptyset))), Par(P^{w_1,w_2}_{D(G)}(v_{i+1})), \dots, Par(P^{w_1,w_2}_{D(G)}(v_k)), \{(w_1(e_i), w_2(e_i))\}$. As a result, we obtain the set $Par(P^{w_1,w_2}_{D(G)}(v(e_i)))$ – see Proposition 2. Using (7) and Proposition 3 we obtain that the construction of the set $Par(P^{w_1,w_2}_{D(G)}(v(e_i)))$ requires $O(k^2 m^2 \log(km))$ additions and comparisons. To construct the set

$$Par(P^{w_1,w_2}_{D(G)}(v(e_i)))$$

for each $i \in \{1, \ldots, k\}$ we need $O(k^3m^2 \log(km))$ additions and comparisons.

To construct $Par(P^{w_1,w_2}_{D(G)}(v))$ we use the equality (6). According to Lemma 1, $Par(P^{w_1,w_2}_{D(G)}(v)) \subseteq \bigcup_{\sigma\in\{e_1,\ldots,e_k,\emptyset\}} Par(P^{w_1,w_2}_{D(G)}(v(\sigma)))$. By Lemma 2,

$$Par(P^{w_1,w_2}_{D(G)}(v)) = Par(\bigcup_{\sigma\in\{e_1,\ldots,e_k,\emptyset\}} Par(P^{w_1,w_2}_{D(G)}(v(\sigma)))).$$

We construct the set $\bigcup_{\sigma\in\{e_1,\ldots,e_k,\emptyset\}} Par(P^{w_1,w_2}_{D(G)}(v(\sigma)))$ which cardinality is, by (7), at most $(k+1)(m+1)$, and apply to it the algorithm $\mathcal{A}_1$. As a result, we obtain the set $Par(P^{w_1,w_2}_{D(G)}(v))$ using $O(km \log(km))$ comparisons – see Proposition 1.

To process a nonterminal node v of G and all auxiliary nodes of $D(G)$ corresponding to v we make $O(k^3m^2 \log(km))$ comparisons and additions. The number of such nodes is at most n. It is clear that $k \leq n$. Therefore, to construct the sets of POPs corresponding to nodes of $D(G)$ we make $O(n^4m^2 \log(nm))$ comparisons and additions, where $m = n \cdot W$ and $W = \max\{|w_1(e)| : e \in E(G)\}$. Then the procedure of bi-criteria optimization requires $O(n^6W^2 \log(nW))$ comparisons and additions. Hence it has polynomial time complexity depending on n and W.

5 Example

We consider an example of matching bi-criteria optimization problem for the tree G depicted in Fig. 4 (a) which has five nodes v_1, v_2, v_3, v_4, v_5 (each node corresponds to a pair (donor, recipient)) and four edges e_1, e_2, e_3, e_4 (two nodes are connected by an edge if pairs corresponding to the nodes can exchange kidneys).

For simplicity, we consider here natural weights and problems of maximization of weights. The first weight function w_1 corresponds the weight 1 to each edge (we maximize the cardinality of matching, i.e., the number of kidney exchanges). The second weight function w_2 corresponds weights $2, 5, 1, 2$ (compatibilities of exchanges) to edges e_1, e_2, e_3, e_4, respectively (we maximize the weight of matching relative to this function, i.e., maximize the sum of compatibilities).

There are two matchings with maximum cardinality $\{e_1, e_3\}$ and $\{e_1, e_4\}$, and one matching $\{e_2\}$ with maximum edge weight relative to w_2.

First, we construct the graph $D(G)$ – see Fig. 4 (b). Next, we apply to the graph $D(G)$ the procedure of bi-criteria optimization relative to the weight functions w_1 and w_2. As a result, each node of $D(G)$ is labeled with the set of POPs corresponding to the set of matchings associated with this node (see Fig. 5). In particular, the whole set of matchings (it is associated with the node v_1) has two POPs $(1, 5)$ and $(2, 4)$. It means that it is possible to choose between one exchange with compatibility five and two exchanges with total compatibility four.

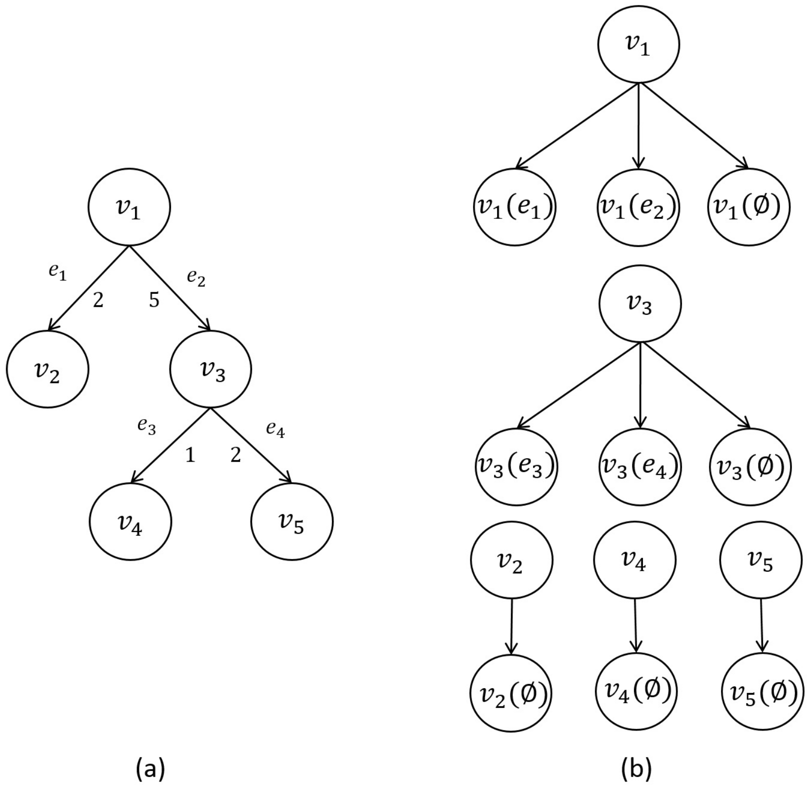

Fig. 4. Graph $D(G)$ (b) corresponding to tree G (a)

6 Conclusions

In this paper, we proposed a procedure of bi-criteria optimization of matchings in trees relative to two weight functions. This method can be useful in transplantology if besides of the maximization of the number of transplanted kidneys we would like to maximize the compatibility of the transplantations. In the future, we will try to generalize this procedure to other classes of graphs.

Acknowledgments

Research reported in this publication was supported by King Abdullah University of Science and Technology (KAUST).

We are indebted to anonymous reviewer for useful suggestions.

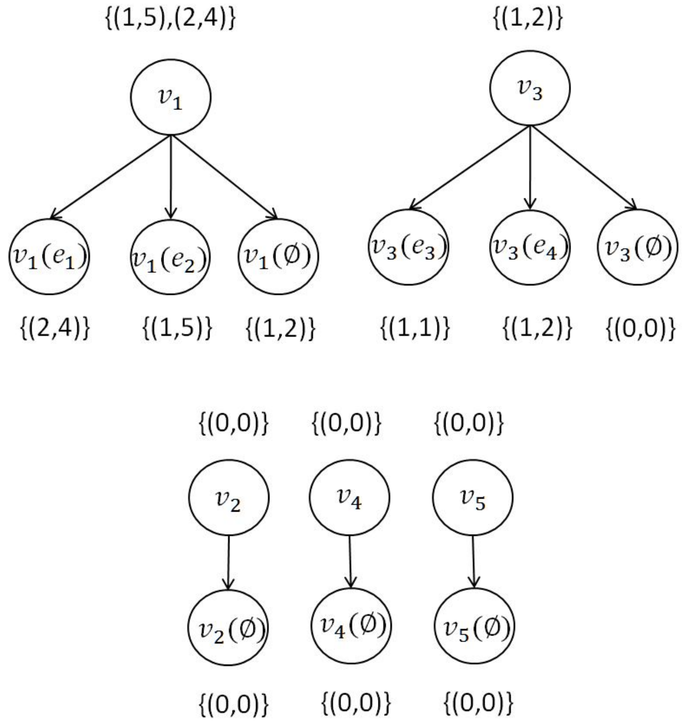

Fig. 5. Sets of POPs attached to nodes of $D(G)$

References

1. Alkhalid, A., Amin, T., Chikalov, I., Hussain, S., Moshkov, M., Zielosko, B.: Optimization and analysis of decision trees and rules: dynamic programming approach. International Journal of General Systems 42(6), 614–634 (2013)
2. Edmonds, J.: Paths, trees, and flowers. Canadian Journal of Mathematics 17, 449–467 (1965)
3. Gentry, S.E.: Optimization over graphs for kidney paired donation. In: Lim, G.J., Lee, E.K. (eds.) Optimization in Medicine and Biology, pp. 177–195. Auerbach Publications, Taylor & Francis Group, Boca Raton, FL (2008)
4. Goddard, W.: Introduction to Algorithms. Part 2: Greedy Algorithms, Dynamic Programming, Graph Algorithms. Clemson University (2004), https://people.cs.clemson.edu/∼goddard/texts/cpsc8400/part2.pdf
5. Halappanavar, M.: Algorithms for Vertex-weighted Matching in Graphs. Ph.D. thesis, Old Dominion University, Norfolk, VA, USA (2009), http://hpc.pnl.gov/people/hala/files/HalappanavarThesisVertexWtdMatching.pdf
6. Hussain, S.: Extensions of Dynamic Programming: Decision Trees, Combinatorial Optimization, and Data Mining. Ph.D. thesis, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia (2016), http://hdl.handle.net/10754/615875

7. Lovász, L., Plummer, M.D.: Matching Theory (AMS Chelsea Publishing). Akadémiai Kiadó - North Holland, Budapest (1986)
8. Mankowski, M., Moshkov, M.: Multi-stage optimization of matchings in trees with application to kidney exchange. In: Polkowski, L., Yao, Y., Artiemjew, P., Ciucci, D., Liu, D., Slezak, D., Zielosko, B. (eds.) Rough Sets - International Joint Conference, IJCRS 2017, Olsztyn, Poland, July 3-7, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10313, pp. 123–130. Springer (2017)
9. Pawlak, Z.: Rough Sets – Theoretical Aspect of Reasoning About Data. Kluwer Academic Publishers, Dordrecht (1991)
10. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
11. Segev, D.L., Gentry, S.E., Warren, D.S., Reeb, B., Montgomery, R.A.: Kidney paired donation and optimizing the use of live donor organs. Journal of the American Medical Association 293(15), 1883–1890 (2005)