Maximum weight cycle packing in optimal kidney exchange programs

Péter Biró, David F. Manlove and Romeo Rizzi

Department of Computing Science
University of Glasgow
Glasgow G12 8QQ
UK

Technical Report
TR-2009-298
April 2009

University of Glasgow | Department of Computing Science

Maximum weight cycle packing in optimal kidney exchange programs

Péter Biró[1,*,†], David F. Manlove[1,*] and Romeo Rizzi[2]

[1] *Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK.*
Email: {pbiro,davidm}@dcs.gla.ac.uk.

[2] *Dipartimento di Matematica ed Informatica (DIMI), Università di Udine, Italy.*
Email: romeo.rizzi@dimi.uniud.it.

Abstract

Centralised matching programs have been established in several countries to organise kidney exchanges between incompatible patient-donor pairs. At the heart of these programs are algorithms to solve kidney exchange problems, which can be modelled as cycle packing problems in a directed graph, involving cycles of length 2, 3, or even longer. Usually the goal is to maximise the number of transplants, but sometimes the total benefit is maximised by considering the differences between suitable kidneys. These problems correspond to computing cycle packings of maximum size or maximum weight. Here we prove the APX-completeness of the problem of finding a maximum size exchange involving only 2-cycles and 3-cycles. We also present an approximation algorithm and two exact algorithms for the problem of finding a maximum weight exchange involving cycles of bounded length. One of the exact algorithms has been used to provide optimal solutions to real kidney exchange problems arising from the National Matching Scheme for Paired Donation run by NHS Blood and Transplant, and we describe practical experience based on this collaboration.

1 Introduction

Transplantation is the most effective treatment that is currently known for kidney failure. A patient who requires a transplant may have a willing donor but often he/she is unable to donate a kidney to the intended recipient because of immunological incompatibilities. However these incompatible patient-donor pairs may be able to exchange kidneys with other pairs in a similar position. Kidney exchange programs have already been established in several countries, for example the USA [28, 2, 29], the Netherlands [21, 22], South Korea [32, 23, 31], Romania [25, 24] and the UK [12, 19].

In the case of many of the current programs [35, 34, 36, 37, 38], the organisers wish to maximise the number of patients who receive a kidney in the exchange by considering only the suitability of the grafts. Other schemes consider also distinctions between suitable kidneys [12]. In this context, some models [33, 9, 10, 7, 8] require, as their primary optimality property, the stability of the solution under various criteria. In a third approach, an exchange is said to be optimal if the sum of the benefits is maximal. This model was described in [39] for *pairwise exchanges* (i.e., exchanges involving only 2-cycles).

*Supported by EPSRC grant EP/E011993/1.
†Supported by OTKA grant K69027.

The lengths of the cycles in the exchanges that arise in kidney exchange programs are bounded in practice, since all operations along a cycle have to be performed simultaneously. Most programs allow only pairwise exchanges, however sometimes 3-cycles are also possible, such as in the New England Program [28] and also in the National Matching Scheme for Paired Donation (NMSPD) administered by the Directorate of Organ Donation and Transplantation, NHS Blood and Transplant (NHSBT) [12]. Furthermore, the Proposal for a National Paired Donation Program in the US [40] declares as a goal the possibility to construct 3-cycles. Moreover, longer cycles may be considered as well: an exchange along a 6-cycle has already been performed in the US [5]. In this paper, we study the problem of maximising the total score of an exchange in situations where 2-cycles, 3-cycles and longer cycles are also allowed. This particular problem variant was also studied independently by Abraham *et al.* [1].

Our contribution in this paper is the following. In Section 2 we give some definitions and preliminary results. Next, in Section 3, we prove that the problem of finding an exchange involving only 2-cycles and 3-cycles, henceforth a *≤3-way exchange*, that covers the maximum number of patient-donor pairs is APX-complete. We note that the NP-hardness of the problem was proved by Abraham *et al.* [1]. Then in Section 4, we give a $(2 + \varepsilon)$-approximation algorithm for the problem of finding a maximum weight ≤3-way exchange. This performance ratio is the best known for this particular problem. The algorithms of Arkin and Hassin [3] and of Berman [6] also achieve this ratio; we claim that our alternative method and analysis is simpler. In Section 5, we present exact algorithms for the maximum weight exchange problems involving cycles of lengths at most 3 and 4, leading to parameterised complexity results. The first exact algorithm has been implemented and successfully used for solving kidney exchange problem instances arising from the NMSPD; this practical experience is described in Section 6. Finally we describe some future work directions in Section 7.

2 Definitions and preliminaries

We model the kidney exchange problem by a simple directed graph $D = (V, A)$, where the vertices $V(D)$ in correspond to the incompatible patient-donor pairs and we have an arc $(u, v) \in A(D)$ if the kidney of u's donor is suitable for v's patient. An *exchange* is a permutation π of V such that, for each $v \in V$, $\pi(v) \neq v$ implies $(v, \pi(v)) \in A$. If a vertex v is involved in a cycle of length at least 2 in π then v is said to be *covered*. The *size* of an exchange π is the number of vertices covered by π. In a *$\leq k$-way exchange* there is no cycle of length more than k. Note that an $\leq k$-way exchange is equivalent to a vertex-disjoint packing of directed cycles of length at most k in D.

Let MAX SIZE $\leq k$-WAY EXCHANGE denote the problem of finding an $\leq k$-way exchange of maximum size. Furthermore, if $w : A(D) \longrightarrow \mathbb{R}_+$ is a weight function on the arcs, then let MAX ARC WEIGHT $\leq k$-WAY EXCHANGE denote the problem of finding a maximum weight $\leq k$-way exchange, where the weight of the exchange is equal to the sum of weights of the arcs in the exchange. Finally, if $\mathcal{C}$ is the set of directed cycles in D, then let $w_C : \mathcal{C} \longrightarrow \mathbb{R}_+$ be a weight function on the cycles of D, and define MAX CYCLE WEIGHT $\leq k$-WAY EXCHANGE to be the problem of finding a maximum weight exchange, where the weight of an exchange is equal to the sum of the weights of the cycles in the exchange.

In the case where we seek an *unbounded exchange* (i.e., there is no upper bound on the lengths of the cycles), we simply omit "$\leq k$-way" from the names of the MAX SIZE $\leq k$-WAY EXCHANGE, MAX ARC WEIGHT $\leq k$-WAY EXCHANGE and MAX CYCLE WEIGHT $\leq k$-WAY EXCHANGE problems when referring the corresponding problems in this more general context.

2.1 Pairwise exchanges

It turns out that MAX SIZE $\leq$2-WAY EXCHANGE, MAX ARC WEIGHT $\leq$2-WAY EXCHANGE and MAX CYCLE WEIGHT $\leq$2-WAY EXCHANGE are all solvable in polynomial time. For, if only 2-cycles are allowed in a given exchange, the problem of finding a so-called *optimal pairwise exchange* becomes a matching problem in an undirected graph G with the same vertex set as D. Here, an edge links two vertices if there is a 2-cycle involving the corresponding two pairs in D. So $\{u, v\} \in E(G)$ if both (u, v) and $(v, u) \in A(D)$. A maximum size pairwise exchange in D corresponds to a maximum cardinality matching in G. The fastest current implementation of Edmonds' algorithm for maximum matching [14] is due to Micali and Vazirani [27], and has $O(\sqrt{n}m)$ complexity, where $n = |V|$ and $m = |E|$. The use of maximum matching in an undirected graph was described as a possible solution strategy for kidney exchange programs in [35]. Similarly, a maximum weight pairwise exchange in D is equivalent to a maximum weight matching in G with weights $w(\{u, v\}) = w(u, v)+w(v, u)$ or $w(\{u, v\}) = w_C(u, v)$, according to whether we are considering arc weights or cycle weights, respectively. The fastest current implementation of Edmonds' algorithm for maximum weight matching [15] is due to Gabow [16], and has complexity $O(n(m + n \log n))$. A detailed description of this problem model for kidney exchange can be found in [39].

2.2 Unbounded exchanges

The MAX SIZE EXCHANGE and MAX ARC WEIGHT EXCHANGE problems are solvable in polynomial time. To see this, we reduce an instance of MAX ARC WEIGHT EXCHANGE to an instance of the maximum weight perfect matching problem in a bipartite graph in the following way. We define a bipartite graph G_{dp} by separating the vertices of D into donors and patients, so $v_d, v_p \in V(G_{dp})$ if $v \in V(D)$. Then we link the donors with the potential recipients and with their original partners: $\{u_d, v_p\} \in E(G_{dp})$ if $(u, v) \in A(D)$ or $u = v$. Let the weight be 0 on the edges between the couples, and let us use the original arc weights otherwise, so $w(\{u_d, v_p\}) = 0$ if $u = v$ and $w(\{u_d, v_p\}) = w(u, v)$ if $(u, v) \in A(D)$. A perfect matching in this bipartite graph corresponds to an exchange in the original problem in a natural way: if $\{u_d, v_p\}$ is in the matching then the donation (u, v) belongs to the exchange, whilst if $\{u_d, u_p\}$ is in the matching then u is uncovered in the exchange. Moreover, the weight of a perfect matching in this bipartite graph is equal to the weight of the corresponding exchange (assuming arc weights rather than cycle weights), so the reduction is complete.[1] A maximum size solution can be found in the same way by using unit weights.

We note that the above reduction can be considered as folklore in the graph theory literature and was also observed by Abraham *et al.* [1]. However, in some recent papers, general integer programming (IP) methods were used to solve the above versions of the kidney exchange problem (see [34, 38]).

In the case of cycle weights, however, the problem of finding a maximum weight unbounded exchange becomes NP-hard. To see this, consider MAX SIZE $\leq$3-WAY EXCHANGE, which is NP-hard as mentioned in Section 1. An instance D of this problem may be transformed to an instance of MAX CYCLE WEIGHT EXCHANGE by creating a weight function w_C on the set of directed cycles $\mathcal{C}$ in D as follows. Let $C' \in \mathcal{C}$. If C' is a 2-cycle, set $w_C(C') = 2$; if C' is a 3-cycle, set $w_C(C') = 3$; otherwise set $w_C(C') = 0$. Clearly π is a

[1]The problem can be reduced to the minimum cost circulation problem as well in the following way. We direct the edges of G_{dp} from u_p to v_d if $u = v$ and from v_d to u_p otherwise. We set lower and upper bounds $f(a) = 0$ and $g(a) = 1$, respectively for each arc and use the same weights as in G_{dp} multiplying by -1. Here, a minimum cost 0-1 circulation corresponds to a maximum weight exchange.

maximum size $\leq$ 3-way exchange in D if and only if π is a maximum weight exchange in D according to the cycle weight function w_C.

3 APX-completeness of MAX SIZE $\leq k$-WAY EXCHANGE

In this section, we show that MAX SIZE $\leq k$-WAY EXCHANGE is APX-complete for any constant $k \geq 3$. We begin in the following subsection by giving some preliminary definitions relating to APX-completeness and we also define the problem that we reduce from.

3.1 Preliminaries

Let Q be an optimisation problem and let A_Q be an approximation algorithm for Q. For every instance x of Q and for every feasible solution y of x, let $c_Q(x, y)$ denote the *cost* of y. For a given instance x of Q, let $opt_Q(x)$ denote the optimal cost of a feasible solution and let $A_Q(x)$ denote the cost of the feasible solution constructed by A_Q. One way to prove that Q cannot be approximated within some fixed constant δ unless P=NP is via an *L-reduction* [30], which we define as follows.

Definition 1. *Let P and Q be two optimisation problems. An* L-reduction *from P to Q is a four-tuple $(t_1, t_2, \alpha, \beta)$ where t_1, t_2 are polynomial time computable functions and α, β are positive constants with the following properties:*

1. *t_1 maps instances of P to instances of Q such that, for every instance x of P, $opt_Q(t_1(x)) \leq \alpha \cdot opt_P(x)$.*
2. *for every instance x of P, t_2 maps pairs $(t_1(x), y')$ (where y' is a feasible solution of $t_1(x)$) to a feasible solution y of x such that*

$$|opt_P(x) - c_P(x, t_2(t_1(x), y'))| \leq \beta |opt_Q(t_1(x)) - c_Q(t_1(x), y')|.$$

If there is an L-reduction from P to Q, we say that P is L-reducible *to Q and denote this by $P \leq_{\mathrm{L}} Q$.*

Let APX denote the class of optimisation problems that are approximable within c, for some constant c. Suppose that Q is a problem in APX. By [4, Lemma 8.2], we may define Q to be *APX-complete* if $P \leq_{\mathrm{L}} Q$ for every problem P in APX. By the transitivity of the L-reduction [30], to show that Q is APX-complete it is sufficient to show that $P \leq_{\mathrm{L}} Q$ for some APX-complete problem P. A consequence of this definition is that if Q is APX-complete then there exists some constant c such that Q is not approximable within c unless P=NP [4, p.261].

In the following subsections we will show that MAX SIZE $\leq k$-WAY EXCHANGE is APX-complete for any constant $k \geq 3$. We firstly remark that this problem belongs to APX for each $k \geq 3$, as we will demonstrate in Section 4. We now define the problem that forms the starting point of our L-reduction that establishes APX-completeness for MAX SIZE $\leq k$-WAY EXCHANGE for each $k \geq 3$.

Let $G = (V, E)$ be a simple and undirected graph. A *triangle* of G is any induced subgraph of G having precisely 3 edges and 3 vertices. A family of triangles $T_1, \ldots, T_l$ of G is called a *vertex-packing of triangles* if $T_1, \ldots, T_l$ are vertex-disjoint. The *size* of this packing is l. The problem of finding a maximum size vertex-packing of triangles in a given graph G, called VERTEX-DISJOINT TRIANGLE PACKING (VDTP), is NP-hard [17].

Moreover, VDTP is APX-complete [20]. The above negative results also hold in the case that we restrict the problem to 3-partite graphs, that is, when the vertex set V of G

is partitioned into three disjoint colour classes $V = A \cup B \cup C$ and no edge of G has its two end-vertices in the same colour class. Notice that when G has this special structure, every triangle of G must have precisely one vertex in each one of the three colour classes.

3.2 Inapproximability of MAX SIZE $\leq$3-WAY EXCHANGE

In this subsection we show that MAX SIZE $\leq$3-WAY EXCHANGE is APX-complete and then in the next subsection we generalise the construction to establish the APX-completeness of MAX SIZE $\leq k$-WAY EXCHANGE for $k \geq 3$. Let $G = (V, E)$ be a 3-partite graph given as an instance of VDTP. Let A, B and C be the 3 colour classes in which V is partitioned. We construct a digraph $D = (V, A)$ by simply orienting the edges of G in a circular way as follows:

$$
\begin{aligned}
A(D) \quad := \quad & \{(u,v) \mid uv \in E(G), u \in A, v \in B\} \cup \\
& \{(u,v) \mid uv \in E(G), u \in B, v \in C\} \cup \\
& \{(u,v) \mid uv \in E(G), u \in C, v \in A\}\,.
\end{aligned}
$$

Clearly, the digraph D can be constructed in polynomial time starting from the graph G. Moreover, the following lemma says that the above is an objective function preserving reduction (a primary case of an L reduction), from which the claimed APX-completeness result follows.

Lemma 2. *The graph G admits a packing of vertex-disjoint triangles covering t vertices if and only if the digraph D admits a $\leq$3-way exchange covering t vertices.*

Proof. Three vertices $u, v, z \in V$ induce a triangle in G if and only if they induce a directed cycle with length 3 in D. As a consequence, a packing of triangles in G can be regarded as a 3-way exchange covering precisely the same set of vertices. In the other direction, notice that D contains no 2-cycles. Thus, any $\leq$3-way exchange contains only cycles of length 3 and can hence be regarded as a packing of triangles in G covering precisely the same set of vertices. □

3.3 Inapproximability of MAX SIZE $\leq k$-WAY EXCHANGE for $k \geq 3$

The trick to generalise the above reduction to a generic $k \geq 3$ is as follows. Once D has been obtained, we obtain a second digraph D' from D as follows. For each arc $a = (u, v)$ of D with $u \in C$ and $v \in A$, add the vertices $w_{a,1}, \ldots, w_{a,k-3}$ and replace the arc $a = (u, v)$ with the arcs $(u, w_{a,1})$, $(w_{a,k-3}, v)$ and $(w_{a,i}, w_{a,i+1})$ for $i = 1, \ldots, k-4$. To summarise, $V(D') := V(D) \cup \{w_{a,i} \mid a = (u,v) \in A(D), u \in A, v \in C, i = 1, \ldots, k-3\}$, and

$$
\begin{aligned}
A(D') \quad := \quad & \{(u,v) \mid uv \in E(G), u \in A, v \in B\} \cup \\
& \{(u,v) \mid uv \in E(G), u \in B, v \in C\} \cup \\
& \{(u, w_{a,1}), (w_{a,k-3}, v) \mid uv \in E(G), u \in C, v \in A\} \cup \\
& \{(w_{a,i}, w_{a,i+1}) \mid uv \in E(G), u \in C, v \in A, i = 1, \ldots, k-3\}\,.
\end{aligned}
$$

Clearly, the digraph D' can be constructed in polynomial time starting from the graph G. Moreover, the following lemma says that the above is an L-reduction, which implies that MAX SIZE $\leq k$-WAY EXCHANGE is APX-complete for any constant $k \geq 3$.

Lemma 3. *The graph G admits a packing of vertex-disjoint triangles covering t vertices if and only if the digraph D' admits a $\leq k$-way exchange covering $tk/3$ vertices.*

Proof. Notice that D' contains no directed cycle of length less then k. Moreover, three vertices $a \in A$, $b \in B$ and $c \in C$ induce a triangle in G if and only if the arc $f = (c, a)$ belongs to D and the vertices $a, b, c, w_{f,1}, \ldots, w_{f,k-4}, w_{f,k-3}$ induce a directed cycle in D'. As a consequence, a packing of triangles in G can be regarded as an $\leq k$-way exchange in D' covering precisely $k/3$ as many vertices. In the other direction, since D' contains no directed cycle of length less then k, then any $\leq k$-way exchange contains only cycles of length k and can hence be regarded as a packing of triangles in G covering precisely $3/k$ as many vertices. □

Lemmas 2 and 3 imply the following theorem.

Theorem 4. MAX SIZE $\leq k$-WAY EXCHANGE *is APX-complete for any integer* $k \geq 3$.

We remark, that MAX ARC WEIGHT $\leq k$-WAY EXCHANGE and MAX CYCLE WEIGHT $\leq k$-WAY EXCHANGE are also APX-complete, since MAX COVER $\leq k$-WAY EXCHANGE is their special case with unit weights.

4 Approximation algorithm for MAX CYCLE WEIGHT $\leq k$-WAY EXCHANGE

In this section we give a $(k - 1 + \varepsilon)$-approximation algorithm for the MAX CYCLE WEIGHT $\leq k$-WAY EXCHANGE problem (and hence the MAX ARC WEIGHT $\leq k$-WAY EXCHANGE problem) for any $\varepsilon > 0$ and for any $k \geq 3$. We begin by showing that MAX CYCLE WEIGHT $\leq k$-WAY EXCHANGE can be reduced to the maximum weight matching problem in a hypergraph. Let $H = (V, E)$ be defined on the same vertex set as D. A hyperedge e_X corresponds to a set of vertices $X \subseteq V(D)$ if $|X| \leq k$ and there exists a directed cycle on X that covers precisely the vertices in X. Let the weight of e_X in H, denoted by $w_H(e_X)$, be equal to the weight of a maximum weight cycle in D that can be formed on X. Obviously, there is a one-to-one correspondence between the $\leq k$-way exchanges in D and the matchings in H. Moreover, the weights of the maximum weight solutions are equal, so that MAXWEIGHT $\leq k$-SET PACKING, defined as follows, is a generalisation. Given a hypergraph $H = (V, E)$ where every hyperedge has size at most k and has a non-negative weight $w_H(e) \in \mathbb{R}_+$ associated to each hyperedge e, define MAXWEIGHT $\leq k$-SET PACKING to be the problem of finding a maximum weight matching M of H.

The MAXWEIGHT $\leq k$-SET PACKING problem can be further reduced to the MAXWEIGHT INDEPENDENT SET problem (the problem of finding a maximum weight independent set) in $(k+1)$-claw free graphs[2], using the *intersection graph* of the hypergraph. In this simple graph, denoted by $L(H)$, the vertices of $L(H)$ are the edges of H, and two vertices of $L(H)$ are adjacent if the corresponding edges of H intersect. The fact that H contains only edges with size at most k implies that $L(H)$ is a $(k + 1)$-claw free graph. We define a weight function on the vertices of $L(H)$ in a natural way: given an edge X of H, let v_X denote the corresponding vertex in $L(H)$. The weight of v_X in $L(H)$, denoted by $w_L(v_X)$, is equal to $w_H(X)$. Obviously, a maximum weight independent set in $L(H)$ corresponds to a maximum weight matching in H (which in turn corresponds to a maximum weight exchange in D). Our goal is to approximate the MAXWEIGHT INDEPENDENT SET problem in $(k + 1)$-claw free graphs.

4.1 Local search method

Let I be an independent set in $L(H)$. A natural idea to improve a sub-optimal solution is the *t-local search* technique (see [3]). In this method, we attempt to add an independent

[2] A graph is $(k + 1)$-*claw-free* if it does not contain $K_{1,k+1}$ as an induced subgraph.

set X to I with cardinality at most t and remove the subset of I that is in $N(X)$, so that the total weight increases. If no such t-local improvement exists then the solution is a *t-local optimum.*

Note, that if we compare two disjoint independent sets, say I and I_{opt}, then these sets can be viewed as the two sides of a bipartite subgraph of $L(H)$. Moreover, each vertex in this subgraph has degree at most k, by the $(k+1)$-claw freeness of $L(H)$. That is why the conditions of the following theorem can describe the relation of a t-local optimum I and a global optimum I_{opt}.

Theorem 5 ([3]). *For any given k and t and every instance $G = (A, B, E)$ satisfying the following three conditions:*

- $|N(a)| \leq k$ *for each* $a \in A$*;*
- $|N(b)| \leq k$ *for each* $b \in B$*;*
- *any subset $X \subseteq A$ of at most t vertices satisfies $w(X) \leq w(N(X))$;*

we have

$$\frac{w(A)}{w(B)} \leq k - 1 + \frac{1}{t}\,.$$

Applying the above theorem to the case where $A = I_{opt}$ and $B = I$ thus implies that a t-local optimum approximates the global optimum within $k-1+\frac{1}{t}$ for any t. It follows that an algorithm based on t-local search also approximates the MAX CYCLE WEIGHT $\leq k$-WAY EXCHANGE problem within a factor of $k - 1 + \varepsilon$ for any $\varepsilon > 0$.

4.2 Local search via augmenting paths

Here we show, that the same performance ratio can be reached by using only a particular t-local search. The approach is straightforward and the same performance ratio as stated in the previous subsection for general local search can be established using a relatively simple argument (the proof in [3] runs to several pages).

The *t-augmenting path search* is a specific type of t-local search, where the new set X is chosen along an alternating path in $L(H)$. Formally, let $X = x_1, x_2, \ldots, x_s$, where $s \leq t$ and there is a subset Y of the actual solution I, such that $|Y| = s - 1$ and $X = x_1, y_1, x_2, y_2, \ldots x_{s-1}, y_{s-1}, x_s$ is a path in the intersection graph, $L(H)$.

Theorem 6. *For any given k and t and every instance $G = (A, B, E)$ satisfying the following three conditions:*

- $|N(a)| \leq k$ *for each* $a \in A$*;*
- $|N(b)| \leq k$ *for each* $b \in B$*;*
- *any subset $X \subseteq A$ of at most t vertices, where X is a set of alternate vertices in an alternating path of G, satisfies $w(X) \leq w(N(X))$;*

we have

$$\frac{w(A)}{w(B)} \leq k - 1 + \frac{2}{t}\,.$$

This theorem implies an alternative proof for the existence of an approximation algorithm – that use "only" augmenting path searches – with factor $k - 1 + \varepsilon$ for any $\varepsilon > 0$.

We use the following well-known lemma in the proof of Theorem 6:

Lemma 7. *If $G = (A, B, E)$ is a k-regular bipartite graph, then the set of edges, $E(G)$ can be partitioned into k disjoint perfect matchings.*

Proof. **(of Theorem 6)** We complete G into a k-regular graph G' by adding dummy vertices with zero weight and some edges. By Lemma 7 we can partition the set of edges into k disjoint perfect matchings. Two of these perfect matchings form a 2-factor, that is a perfect covering by a set of disjoint alternating cycles. Let us fix such a 2-factor $\mathcal{C}$.

The main idea of the proof is that the cycles in $\mathcal{C}$ of length exceeding $2t$ can be cut into alternating paths of length at most $2t$ such that the total weight of the end-vertices of these paths in B is at most $\frac{2}{t}w(B)$. We can achieve this in the following way. Consider a cycle $C_i = (X_i|Y_i) = (x_i^0, y_i^0, x_i^1, \dots, x_i^{c-1}, y_i^{c-1})$ from $\mathcal{C}$, with $|X_i| = |Y_i| = c > t$. We show that we can always find a set of vertices $R_i \subseteq Y_i$ (the so-called *cut vertices* of Y_i) with cardinality $\lceil \frac{c}{t} \rceil$, that satisfies the following properties:

$$\text{If } r_i^j = y_i^p \text{ and } r_i^{j+1} = y_i^{p+s} \text{ are in } R_i \text{ then } s \le t \pmod{c} \tag{1}$$

$$\frac{w(R_i)}{|R_i|} \le \frac{w(Y_i)}{|Y_i|} \tag{2}$$

(Here, property (1) ensures that between two consecutive vertices in R_i the distance is at most $2t$. Property (2) says that the average weight of the vertices in R_i is less than or equal to the average weight of the vertices in Y_i.)

The proof of the existence is straightforward: first we choose a set of vertices from Y_i such that its cardinality satisfies property (1), then we rotate this set along the cycle c times (increasing every index one by one) and we select the set of vertices of minimum total weight to be R_i.

Using property (2) and $c > t$, we obtain

$$w(R_i) \le \frac{|R_i|}{|Y_i|} w(Y_i) = \frac{\lceil \frac{c}{t} \rceil}{c} w(Y_i) < \frac{2}{t} w(Y_i)\,.$$

If we consider the subset of cycles in $C_i \in \mathcal{C}$ of length more than $2t$, then for $R = \cup R_i$ we get

$$w(R) = \sum_{i:|X_i|>t} w(R_i) < \sum_{i:|X_i|>t} \frac{2}{t} w(Y_i) \le \frac{2}{t} w(B). \tag{3}$$

Now, we create a partition of A the following way. First we remove R together with the incident edges from $\mathcal{C}$, and then we also remove the additional vertices and edges (i.e. $G' \setminus G$). The remaining graph, denoted by $\mathcal{P}$, consists of disjoint paths and cycles, where the cycles have length at most $2t$ and each path contains at most t vertices from A. Now, we form partition $\mathcal{A} = X_1 \cup X_2 \cup \ldots \cup X_p$ such that each set $X_i \subseteq A$ consists of vertices that are in the same component of $\mathcal{P}$. Therefore, each set X_i is a set of consecutive vertices in some alternating path of G.

To show that the set of inequalities $w(X_i) \le w(N(X_i))$, for $1 \le i \le p$, imply the statement of the theorem, first we note that any vertex $b \in B$ belongs to at most k sets of the form $N(X_i)$, for $1 \le i \le p$. Moreover, if $b \in B \setminus R$ then b belongs to at most $k-1$ sets of the form $N(X_i)$, for $1 \le i \le p$, since either b has at most one neighbour in $\mathcal{P}$ which implies $|N(b)| \le k-1$, or b has two neighbours in $\mathcal{P}$ which means that these neighbours are in the same class X_i of $\mathcal{A}$. Therefore, by summing up these inequalities and using (3) (since only the vertices of R can be counted k times, from each of their k neighbours), we obtain:

$$\begin{aligned} w(A) &= \sum_{X_i \in \mathcal{A}} w(X) \le \sum_{X_i \in \mathcal{A}} w(N(X)) \\ &\le (k-1)w(B \setminus R) + k\, w(R) \le (k-1)w(B) + w(R) \\ &< \left(k - 1 + \frac{2}{t}\right) w(B)\,. \end{aligned}$$

The proof is complete. □

4.3 Other methods

Chandra and Halldórsson [11] gave a $(2(k+1)/3+\varepsilon)$-approximation algorithm for the MAXWEIGHT INDEPENDENT SET problem in $(k+1)$-claw free graphs by combining a local search method with the greedy algorithm, which we now describe. They start with an independent set I obtained by the greedy algorithm (this algorithm repeatedly chooses a vertex of maximum weight, deleting both it and its neighbours from the graph). Then I is improved by a special type of local search, where the new additional set X is chosen from the neighbours of a vertex from I, in such a way that the ratio of the weights of X and the removed set, $N(X) \cap I$ is always maximal.

Finally, the best approximation known so far for the MAXWEIGHT INDEPENDENT SET problem in $(k+1)$-claw free graphs was given by Berman [6]. He showed that an $((k+1)/2+\varepsilon)$-approximation is possible by a polynomial time algorithm that is based on local improvement by considering the squared total weights. This provides the same approximation ratio for $k=3$ as our local search algorithm via augmenting path method, but has better performance for larger k. Again, the advantage of our method for the case that $k=3$ is that the derivation of the performance ratio is much simpler as compared to [6].

4.4 Approximability of MAX SIZE $\le k$-WAY EXCHANGE

For the approximability of MAX SIZE $\le k$-WAY EXCHANGE, a general result of [18] leads to a polynomial time $(k/2+\varepsilon)$-approximation algorithm (for any $\varepsilon > 0$) for any fixed k. In the special case of $k=3$, this gives a $(3/2+\varepsilon)$-approximation algorithm. An approximation algorithm improving this ratio could be directly translated into an algorithm for VDTP that improves on the best known performance ratio for this problem.

5 Exact algorithms and parameterised complexity

In this section we give exact algorithms for MAX CYCLE WEIGHT $\le$3-WAY EXCHANGE and MAX CYCLE WEIGHT $\le$4-WAY EXCHANGE, leading to parameterised complexity results for these problems.

5.1 MAX CYCLE WEIGHT $\le$3-WAY EXCHANGE

As described in [36, 38], the MAX SIZE $\le$3-WAY EXCHANGE problem is currently solved in the New England Program for Kidney Exchange by IP-based methods. Recently, Abraham *et al.* [1] implemented a specialised IP-heuristic for the MAX CYCLE WEIGHT $\le$3-WAY EXCHANGE problem that would be capable of handling the data of a future national kidney exchange program in the USA (for up to approximately 10,000 couples) according to their simulations. Despite these excellent empirical results, it is still an interesting question to construct an exact combinatorial algorithm for this problem, and we describe such a method in this section. One source of motivation for this is that the IP-technique does not give any guarantee for the running time in a theoretical sense. Another motivating factor is that our alternative technique may also be used as a heuristic in conjunction with other methods.

To describe our method, suppose that we are given an instance of MAX CYCLE WEIGHT $\leq$3-WAY EXCHANGE in a digraph D, and denote by π^* an optimal solution. Let $\{C_1, C_2, \ldots, C_l\}$ be the 3-cycles in π^* and let Y be a set of arcs $\{a_1, a_2, \ldots a_l\}$ where $a_i \in C_i$ $(1 \leq i \leq l)$ so that Y contains one arc from each 3-cycle. We show that with this knowledge of M, we can efficiently find a maximum weight 3-way exchange (i.e. either π^* or another optimal solution).

We transform our instance of MAX CYCLE WEIGHT $\leq$3-WAY EXCHANGE to a maximum weight matching problem in an undirected graph G_Y in the following way. We denote by $V(Y)$ the set of vertices in D that are covered by Y. Let $y_{i,j} \in V(G_Y)$ if $(v_i, v_j) \in Y$, otherwise let $x_i \in V(G_Y)$ if $v_i \in V(D) \setminus V(Y)$. Let $\{x_i, x_j\} \in E(G_Y)$ if both (v_i, v_j) and $(v_j, v_i) \in A(D)$, and let $\{x_k, y_{i,j}\} \in E(G_Y)$ if both (v_k, v_i) and $(v_j, v_k) \in A(D)$. Considering the weights, $w'(\{x_i, x_j\}) := w(v_i, v_j) + w(v_j, v_i)$ and $w'(\{x_k, y_{i,j}\}) := w(v_k, v_i) + w(v_i, v_j) + w(v_j, v_k)$ for the MAX ARC WEIGHT $\leq$3-WAY EXCHANGE problem, and $w'(\{x_i, x_j\}) := w_C(v_i, v_j)$ and $w'(\{x_k, y_{i,j}\}) := w_C(v_k, v_i, v_j)$ for the MAX CYCLE WEIGHT $\leq$3-WAY EXCHANGE problem.

Obviously, a matching M in G_Y corresponds to a $\leq$3-way exchange π in D, in such a way that $\{x_i, x_j\} \in M$ if and only if $\pi(v_i) = v_j$ and $\pi(v_j) = v_i$ is a 2-cycle in D, furthermore $\{x_k, y_{i,j}\} \in M$ if and only if $\pi(v_k) = v_i$, $\pi(v_i) = v_j$ and $\pi(v_j) = v_k$ is a $\leq$3-way exchange in D. Using this correspondence, the weight of a matching M is equal to the weight of the corresponding exchange π. So, if π^* is an optimal solution in D, and Y is a set of arcs as described above, then by solving a maximum weight matching problem in G_Y we can find a matching that corresponds either to π^* or to some other maximum weight $\leq$3-way exchange in D.

An exact algorithm can be constructed using the above idea as follows. First we guess a set of arcs $Y_i \subseteq A(D)$ and then we find a maximum weight matching in G_{Y_i} that corresponds to a feasible solution in D.

Furthermore, it is possible to reduce the set of arcs we choose from by considering only subsets of a set of arcs S, where S contains at least one arc from each 3-cycle in D (so by deleting S from D, the obtained subgraph does not contain any 3-cycle). For if $\{C_1, C_2, \ldots, C_l\}$ are the 3-cycles of an optimal solution π^*, then we can always find a suitable subset $Y_S = \{a_1, a_2, \ldots, a_l\}$ from S, as required. Therefore, if $s = |S|$, then the number of guesses is at most 2^s. (Moreover, since this subset Y_S must contain independent arcs, we need to check only the matchings of S.)

Let us choose S to be of minimum size such that S covers at least one arc from each 3-cycle of D, and consider $s = |S|$ as a parameter. Using the $O(n(m + n \log n))$ time maximum weight matching algorithm (see [16]) as a subroutine, we obtain the following parameterised complexity of MAX CYCLE WEIGHT $\leq$3-WAY EXCHANGE (in the statement of the following theorem, we use $O^*(c^k)$ to refer to $O(c^k f(n))$ in relation to the parameterised complexity of a given problem, where c is a constant, n is the input size and $f(n)$ is a polynomial in n [13].)

Theorem 8. MAX CYCLE WEIGHT $\leq$3-WAY EXCHANGE *can be solved in* $O^*(2^s)$ *time, where* s *is the minimum number of arcs that cover at least one arc from each 3-cycle of* D.

Finding a set S of minimum size satisfying the conditions described above may well be an NP-hard problem in itself. However, we can give an $O^*(3^s)$ parameterised algorithm for this, where $s = |S|$, as follows. First we take a 3-cycle randomly. At the branching stage we select one arc from the three, adding this arc to S and removing it from the graph. Then we continue this brute-force search until no 3-cycle remains. It is easy to see that s will contain at least one arc from each 3-cycle. Furthermore, the depth of the search tree is s, and the detection of a 3-cycle can be carried out in $O(n+m)$ time, therefore we obtain the $O^*(3^s)$ running time.

Finally, we note that $s \leq \frac{m}{2}$ always holds, since every directed graph becomes acyclic by removing at most half of its arcs. As a result, the method described prior to Theorem 8 gives an $O(2^{\frac{m}{2}})$ time exact algorithm for MAX CYCLE WEIGHT $\leq$3-WAY EXCHANGE.

5.2 MAX CYCLE WEIGHT $\leq$4-WAY EXCHANGE

In this subsection, we show how the exact algorithm for MAX CYCLE WEIGHT $\leq$3-WAY EXCHANGE can be generalised for MAX CYCLE WEIGHT $\leq$4-WAY EXCHANGE. Let T be a subset of arcs such that T contains at least two non-consecutive arcs from each 4-cycle and at least one arc from each 3-cycle. Again, we test every matching Y of T in a similar way to the method employed in the previous subsection.

For each set of arcs Y, we transform the MAX CYCLE WEIGHT $\leq$4-WAY EXCHANGE problem to a maximum weight matching problem in an undirected graph G_Y as follows. Again, we denote by $V(Y)$ the set of vertices in D that are covered by Y. Let $y_{i,j} \in V(G_Y)$ if $(v_i, v_j) \in Y$, otherwise let $x_i \in V(G_Y)$ if $v_i \in V(D) \setminus V(Y)$. Let $\{x_i, x_j\} \in E(G_Y)$ if both (v_i, v_j) and $(v_j, v_i) \in A(D)$, and let $\{x_k, y_{i,j}\} \in E(G_Y)$ if both (v_k, v_i) and $(v_j, v_k) \in A(D)$ with the weights as defined above in each case. But here, in addition, let $\{y_{i,j}, y_{k,l}\} \in E(G_Y)$ if both (v_j, v_k) and $(v_l, v_i) \in A(D)$, with weight $w'(\{y_{i,j}, y_{k,l}\}) := w(v_i, v_j) + w(v_j, v_k) + w(v_k, v_l) + w(v_l, v_i)$ for the MAX ARC WEIGHT $\leq$4-WAY EXCHANGE problem, and $w'(\{y_{i,j}, y_{k,l}\}) := w_C(v_i, v_j, v_k, v_l)$ for the MAX CYCLE WEIGHT $\leq$4-WAY EXCHANGE problem.

Here, a matching M in G_Y corresponds to a $\leq$4-way exchange, π in a similar way as we described in the previous subsection, only with the difference that $\{y_{i,j}, y_{k,l}\} \in M$ if and only if $(v_i, v_j, v_k, v_l) \in \pi$. So, an optimal $\leq$4-way exchange, π can be found by testing a particular subset Y^* of T, where Y^* contains one arc from each 3-cycle and two opposite arcs from each 4-cycle of an optimal exchange π^*.

Let us choose T to be of minimum size such that T contains at least two non-consecutive arcs from each 4-cycle and at least one arc from each 3-cycle, and consider $t = |T|$ as a parameter. The searching method, described above, give a parameterised algorithm with the following complexity.

Theorem 9. MAX CYCLE WEIGHT $\leq$4-WAY EXCHANGE *can be solved in* $O^*(2^t)$ *time, where* t *is the minimum number of arcs that cover at least two non-consecutive arcs from each 4-cycle and at least one arc from each 3-cycle of* D.

Again, it is possible to give an $O^*(3^t)$ time parameterised algorithm for the problem of finding such a set T of minimum size satisfying the conditions described above, where $t = |T|$. First we consider the 4-cycles of D. We take one 4-cycle randomly and we select two non-consecutive arcs from this cycle. We mark these arcs, but we do not remove them from D. In each subsequent round we chose a new 4-cycle that does not contain two non-consecutive marked arcs, and we select two non-consecutive arcs (one of them may have been already marked), and we mark them. If every 4-cycle contains two non-consecutive marked arcs then we add the marked arcs to T, and we remove them from D. Afterwards, we continue with the 3-cycles just as we described in the previous subsection. Obviously, the depth of the search tree is t, and we chose between two possibilities when we marked the non-consecutive arcs from each 4-cycles, whilst we had three choices when selecting an arc from each 3-cycle, so the number of possible selections is $O(3^t)$ in total.

6 Practical experience: NHS Blood and Transplant

As part of their administration of the NMSPD, NHSBT maintain a database of incompatible patient-donor pairs who would be willing to participate in a live-donor kidney exchange with one or more other patient-donor pairs. At regular intervals (every three months at the time of writing), a matching run is carried out in which an optimal exchange is constructed from the dataset. At present, exchanges involving only 2-cycles and 3-cycles are sought, though (again at the time of writing), an exchange along a cycle with more than two couples has yet to be carried out in the UK. The term *optimal* refers to the fact that the overriding constraint is to maximise the number of transplants that can be carried out, and subject to this, to maximise the overall score of the exchange. The score of an exchange is based on the points system that NHSBT employs for couples involved in the process – see their web page [12] for further details. This optimisation problem can be reduced to a maximum weight exchange problem by increasing the weight of each arc by an extra weight that is greater than the maximum total weight of any exchange.

We implemented our exact algorithm[3] as described in Section 5.1 for computing an optimal $\leq$ 3-way exchange in C++, using a LEDA implementation of an algorithm for the maximum weight matching problem [26]. We tested this implementation on random samples from a pool of 392 incompatible patient-donor pairs whose data was collected by NHSBT. In particular, we took 10 random samples of n patient-donor pairs from this pool, for n in the range 30 to 50 (in intervals of 5). For each sample, we found an optimal pairwise exchange, an optimal $\leq$ 3-way exchange, and an optimal unbounded exchange.

Table 1 shows the average results that we obtained for each value of n, taken over the 10 instances. The column corresponding to a given value of n shows the following data. Rows 2-4 indicate the average number of arcs in D, and the average number of 2-cycles and 3-cycles in D. Rows 5-7 indicate the average number of 2-cycles in an optimal pairwise exchange, together with the average size and weight of such an exchange. Rows 8-11 show the average number of 2-cycles and 3-cycles in an optimal $\leq$ 3-way exchange, together with the average size and weight of such an exchange. The average size of S, together with the average number of subsets Y of S involved in computing an optimal $\leq$ 3-way exchange are shown in rows 12-13, together with the average time taken by the algorithm in row 14. It turned out that, for some values of n, a handful of the 10 instances in each case gave rise to a much larger running time than the others, so we also indicate the median running time in row 15 (this is actually the "upper" median, i.e., 6th smallest). Finally, rows 16-18 show the average size and weight of an optimal unbounded exchange, together with the average length of the longest cycle that was computed in such an exchange.

All table values are rounded to 1 decimal place apart from the number of subsets Y of S, the average and median running times, which are rounded to the nearest integer. The running times for computing an optimal pairwise exchange and an optimal unbounded exchange for each sampled subset, for each value of n, were all under half a second and are not shown in the table.

The results in the table suggest that the running time of the algorithm for constructing $\leq$ 3-way exchanges rises sharply for instances of size 50 or greater. However, as we will discuss in more detail below, it turned out that the algorithm was able to cope better with

[3] To find a set S, as described in Section 5.1, we used the following heuristic. We selected the arcs to be added to S one-by-one, by choosing the arc involved in the largest number of 3-cycles and removing it from the graph, until no 3-cycle remained.

Number of pairs		30	35	40	45	50
Number of possible donations		168.3	201.3	293.7	351.4	409.8
Total number of	2-cycles	12.4	13.6	19.1	23.3	26.4
	3-cycles	38.8	42.4	76.6	100.7	117.1
Pairwise exchanges	#2-cycles	5.6	5.6	7.7	8.1	8.8
	size	11.2	11.2	15.4	16.2	17.6
	weight	626.8	552.2	861.2	1038.6	1087.2
$\leq$3-way exchanges	#2-cycles	1.7	1.3	2.2	2.4	3
	#3-cycles	4.3	4.8	5.8	6.2	6.6
	size	16.3	17	21.8	23.4	25.8
	weight	966.2	1016.2	1372.4	1493	1667.2
parameters of the exact algorithm	size of S	12.9	13.9	21.5	24	29.4
	$\# \; Y \subseteq S$	1358	997	24137	158555	323924
running time in seconds	average	10	8	250	2370	7457
	upper median	2	7	139	76	938
Unbounded exchanges	size	17.7	19.8	24.9	24.9	28.1
	weight	1196.5	1383.6	1780.0	1888.0	2088.2
	longest c.	11.7	14.3	15.4	15.6	17.3

Table 1: Results obtained for random samples from a large (real) dataset.

real instances involving a larger number of patient-donor pairs, because typically these instances were sparser than the random samples considered here. The average sizes and weights of the optimal $\leq$ 3-way exchanges as compared to those of the optimal pairwise exchanges indicate the clear benefit of allowing 3-cycles in addition to 2-cycles. The lengths of the longest cycles detected in the optimal unbounded exchanges were typically long enough to render these solutions impractical. However not only are these results of theoretical interest, they can be of some practical interest too, since they reveal an upper bound on the improvement that could be obtained (in terms of both the size and weight of an exchange), should cycles of length 4 (or greater) be permitted in the future.

As already observed, whilst allowing 3-cycles makes a significant difference to the sizes and weights of optimal pairwise exchanges, longer cycles do not yield the same level of benefit (at least for the instances considered by these trials). A theoretical argument in support of this observation was given by Roth *et al.* [36] for the general case.

The data in this table concerning the relationship between the sizes and weights of optimal pairwise exchanges and those of optimal $\leq$ 3-way exchanges were a small contributing factor to NHSBT deciding to allow 3-cycles in the NMSPD from March 2008 (prior to this only pairwise exchanges had been sought).

We also used our exact algorithm to find optimal exchanges for NHSBT for the quarterly matching runs of the NMSPD from April 2008 to April 2009 inclusive. The results corresponding to these input datasets are contained in Table 2. This table is organised along similar lines to Table 1 as described above, however we do not include median running times here as only one kidney exchange instance corresponds to each column. The final four rows indicate the exchange that was ultimately chosen by NHSBT – this will be described in more detail below. All entries in Table 2 are rounded to the nearest integer, except for the running times which are rounded to 1 decimal place.

As can be seen from the table, the total number of 2-cycles and 3-cycles in each of the digraphs corresponding to the April and July 2008 instances were small, and in each case

NHSBT were able to find an optimal exchange, using our exact algorithm to verify the optimality of the solution (this was essentially trivial in the case of the July 2008 dataset). However the digraphs corresponding to the October 2008, January 2009 and April 2009 datasets were much richer in terms of the numbers of 2-cycles and 3-cycles. A consequence of this was that the problem instances were much harder to solve by hand, and therefore the results were provided by the exact algorithm in each of these two cases.

In general the real instances were sparser than the random samples described above, because of the large number of highly sensitised patients in the actual pools drawn from the quarterly matching runs. However, despite the number of 2-cycles and 3-cycles in the October 2008, January 2009 and April 2009 datasets being roughly comparable to the average number of 2-cycles and 3-cycles in the random samples of sizes 40, 45 and 50 as shown in Table 1, the running time of the exact algorithm compared much more favourably in the case of the real data. This was primarily a consequence of the size of S remaining relatively small, so that fewer subsets Y of S needed to be considered, and therefore the optimisation problem in each of the cases of the October 2008, January 2009 and April 2009 data was comfortably tractable by our exact algorithm.

We finally note that, in the cases of the October 2008, January 2009 and April 2009 datasets, the actual solutions chosen by NHSBT were different from the optimal (according to the original criteria) $\leq$3-way exchanges. This stemmed from the fact that none of the four 3-cycles identified in April 2008 led to transplants, for a range of reasons (e.g., a positive crossmatch being discovered late in the process following more detailed tests, a patient and/or a donor becoming ill, a patient deciding to proceed with an antibody incompatible transplantation, etc. – for further details about this, see [19]). Due to the risk involved with 3-cycles, our exact algorithm was already computing optimal $\leq$ 3-way exchanges subject to the additional constraint that the exchange could involve at most k 3-cycles, for $0 \leq k \leq k'$, where k' was the number of 3-cycles in an optimal $\leq$ 3-way exchange without this constraint. As such, we were able to compute an optimal $\leq$ 3-way exchange that contained the minimum number of 3-cycles. However from October 2008, NHSBT decided that this was not sufficient, and changed their policy as follows. What is now sought is an optimal $\leq$ 3-way exchange π with the additional constraint that π contains k 2-cycles, where k is the number of 2-cycles in an optimal pairwise exchange. That is, 3-cycles are permitted so long as they do not lead to a reduction in the maximum possible number of 2-cycles. Finding π is not, however, simply a matter of adding 3-cycles to an optimal pairwise exchange, since NHSBT permit the total weight of the 2-cycles involved in an optimal pairwise exchange to decrease in order to accommodate the additional 3-cycles. Bearing this in mind, we were able to compute the alternative solutions shown under "chosen solutions" in Table 2 corresponding to the October 2008, January 2009 and April 2009 datasets using our exact algorithm.

7 Future work

Finding improved approximation algorithms, for each variation of the problems we studied in this paper, remains an important theoretical challenge. It may be worth trying to tackle these problems directly rather than transforming them to the maximum weight independent set problem in $(k + 1)$-claw free graphs.

It would be interesting to see whether our graph-based exact algorithm could be speeded up by additional techniques. One possible approach might be to use a special branch and bound method as follows. Given a subset Y of S and a current optimum found so far during the search, let us find a maximum weight exchange such that each arc of Y is involved in an unbounded exchange. If this relaxed optimum is less than or

Matching run		Apr 08	Jul 08	Oct 08	Jan 09	Apr 09
Number of pairs		76	85	123	126	122
Number of possible donations		287	235	704	576	760
Total number of	2-cycles	5	2	14	16	20
	3 cycles	5	0	109	65	68
Pairwise exchanges	#2-cycles	2	1	6	5	5
	size	4	2	12	10	10
	weight	91	6	499	264	388
$\leq$3-way exchanges	#2-cycles	2	1	2	1	2
	#3-cycles	4	0	7	5	5
	size	16	2	25	17	19
	weight	620	6	1122	633	757
parameters of the exact algorithm	size of S	5	0	18	13	14
	$\#Y \subseteq S$	24	0	3480	588	1440
Running time in seconds		0.3	0.0	66.0	7.5	19.2
Unbounded exchanges	size	22	2	33	28	28
	weight	857	6	1546	1134	1275
	longest c.	20	2	27	19	23
Chosen solution (NHSBT)	#2-cycles	2	1	6	5	5
	#3-cycles	4	0	3	1	2
	size	16	2	21	13	16
	weight	620	6	930	422	618

Table 2: Results arising from matching runs from April 2008 to April 2009.

equal to the current optimum then we can cut the search, since we do not need to test any superset $Y' \supseteq Y$. Furthermore, we would like to compare the running time of our exact algorithm with that of other methods, e.g. the IP-heuristic implemented by Abraham *et al.* [1]. This was shown to be very powerful for generated instances but might be less successful for real data combined with specific optimisation criteria such finding a maximum weight maximum cardinality matching.

Regarding the practical application, one of the current challenges is to incorporate the possibility of altruistic chains. Such a chain starts with an altruistic donor who donates her kidney to a patient p, whilst p's incompatible donor continues the chain, with the final kidney being donated to the deceased donor list. Basically, we can model this extension using our existing techniques by introducing a dummy patient p' for each altruistic donor, such that p' is compatible with every donor in the pool. However, when considering the optimisation criteria, we may need to distinguish between an altruistic chain and a normal cycle of the same length, given that the risk and benefit in each case may differ (for further description about altruistic chains, see e.g., [38]). Another possible change in the application could be the introduction of 4-cycles, which is a long-term plan of NHSBT, giving rise to further algorithmic challenges.

Acknowledgements

We would like to thank Rachel Johnson and Joanne Allen of the Directorate of Organ Donation and Transplantation, NHS Blood and Transplant, for helpful discussions and for the ongoing collaboration. We would also like to thank Kirstin MacDonald for useful observations concerning this paper. Finally, we would like to thank Ildikó Schlotter for helpful comments concerning the parameterised complexity aspects of the exact algorithms of Section 5.

References

[1] D.J. Abraham, A. Blum, and T. Sandholm. Clearing algorithms for barter-exchange markets: Enabling nationwide kidney exchanges. In *Proceeding of EC '07: the Eighth ACM Conference on Electronic Commerce*, pages 295–304. ACM, 2007.

[2] Alliance for Paired Donation website. *http://www.paireddonation.org*.

[3] E.M. Arkin and R. Hassin. On local search for weighted packing problems. *Mathematics of Operation Research*, 23:640–648, 1998.

[4] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation*. Springer, 1999.

[5] BBC news website. "'Six-way' kidney transplant first" (9/04/2008). *http://news.bbc.co.uk/1/health/7338437.stm*.

[6] P. Berman. A $d/2$ approximation for maximum weight independent set in d-claw free graphs. In *Proceedings of SWAT 2000: the 7th Scandinavian Workshop on Algorithm Theory*, volume 1851 of *Lecture Notes in Computer Science*, pages 214–219. Springer, 2000.

[7] P. Biró and K. Cechlárová. Inapproximability of the kidney exchange problem. *Information Processing Letters*, 101(5):199–202, 2007.

[8] P. Biró and E. McDermid. Three-sided stable matchings with cyclic preferences and the kidney exchange problem. In *Proceedings of COMSOC: 2nd International Workshop on Computational Social Choice*, pages 97–108. Department of Computer Science, University of Liverpool, 2008.

[9] K. Cechlárová, T. Fleiner, and D. Manlove. The kidney exchange game. In *Proceedings of SOR '05: the 8th International Symposium on Operations Research in Slovenia*, pages 77–83, 2005. Technical Report, IM Preprint series A, no. 5/2005, PJ Šafárik University, Faculty of Science, Institute of Mathematics, 2005.

[10] K. Cechlárová and V. Lacko. The kidney exchange game: How hard is to find a donor? Technical report, IM Preprint series A, no. 4/2006, PJ Šafárik University, Faculty of Science, Institute of Mathematics, 2006.

[11] B. Chandra and M. Halldórsson. Greedy local improvement and weighted set packing approximation. In *Proceedings of SODA '99: the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 169–176. ACM, 1999.

[12] Directorate of Organ Donation and Transplantation, NHS Blood and Transplant website. *http://www.organdonation.nhs.uk*.

[13] R.G. Downey and M.R. Fellows. *Parameterized complexity*. Monographs in Computer Science. Springer, New York, 1999.

[14] J. Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards Section B*, 69B:125–130, 1965.

[15] J. Edmonds. Paths, trees and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[16] H.N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In *Proceedings of SODA '90: the 1st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 434–443. ACM, 1990.

[17] M.R. Garey and D.S. Johnson. *Computers and intractability.* W. H. Freeman and Co., 1979.

[18] C.A.J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM Journal on Discrete Mathematics*, 2(1):68–72, 1989.

[19] R.J. Johnson, J.E. Allen, S.V. Fuggle, J.A. Bradley, and C.J. Rudge. Early experience of paired living kidney donation in the United Kingdom. *Transplantation*, 86:1672–1677, 2008.

[20] V. Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Information Processing Letters*, 37(1):27–35, 1991.

[21] K.M. Keizer, M. de Klerk, B.J.J.M. Haase-Kromwijk, and W. Weimar. The Dutch algorithm for allocation in living donor kidney exchange. *Transplantation Proceedings*, 37:589–591, 2005.

[22] M. de Klerk, K.M. Keizer, F.H.J. Claas, M. Witvliet, B.J.J.M. Haase-Kromwijk, and W. Weimar. The Dutch national living donor kidney exchange program. *American Journal of Transplantation*, 5:2302–2305, 2005.

[23] J.Y. Kwak, O.J. Kwon, K.S. Lee, C.M. Kang, H.Y. Park, and J.H. Kim. Exchange-donor program in kidney transplantation: A single-center experience. *Transplantation Proceedings*, 31:344–345, 1999.

[24] M. Lucan. Five years of single-center experience with paired kidney exchange transplantation. *Transplantation Proceedings*, 39:1371–1375, 2007.

[25] M. Lucan, P. Rotariu, D. Neculoiu, and G. Iacob. Kidney exchange program: a viable alternative in countries with low rate of cadaver harvesting. *Transplantation Proceedings*, 35:933–934, 2003.

[26] K. Mehlhorn and S. Naher. *LEDA: A Platform for Combinatorial and Geometric Computing.* Cambridge University Press, 1999.

[27] S. Micali and V.V. Vazirani. An $O(\sqrt{|V|}\cdot|E|)$ algorithm for finding maximum matching in general graphs. In *Proceedings of FOCS '80: the 21st Annual IEEE Symposium on Foundations of Computer Science*, pages 17–27. IEEE Computer Society, 1980.

[28] New England Program for Kidney Exchange website. *http://www.nepke.org.*

[29] Paired Donation Network website. *http://www.paireddonationnetwork.org.*

[30] C.H. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[31] K. Park, J.H. Lee, S.I. Kim, and Y.S. Kim. Exchange living-donor kidney transplantation: Diminution of donor organ shortage. *Transplantation Proceedings*, 36:2949–2951, 2004.

[32] K. Park, J.II Moon, S.II Kim, and Y.S. Kim. Exchange-donor program in kidney transplantation. *Transplantation Proceedings*, 31:356–357, 1999.

[33] A.E. Roth, T. Sönmez, and M.U. Ünver. Kidney exchange. *Quarterly Journal of Economics*, 119:457–488, 2004.

[34] A.E. Roth, T. Sönmez, and M.U. Ünver. A kidney exchange clearinghouse in New England. *American Economic Review, Papers and Proceedings*, 95(2):376–380, 2005.

[35] A.E. Roth, T. Sönmez, and M.U. Ünver. Pairwise kidney exchange. *Journal of Economic Theory*, 125(2):151–188, 2005.

[36] A.E. Roth, T. Sönmez, and M.U. Ünver. Coincidence of wants in markets with compatibility based preferences. *American Economic Review*, 97(3):828–851, 2007.

[37] S.L. Saidman, A.E. Roth, T. Sönmez, M.U. Ünver, and S.L. Delmonico. Increasing the opportunity of live kidney donation by matching for two and three way exchanges. *Transplantation*, 81(5):773–782, 2006.

[38] S.L. Saidman, A.E. Roth, T. Sönmez, M.U. Ünver, and S.L. Delmonico. Utilizing list exchange and undirected good samaritan donation through chain paired kidney exchanges. *American Journal of Transplantation*, 6(11):2694–2705, 2006.

[39] S.L. Segev, S.E. Gentry, D.S. Warren, B. Reeb, and R.A. Montgomery. Kidney paired donation and optimizing the use of live donor organs. *Journal of the American Medical Association*, 293:1883–1890, 2005.

[40] United Network for Organ Sharing website. *http://www.unos.org*.