Neural Computing and Applications (2024) 36:22265–22275
https://doi.org/10.1007/s00521-024-10352-6

ORIGINAL ARTICLE

Designing dataless neural networks for kidney exchange variants

Sangram K. Jena[1] · K. Subramani[2] · Alvaro Velasquez[3]

Received: 10 May 2024 / Accepted: 30 July 2024 / Published online: 18 September 2024

Abstract

Kidney transplantation is vital for treating end-stage renal disease, impacting roughly one in a thousand Europeans. The search for a suitable deceased donor often leads to prolonged and uncertain wait times, making living donor transplants a viable alternative. However, approximately 40% of living donors are incompatible with their intended recipients. Therefore, many countries have established kidney exchange programs, allowing patients with incompatible donors to participate in "swap" arrangements, exchanging donors with other patients in similar situations. Several variants of the vertex-disjoint cycle cover problem model the above problem, which deals with different aspects of kidney exchange as required. This paper discusses several specific vertex-disjoint cycle cover variants and deals with finding the exact solution. We employ the dataless neural networks framework to establish single differentiable functions for each variant. Recent research highlights the framework's effectiveness in representing several combinatorial optimization problems. Inspired by these findings, we propose customized dataless neural networks for vertex-disjoint cycle cover variants. We derive a differentiable function for each variant and prove that the function will attain its minimum value if an exact solution is found for the corresponding problem variant. We also provide proof of the correctness of our approach.

Keywords Combinatorial optimization · Operations research · Non-convex optimization · Kidney exchange · Discrete optimization · Dataless neural networks

1 Introduction

The Kidney Exchange Problem (KEP) can be defined as the challenge of identifying the most optimal kidney exchanges, where optimality is often measured by maximizing the total weight within a pool of donor-patient pairs. To approach this problem, we introduce a directed graph known as the "compatibility graph," denoted as $G = (V = P \cup N, A)$, which includes a weight function w on its arcs, with $w : A \rightarrow \mathbb{R}^+$. In this graph, the set of vertices P represents incompatible pairs of donors and patients, while the set of vertices N represents altruistic donors. Weighted arcs are introduced between vertices u and v if donor u can provide their kidney to patient v, with the weight $w(uv)$ representing the medical benefit of the transplant. As such, a solution to the KEP entails identifying a collection of disjoint walks, which include cycles and chains, each having the maximum possible weight within graph G. These walks must be disjoint because each donor can contribute their kidney only once, and each patient should receive only one kidney. In practical terms, it is essential to limit the size of a cycle, denoted as k, since it involves coordinating $2 \cdot k$ simultaneous surgical operations. Similarly, the size of a chain can also be limited, although there is no universally agreed-upon parameter for this limitation. Therefore, addressing the KEP is essentially equivalent to tackling a Maximum Weighted $\leq k$-cycle and $\leq \ell$-chain Packing Problem. This involves finding a solution that identifies disjoint cycles, each with a size no greater than k, and disjoint chains, each with a size no greater than ℓ, all while maximizing their

✉ K. Subramani
k.subramani@mail.wvu.edu

Sangram K. Jena
jenas2@miamioh.edu

Alvaro Velasquez
alvaro.velasquez@colorado.edu

1 Department of Computer Science and Software Engineering, Miami University, Oxford, OH, USA

2 LDCSEE, West Virginia University, Morgantown, WV, USA

3 Department of Computer Science, University of Colorado Boulder, Boulder, CO, USA

combined weight. Note that the KEP is an optimization problem, and solving the KEP is **NP-hard** for $k \geq 3$ [1, 14].

Optimization problems that are **NP-hard** have traditionally been addressed through the use of approximation algorithms and heuristics, with a heavy emphasis on convex relaxations for many years. Nevertheless, the utilization of non-convex optimization made possible by the incorporation of loss functions in modern neural networks and their accompanying backpropagation training methods introduces new opportunities for more efficiently tackling discrete optimization problems. This innovative approach suggests the potential to replicate the remarkable achievements observed in neural networks, as demonstrated by systems such as Chat-GPT and AlphaGo within the domain of combinatorial optimization. The conventional approach to incorporating neural networks into solving combinatorial optimization problems in the past relied on extensive datasets to extract relevant patterns. However, a more recent methodology has surfaced, one that operates entirely without the need for any data. In a notable study conducted by Alkhouri et al. [3], they ingeniously redefined the Maximum Independent Set problem as a single differentiable function. This formulation enables the utilization of neural networks and backpropagation to address these problems, with only the input instance being required and no additional data. This unconventional approach challenges the standard machine learning paradigm, which assumes the necessity of datasets to fine-tune model parameters, such as neural network weights, to enhance predictive accuracy.

In contrast, dataless neural networks adapt their parameters based on the network's internal structure or other factors independent of external ground truth data. To illustrate this, consider a conventional neural network represented as f with parameters denoted by θ, trained on a dataset comprising pairs $\{(x_i, y_i)\}$. Here, x_i could represent an instance of a discrete optimization problem; while, y_i contains the values of the optimal solutions. The parameters θ are conventionally updated via backpropagation, minimizing a differentiable loss function $L(x_i, f(x_i;\theta))$ to bring the network's output $f(x_i;\theta)$ as close as possible to y_i. Backpropagation adjusts the parameters according to $\theta := \theta - \alpha \cdot \partial L(x_i, f(x_i;\theta))/\partial\theta$, where α controls the learning rate. The concept underlying dataless neural networks challenges the very existence of training data. In this context, the output of the neural network is simplified to $f(e_n;\theta) = f(\theta)$, with e_n representing an all-one vector, essentially serving as a trivial input to the neural network. Consequently, instead of attempting to extract patterns from datasets, dataless neural networks aim to identify optimal solutions for specific discrete optimization problems by imposing a specific structure on f and θ.

The rest of this paper is organized as follows: In Sect. 2, we formally define the problems and the notations considered in the paper. Section 3 describes related work in the literature. In Sect. 4, we design dataless neural networks (dNNs) for the variants of Kidney Exchange. Finally, we conclude in Sect. 5 by summarizing our results and discussing avenues for future work.

2 Statement of problems

In this section, we define the problems and some of the notations considered in this paper.

Definition 1 **Vertex-Disjoint Cycle Cover** (**VDCC**): Given a graph $G = (V, E)$, find a set of disjoint cycles, which are subgraphs of G and contain all vertices of G.

Example ***1*** The set of cycles $C = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}\}$ is a VDCC in the graph in Fig. 1.

We call the VDCC, where the cycle length is at most k the VDk^-CC.

Definition 2 **VD** k^-**CC**: Given a graph $G = (V, E)$ and an integer k, find a set of disjoint cycles of length at most k, which are subgraphs of G and contain all vertices of G.

Example ***2*** For $k = 3$, the set of cycles $C = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}\}$ is a VDk^-CC in the graph in Fig. 1.

VDk^-CC in directed graphs is known as VDk^-CCD.

Definition 3 **VD** k^-**CCD**: Given a directed graph $G = (V, E)$ and an integer k, find a set of disjoint cycles of length at most k, which are subgraphs of G and contain all vertices of G.

Example ***3*** For $k = 3$, the set of cycles $C = \{\{v_1, v_2, v_3\}, \{v_4, v_6, v_5\}\}$ is a VDk^-CCD in the graph in Fig. 2a.

We call the optimization version of VDk^-CCD the VDk^-CCD$_O$.

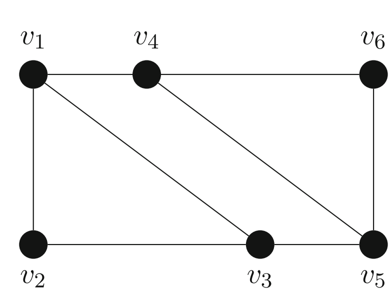


Fig. 1 Example of an instance of the VDCC in a graph G

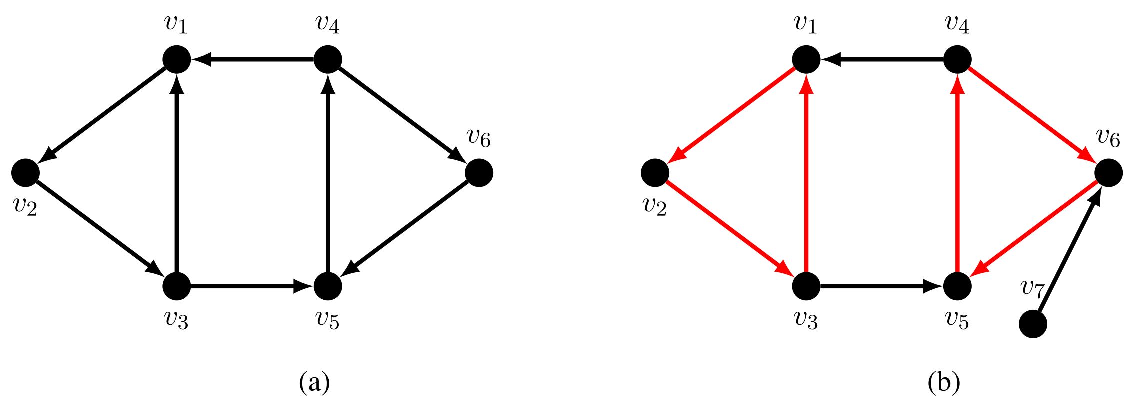


Fig. 2 Instances of VDk^{-}CCD and VDk^{-}CCD$_O$ in a graph G

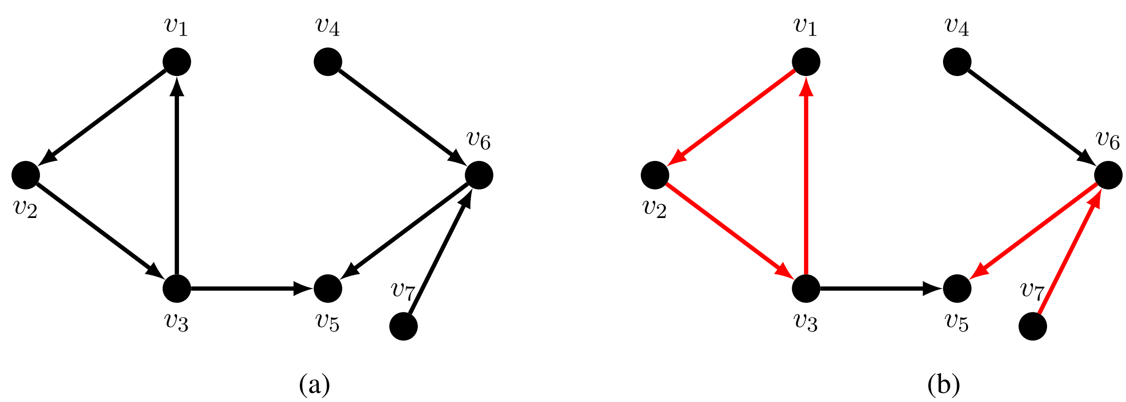


Fig. 3 Example of an instance of the VDk^{-}CCCD$_O$ in a graph G

Definition 4 VD k^{-}CCD$_O$: Given a directed graph $G = (V, E)$, find a set of disjoint cycles of length at most k which are subgraphs of G and contain the maximum number of vertices of G.

***Example* 4** For $k = 3$, the set of cycles $C = \{\{v_1, v_2, v_3\}, \{v_4, v_6, v_5\}\}$ in Fig. 2b covers six out of seven vertices.

Along with the cycles of length at most k, if we allow chains of length at most k to cover the vertices, the problem is known as a variant of the Kidney Exchange problem. We call the problem as VDk^{-}CCCD$_O$.

Definition 5 VD k^{-}CCCD$_O$: Given a directed graph $G = (V, E)$, find a set of disjoint cycles and chains of length at most k, which are subgraphs of G and contain the maximum number of vertices of G.

Example 5 For $k = 3$ in Fig. 3a, if we select a chain $P = \{v_7, v_6, v_5\}$ along with a cycle $C = \{v_1, v_2, v_3\}$ (see Fig. 3b), then they cover six out of seven vertices of the graph.

An activation function in a neural network transforms the summed weighted input from the node into the node's activation or output for that input. In our design of dataless neural networks, we use a rectified linear activation function, also known as the ReLU activation function. It is a piecewise linear function that outputs the input directly if it is positive; otherwise, it outputs zero, i.e., $\sigma(x) = max(0, x)$. For any positive integer n, $[n] := \{1, 2, \ldots, n\}$. Unless mentioned otherwise, $|\cdot|$ represents the absolute value or modulus.

The principal contributions of this paper are as follows:

1. A differential approach for VDk^{-}CCD.
2. A differential approach for VDk^{-}CCD$_O$.
3. A differential approach for VDk^{-}CCCD$_O$.

3 Related work

This section briefly discusses Kidney Exchange and its closely related variants of Cycle Cover. The theoretical underpinnings of kidney exchange were substantially laid by Roth et al. in a series of seminal publications [29–31]. These seminal works delved into the intricate dynamics of efficient pairings within a steady-state kidney exchange.

Building upon this foundation, subsequent investigations by Ashlagi et al. [5], Ashlagi and Roth [6], and Ding et al. [15] addressed critical limitations inherent in these theoretical models, which became apparent as kidney exchange initiatives materialized into practical reality. Game-theoretic models of kidney exchange, which portray transplant centers as agents with private types derived from their internal pools, were introduced and extensively explored by Ashlagi and Roth [6], Toulis and Parkes [34], and Ashlagi et al. [4]. These models shed light on the strategic interactions among stakeholders within the kidney exchange ecosystem.

A VDCC of an undirected graph (if it exists) can be found in polynomial time by transforming the problem into a problem of finding a perfect matching in a larger graph [35]. Finding a VDCC of a directed graph can also be performed in polynomial time by a reduction to perfect matching [9]. It is known that minimal VDCC is **NP-hard** [7]. The problem does not belong to the complexity class **APX** [7]. The problem in directed graphs is also not in **APX**. For the cycle length 3, the vertex-disjoint cycle cover problem in graphs is polynomial-time solvable [35]. The problem is also in **P** when the cycle length is 4 [22]. The complexity status of the problem is not known when the cycle length is fixed to 5. However, David Hartvigsen has some positive results on special cases of this problem [23]. The problem is **NP-complete** when the cycle length is at least 6 [12]. However, in directed graphs, the problem is **NP-complete** when the cycle length is at least 3 [20].

Next, we discuss the state-of-the-art results related to the neural network (NN) and dataless neural network (dNN) available in the literature. Our discussion for NN and dNN is mainly based on many combinatorial optimization problems (COPs). The most interesting COPs are **NP-hard**. It is well-known that such problems do not have polynomial-time efficient algorithms unless some established complexity-theoretic conjectures fail. Although these problems cannot be solved efficiently, they have applications in almost every domain, such as scheduling, routing, telecommunications, planning, transportation, and decision-making processes [8, 17]. Researchers have attempted to address **NP-hard** problems with different efficient, approximate solvers [25]. Broadly, these solvers are categorized into heuristic algorithms [2], approximation algorithms [10], and conventional branch-and-bound methods [32]. Such approaches may produce suboptimal solutions. Some of the other well-studied approaches to dealing with **NP-hard** problems use parameterized [13, 18, 28] and exact exponential algorithmic techniques [19, 21].

Another approach to tackle the Combinatorial Optimization Problems (COPs) involves the utilization of machine learning techniques, as exemplified by studies such as Bengio et al. [8] and Wilder et al. [36]. In the realm of COPs, reinforcement learning has been explored as a means to automate the discovery of heuristics, a concept extensively examined in the works of Drori et al. [16] and Mazyavkina et al. [27]. These models necessitate training based on specific problem instances. They rely on supervised learning, using datasets comprising combinatorial structures of interest sampled from a problem instance distribution. In a pioneering effort, Alkhouri et al. [3] introduced dataless Neural Networks (dNNs) that do not necessitate any data for training. Their approach involves crafting a single differentiable function capable of representing well-known COPs, such as the Maximum Independent Set (MIS) problem. They also devised a similar dNN architecture for addressing the Maximum Clique (MC) and Minimum Vertex Cover (MVC) problems, which are closely linked to the MIS problem. To demonstrate the efficacy of their dNN models in terms of solution quality, they conducted comprehensive experiments on both real-world and synthetic large-scale graphs. In [11], we developed dNNs tailored for solving the Maximum Dissociation Set, k-Coloring, and Maximum Cardinality Distance Matching problems. Additionally, [24] developed the dNNs specifically designed to solve the MAXCUT, MAXkSAT, and MAXNAE2SAT problems.

The existing literature explores various powerful heuristic solvers to address the Maximum Independent Set (MIS) problem. One notable heuristic solver in this context is ReduMIS, as presented by Lamm et al. [25]. ReduMIS comprises two primary components: The first involves an iterative application of graph reduction techniques, while the second employs an evolutionary algorithm. Typically, these approaches entail training neural networks (NNs) using extensive datasets of large graphs for which known solutions exist. In a parallel vein to the dNN method for the MIS problem discussed in [3], another MIS-solving technique was introduced by Schuetz et al. [33]. Notably, the method proposed in [33] dispenses with the requirement for training data, opting instead to utilize a graph neural network. More specifically, its output represents the probability of each node's inclusion in the solution. Unlike the approach presented in [3], Schuetz's method incorporates a loss function to refine its parameterization, capturing the characteristics of the target graph. Furthermore, the Alkhouri et al. [3] approach employs n trainable parameters, where n signifies the number of vertices within the input graph. In contrast, the approach outlined in [33] employs many parameters but confines them to the last layer, utilizing n parameters exclusively in this layer. In the context in [3], the authors presented experimental results and compared them against the most proficient heuristics in the existing literature. They evaluated their success using the solution size obtained by ReduMIS as a benchmark.

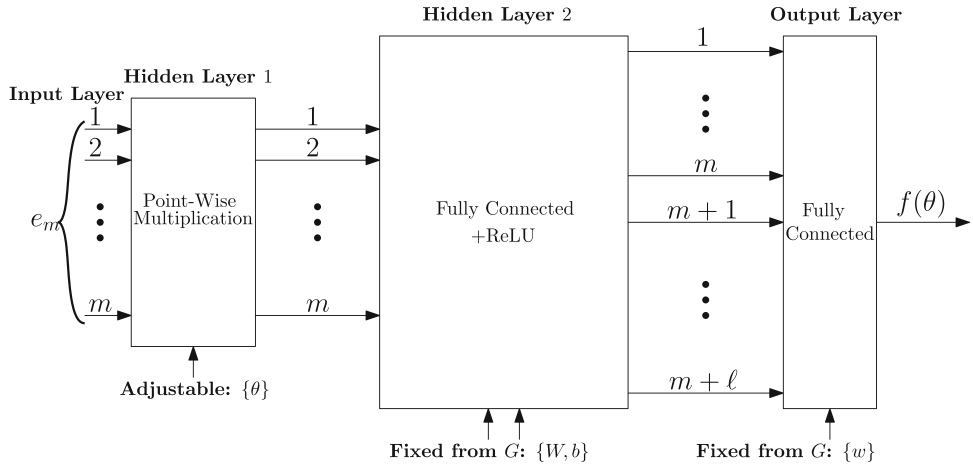
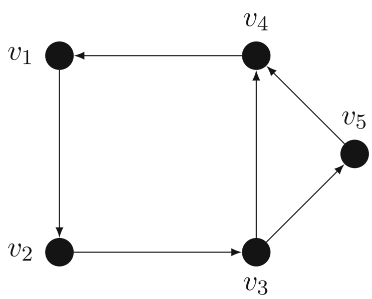


Fig. 4 Block diagram of dNN



(a) A graph G

$$W = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

(b) A binary matrix W

Fig. 5 Representation of a binary matrix W corresponding to G

Moreover, their experimental findings indicated that their approach performed comparably to or surpassed the state-of-the-art learning-based methods detailed in [26].

4 Kidney exchange variants

In this section, we design dataless neural networks for some of the variants of the Kidney Exchange problem.

4.1 VDk^-CCD

This section discusses a dNN for VDk^-CCD. Let $G = (V, E)$ be a directed graph with n vertices over m edges. Let ℓ_1 be the total number of two edges (say (u, v) and (x, v)) toward any vertex v present in G. Let ℓ_2 be the total number of two edges (say (v, w) and (v, y)) from any vertex v present in G. Furthermore, $\ell = \ell_1 + \ell_2$. We construct a dNN f with trainable parameters $\theta \in [0, 1]^m$ with respect to G. That means for each edge $(u, v) \in E$, there is a corresponding trainable parameter θ_{uv} in f. The input to the dNN is an all-one vector e_m, which does not depend upon any data. The output of the dNN is $f(e_m; \theta) = f(\theta) \in \mathbb{R}$. There are four layers in the dNN for VDk^-CCD. The four layers are categorized as one input layer, two hidden layers, and one output layer (see the block diagram in Fig. 4 for the proposed network).

The input layer e_m is connected with the first hidden layer through an element-wise product of the trainable parameters θ. The first hidden layer is connected to the second hidden layer by a binary matrix $W \in \{0, 1\}^{n \times (2 \cdot n)}$. The binary matrix is only dependent on G. At the second hidden layer, there exists a bias vector $b \in \{-1, -1, -k\}^{\ell_1 + \ell_2 + n}$. There is a fully connected weight matrix $w \in \{n, n, m\}^{\ell_1 + \ell_2 + n}$ in the second hidden layer to the output layer. Note that all the parameters are defined as a function of G. The output of f is given as follows:

$$f(e_m; \theta) = f(\theta) = w^T \cdot \sigma((W^T \cdot (e_m \odot \theta)) + b). \qquad (1)$$

Here, $\odot$ denotes the element-wise Hadamard product, which signifies the operation performed by the first hidden layer of the constructed network. The second hidden layer is a fully connected layer composed of a constant matrix denoted as W and a bias vector represented as b, both subjected to a ReLU activation function defined as

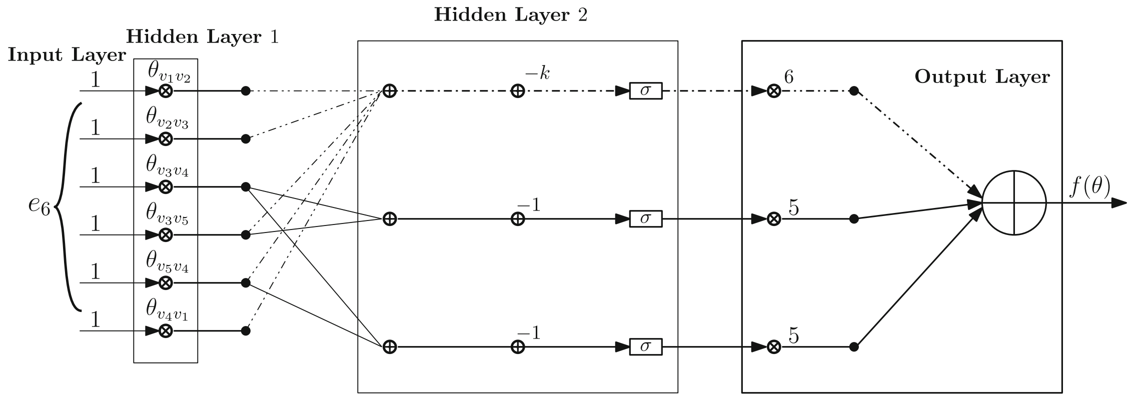


Fig. 6 Construction of dNN f corresponding to the graph in Fig. 5 (a) for VDk^{-}CCD, when $k = 5$ and $C = \{C_1 = \{(v_1, v_2), (v_2, v_3), (v_3, v_5), (v_5, v_4), (v_4, v_1)\}\}$

$\sigma(x) = \max(0, x)$. The final layer of the network is another fully connected layer, and it is described by the vector w.

On the other hand, we prove that when a VDk^{-}CCD solution $C = \{C_1, \cdots, C_p\}$ in G is found, $f(\theta)$ attains its minimum value. Therefore, $f(\theta)$ is an equivalent differentiable function of VDk^{-}CCD solution generated in G. Moreover, C can be constructed from θ as follows. Let $\theta^* = argmin_{\theta\in[0,1]^m} f(\theta)$ be an optimal solution to f. Let $I : [0,1]^m \rightarrow 2^E$ be a VDk^{-}CCD solution corresponding to θ such that $I(\theta) = \{(u,v) \in E \mid \theta^*_{uv} \geq \alpha\}$, for $\alpha > 0$. We show that $|I(\theta^*)| = |C'|$ such that C' consists of edges $(u,v) \in C_i$ for each $C_i \in C$. We choose the edges selected in each $C_i \in C$ in the VDk^{-}CCD solution in G corresponding to the indices of θ whose value exceeds a threshold (say α).

From an input graph $G = (V, E)$, the fixed parameters of f can be constructed as follows: In the binary matrix W, the first $n \times n$ submatrix represents the vertices V of G. Its weights are set equal to the identity matrix I_n (see the 5×5 submatrix in Fig. 5b corresponding to the 5 vertices of G in Fig. 5a). Furthermore, the remaining n columns of W represent the edges of G and for each edge $(u,v) \in E$, the value of $v = 1$ in the column (see the columns v_1 to v_5 in Fig. 5b corresponding to the 6 edges of G in Fig. 5a). For each vertex of G, the corresponding entry of n nodes is $-k$ in the biased vector b. For ℓ_1 number of two edges toward any vertex, the corresponding value in the bias vector is set to -1. Furthermore, for ℓ_2 number of two edges going out from any vertex, the corresponding value in the bias vector is set to -1. Finally, the value of m is assigned in the entries corresponding to the nodes of G in vector w. For ℓ_1 and ℓ_2 entries corresponding to the number of two edges toward any vertex and going out of any vertex v in G, the value is set to n in w. Hence, the parameters W, b, and w are defined as follows:

$$W(i,i) = 1, v_i \in V, i \in [n], \\ W(i, n+j) = 1, \forall (v_i, v_j) \in E, \tag{2}$$

$$b(i) = -k, w(i) = m, v_i \in V, i \in [n], \\ b(n+s_1) = -1, w(n+s_1) = n, s_1 \in [\ell_2] \\ b(n+s_1+s_2) = -1, w(n+s_1+s_2) = n, s_2 \in [\ell_1]. \tag{3}$$

So, the function in (1) can be rewritten as follows:

$$f(\theta) = n \cdot \sum_{\substack{(u,v)\in E \\ (x,v)\in E}} \sigma(\theta_{uv} + \theta_{xv} - 1) + n \cdot \sum_{\substack{(v,w)\in E \\ (v,y)\in E}} \sigma\big(\theta_{vw} + \theta_{vy} - 1\big) + m \cdot \sum_{\substack{v\in V \\ \theta_i \geq 1}} \sigma\big(\theta_{vx} + \theta_{xy} + \cdots + \theta_{zv} - k\big) \tag{4}$$

An example of the above-discussed dNN construction is presented in Fig. 6.

The following theorem establishes the relation between a VDk^{-}CCD solution and the minimum value of f in the constructed dNN with respect to a given graph G.

Theorem 1 *Let $G = (V, E)$ be a directed graph and its corresponding dNN be f. For $k \geq 3$, G has a VDk^{-}CCD solution $C = \{C_1, \cdots, C_p\}$, if and only if the minimum value of f is* 0.

Proof Let $C = \{C_1, \cdots, C_p\}$ be a VDk^{-}CCD solution in G. For each $(u,v) \in E$, set the value of θ_{uv} as follows: For each $C_i \in C$, if $(u,v) \in C_i$, then set $\theta_{uv} = 1$. Otherwise, set

$\theta_{uv} = 0$. Consider the output f for an arbitrary cycle $C_i \in C$ and any set of two edges toward and going from a common vertex. As per the definition of VDk^-CCD, each cycle is of length at most k. So, the part of the function $m \cdot \sum_{\substack{v \in V \\ \theta_i \geq 1}} \sigma(\theta_{vx} + \theta_{xy} + \cdots + \theta_{zv} - k)$ will always be 0. Moreover, each cycle $C_i \in C$ is disjoint. So, there are at most two edges associated with any vertex (one edge toward the vertex and another edge from the vertex) that can be 1. Thus, the other two parts of the function $n \cdot \sum_{\substack{(u,v) \in E \\ (x,v) \in E}} \sigma(\theta_{uv} + \theta_{xv} - 1)$ and $n \cdot \sum_{\substack{(v,w) \in E \\ (v,y) \in E}} \sigma(\theta_{vw} + \theta_{vy} - 1)$ are always 0. Hence, the minimum value of $f(\theta) = 0$.

Conversely, assume that the minimum value of the output function f is $f(\theta) = 0$. We construct a VDk^-CCD solution C in G from f as follows: From the construction of the dNN, it is clear that, for each set of two edges (u, v) and (x, v) toward a common vertex v, $\theta_{uv} + \theta_{xv} \leq 1$. Furthermore, for each set of two edges (v, w) and (v, y) from a common vertex v, $\theta_{vw} + \theta_{vy} \leq 1$. Otherwise, f does not achieve its minimum value. To prove this, assume that $\theta_{uv} + \theta_{xv} > 1$ and/or $\theta_{vw} + \theta_{vy} > 1$. It follows that the function contributes $n \cdot (\theta_{uv} + \theta_{xv} - 1) > 0$ and/or $n \cdot (\theta_{vw} + \theta_{vy} - 1) > 0$ to the output $f(\theta)$. This is a contradiction to the fact that f achieves its minimum value. We can simply assign the value of θ_{uv} or θ_{xv} and/or θ_{vw} or θ_{vy} as zero and reduce the value of f further. So, it is clear that for any two edges (u, v) and (x, v) toward v and two edges (v, w) and (v, y) going out of v in G, the θ value of at most one of the two such edges can be 1. Each such edge, which θ value is one, is a part of a cycle C_i as each vertex is a part of a cycle. Furthermore, the length of any such cycle C_i formed by using edges of θ value 1 is at most k. Otherwise, the function contributes $m \cdot \sigma(\theta_{vx} + \theta_{xy} + \cdots + \theta_{zv} - k) > 0$ to the output $f(\theta)$. For each entry of θ with value 1, consider the corresponding edge and put the cycle C_i in the VDk^-CCD solution C in which the edge is a part. It is clear that C is a VDk^-CCD solution in G. □

4.2 VDk^-CCD$_O$

In this section, we discuss a dNN for the optimization version of VDk^-CCD, which we call VDk^-CCD$_O$. Let $G = (V, E)$ be a directed graph with n vertices over m edges. Let ℓ_1 be the total number of two edges (say (u, v) and (x, v)) toward any vertex v present in G. Let ℓ_2 be the total number of two edges (say (v, w) and (v, y)) going out of any vertex v present in G. Furthermore, $\ell = \ell_1 + \ell_2$. We construct a dNN f with trainable parameters $\theta \in [0,1]^{n+m}$ with respect to G. That means for each vertex $v \in V$, there is a corresponding trainable parameter θ_v and for each edge $(u, v) \in E$, there is a corresponding trainable parameter θ_{uv} in f. The input to the dNN is an all-one vector e_{n+m}, which does not depend upon any data. The output of the dNN is $f(e_{n+m}; \theta) = f(\theta) \in \mathbb{R}$. There are four layers in the dNN for VDk^-CCD$_O$. The four layers are categorized as one input layer, two hidden layers, and one output layer (see the block diagram in Fig. 4 for the proposed network).

The input layer e_{n+m} is connected with the first hidden layer through an element-wise product of the trainable parameters θ. The first hidden layer is connected to the second hidden layer by the binary matrix $W \in \{0,1\}^{n \times (2 \cdot n)}$. The binary matrix is only dependent on G. At the second hidden layer, there exists a bias vector $b \in \{-\frac{3}{4}, -1, -1, -k\}^{n+\ell_1+\ell_2+n}$. There is a fully connected weight matrix $w \in \{-1, n, n, m\}^{n+\ell_1+\ell_2+n}$ in the second hidden layer to the output layer. Note that all the parameters are defined as a function of G. The output of f is given as follows:

$$f(\theta) = -\sum_{v \in V} \sigma(\theta_v - \frac{3}{4}) + n \cdot \sum_{\substack{(u,v) \in E \\ (x,v) \in E}} \sigma(\theta_{uv} + \theta_{xv} - 1) + n \cdot \sum_{\substack{(v,w) \in E \\ (v,y) \in E}} \sigma(\theta_{vw} + \theta_{vy} - 1) + m \cdot \sum_{\substack{v \in V : \theta_v = 1 \\ \theta_i \geq 1}} \sigma(\theta_{vx} + \theta_{xy} + \cdots + \theta_{zv} - k) \tag{5}$$

On the other hand, we prove that when a VDk^-CCD$_O$ solution $C = \{C_1, \cdots, C_p\}$ in G is found, $f(\theta)$ attains its minimum value. Therefore, $f(\theta)$ is an equivalent differentiable function of VDk^-CCD$_O$ solution generated in G. Moreover, C can be constructed from θ as follows. Let $\theta^* = argmin_{\theta \in [0,1]^{n+m}} f(\theta)$ be an optimal solution to f. Let $I : [0,1]^m \rightarrow 2^E$ be a VDk^-CCD$_O$ solution corresponding to θ such that $I(\theta) = \{(u,v) \in E \mid \theta^*_{uv} \geq \alpha\}$, for $\alpha > 0$. We show that $|I(\theta^*)| = |C'|$ such that C' consists of edges $(u, v) \in C_i$ for each $C_i \in C$. We choose the edges selected in each $C_i \in C$ in the VDk^-CCD$_O$ solution in G corresponding to the indices of θ whose value exceeds a threshold (say α). From an input graph $G = (V, E)$, the fixed parameters of f can be constructed as follows: In the binary matrix W, the first $n \times n$ submatrix represents the vertices V of G. Its weights are set equal to the identity matrix I_n. Furthermore, the remaining n columns of W represent the edges of G and for each edge $(u, v) \in E$, the value of $v = 1$ in the column (see Fig. 5). For each vertex of G, the corresponding entry of n nodes is $-k$ in the biased vector b. For ℓ_1 number of two edges toward any vertex, the corresponding value in the bias vector is set to -1. For ℓ_2 number of two edges going out of any vertex, the corresponding value in the bias vector is set to -1. Furthermore, for each vertex of G, the corresponding entry of n nodes is $-\frac{3}{4}$ in the biased vector b. Finally, the value of m is assigned in the entries corresponding to the nodes of G in vector w. For ℓ_1 and ℓ_2 entries corresponding to the number of two edges toward any vertex and going out of any vertex v in G, the value is set to n in w. For each vertex of G, the corresponding entry of n nodes is -1 in the vector w.

The following theorem establishes the relation between a VDk^-CCD$_O$ solution and the minimum value of f in the constructed dNN with respect to a given graph G.

Lemma 1 *Let $G = (V, E)$ be a directed graph having n vertices over m edges and its corresponding dNN be f. For $k \geq 3$, G has a VDk^-CCD$_O$ solution $C = \{C_1, \cdots, C_p\}$ which covers η vertices of G, if and only if the minimum value of f is $-\frac{\eta}{4}$.*

Proof Let $C = \{C_1, \cdots, C_p\}$ be a VDk^-CCD$_O$ solution in G, which covers η vertices in G. For each $(u, v) \in E$, set the value of θ_{uv} as follows: For each $C_i \in C$, if $(u, v) \in C_i$, then set $\theta_{uv} = 1$. Otherwise, set $\theta_{uv} = 0$. Moreover, for each vertex $v \in V$ which is a part of the cycle cover C, we have $\theta_v = 1$; otherwise, $\theta_v = 0$. Consider the output f for an arbitrary cycle $C_i \in C$ and any set of two edges toward and going out of a common vertex. As per the definition of VDk^-CCD$_O$, each cycle is of length at most k. So, the part of the function $m \cdot \sum_{\substack{v \in V \\ \theta_i \geq 1}} \sigma(\theta_{vx} + \theta_{xy} + \cdots + \theta_{zv} - k)$ will always be 0. Moreover, each cycle $C_i \in C$ is disjoint. So, there are at most two edges associated with any vertex (one edge toward the vertex and another edge going out of the vertex) that can be 1. Thus, the other two parts of the function $n \cdot \sum_{\substack{(u,v) \in E \\ (x,v) \in E}} \sigma(\theta_{uv} + \theta_{xv} - 1)$ and $n \cdot \sum_{\substack{(v,w) \in E \\ (v,y) \in E}} \sigma(\theta_{vw} + \theta_{vy} - 1)$ are always 0. Furthermore, there are η vertices covered in the cycle cover C. So, for each vertex in the cycle cover, $f(\theta)$ contributes $-\frac{1}{4}$. Therefore, the minimum value of $f(\theta) = -\frac{\eta}{4}$.

Conversely, assume that the minimum value of the output function f is $f(\theta) = -\frac{\eta}{4}$. Then, it is clear that the θ value of η number of vertices is 1. We construct a VDk^-CCD$_O$ solution C in G from f as follows: From the construction of the dNN, it is clear that, for each set of two edges (u, v) and (x, v) toward a common vertex v, $\theta_{uv} + \theta_{xv} \leq 1$. Furthermore, for each set of two edges (v, w) and (v, y) going out of a common vertex v, $\theta_{vw} + \theta_{vy} \leq 1$. Otherwise, f does not achieve its minimum value. To prove this, assume that $\theta_{uv} + \theta_{xv} > 1$ and/or $\theta_{vw} + \theta_{vy} > 1$. It follows that the function contributes $n \cdot (\theta_{uv} + \theta_{xv} - 1) > 0$ and/or $n \cdot (\theta_{vw} + \theta_{vy} - 1) > 0$ to the output $f(\theta)$. This is a contradiction to the fact that f achieves its minimum value. We can simply assign the value of θ_{uv} or θ_{xv} and/or θ_{vw} or θ_{vy} as zero and reduce the value of f further. So, it is clear that for any two edges (u, v) and (x, v) toward v and two edges (v, w) and (v, y) going out of v in G, the θ value of at most one of the two such edges can be 1. Each such edge, which θ value is one, is a part of a cycle C_i. Furthermore, the length of any such cycle C_i formed by using edges of θ value 1 is at most k. Otherwise, the function contributes $m \cdot \sigma(\theta_{vx} + \theta_{xy} + \cdots + \theta_{zv} - k) > 0$ to the output $f(\theta)$. For each entry of θ with value 1, consider the corresponding edge and put the cycle C_i in the VDk^-CCD$_O$ solution C in which the edge is a part. It is clear that C is a VDk^-CCD$_O$ solution, which covers η number of vertices in G. □

4.3 VDk^-CCCD$_O$

In this section, we discuss a dNN for a variant of Kidney Exchange called VDk^-CCCD$_O$. Let $G = (V, E)$ be a directed graph with n vertices over m edges. Let ℓ_1 be the total number of two edges (say (u, v) and (x, v)) toward any vertex v present in G. Let ℓ_2 be the total number of two edges (say (v, w) and (v, y)) going out of any vertex v present in G. Furthermore, $\ell = \ell_1 + \ell_2$. We construct a dNN f with trainable parameters $\theta \in [0, 1]^{n+m}$ with respect to G. That means for each vertex $v \in V$, there is a corresponding trainable parameter θ_v and for each edge $(u, v) \in E$, there is a corresponding trainable parameter θ_{uv} in f. The input to the dNN is an all-one vector e_{n+m}, which does not depend upon any data. The output of the dNN is $f(e_{n+m}; \theta) = f(\theta) \in \mathbb{R}$. There are four layers in the dNN for VDk^-CCCD$_O$. The four layers are categorized as one input layer, two hidden layers, and one output layer (see the block diagram in Fig. 4 for the proposed network).

The input layer e_{n+m} is connected with the first hidden layer through an element-wise product of the trainable parameters θ. The first hidden layer is connected to the second hidden layer by the binary matrix $W \in \{0, 1\}^{n \times (2 \cdot n)}$. The binary matrix is only dependent on G. At the second hidden layer, there exists a bias vector $b \in \{-\frac{3}{4}, -1, -1, -k, -k\}^{n+\ell_1+\ell_2+n+n}$. There is a fully connected weight matrix $w \in \{-1, n, n, m, m\}^{n+\ell_1+\ell_2+n+n}$ in the second hidden layer to the output layer. Note that all the parameters are defined as a function of G. The output of f is given as follows:

$$f(\theta)=-\sum_{v\in V}\sigma(\theta_v-\frac{3}{4})+n\cdot\sum_{\substack{(u,v)\in E\\(x,v)\in E}}\sigma(\theta_{uv}+\theta_{xv}-1)$$

$$+n\cdot\sum_{\substack{(v,w)\in E\\(v,y)\in E}}\sigma(\theta_{vw}+\theta_{vy}-1)$$

$$+m\cdot\sum_{\substack{u\in V:\theta_u=1\\\theta_i\geq 1}}\sigma(\theta_{uv}+\cdots+\theta_{zu}-k)+m\cdot$$

$$\sum_{\substack{u\in V:\theta_u=1\\\theta_i\geq 1}}\sigma(\theta_{uv}+\cdots+\theta_{yz}-k) \tag{6}$$

On the other hand, we prove that when a VDk^{-}CCCD$_O$ solution C in G is found, $f(\theta)$ attains its minimum value. Therefore, $f(\theta)$ is an equivalent differentiable function of VDk^{-}CCCD$_O$ solution generated in G. Moreover, C can be constructed from θ as follows. Let $\theta^*=argmin_{\theta\in[0,1]^{n+m}}f(\theta)$ be an optimal solution to f. Let $I:[0,1]^m\rightarrow 2^E$ be a VDk^{-}CCCD$_O$ solution corresponding to θ such that $I(\theta)=\{(u,v)\in E\mid\theta^*_{uv}\geq\alpha\}$, for $\alpha>0$. We show that $|I(\theta^*)|=|C'|$ such that C' consists of edges $(u,v)\in C_i$ for each $C_i\in C$. We choose the edges selected in each $C_i\in C$ in the VDk^{-}CCCD$_O$ solution in G corresponding to the indices of θ whose value exceeds a threshold (say α). From an input graph $G=(V,E)$, the fixed parameters of f can be constructed as follows: In the binary matrix W, the first $n\times n$ submatrix represents the vertices V of G. Its weights are set equal to the identity matrix I_n. Furthermore, the remaining n columns of W represent the edges of G and for each edge $(u,v)\in E$, the value of $v=1$ in the column (see Fig. 5). For each vertex of G, the corresponding entry of n nodes is $-k$ in both the parts of the biased vector b. For ℓ_1 number of two edges toward any vertex, the corresponding value in the bias vector is set to -1. For ℓ_2 number of two edges going out from any vertex, the corresponding value in the bias vector is set to -1. Furthermore, for each vertex of G, the corresponding entry of n nodes is $-\frac{3}{4}$ in the biased vector b. Finally, the value of m is assigned in the entries corresponding to the nodes of G in both the parts of the vector w. For ℓ_1 and ℓ_2 entries corresponding to the number of two edges toward any vertex and going out of any vertex v in G, the value is set to n in w. For each vertex of G, the corresponding entry of n nodes is -1 in the vector w.

The following theorem establishes the relation between a VDk^{-}CCCD$_O$ solution and the minimum value of f in the constructed dNN with respect to a given graph G.

Lemma 2 *Let $G=(V,E)$ be a directed graph having n vertices over m edges and its corresponding dNN be f. For $k\geq 3$, G has a VDk^{-}CCCD$_O$ solution $C=\{C_1,\cdots,C_p\}$ which covers η vertices of G, if and only if the minimum value of f is $-\frac{\eta}{4}$.*

Proof Let $C=\{C_1,\cdots,C_p\}$ be a VDk^{-}CCCD$_O$ solution in G, which covers η vertices in G. For each $(u,v)\in E$, set the value of θ_{uv} as follows: For each $C_i\in C$, if $(u,v)\in C_i$, then set $\theta_{uv}=1$. Otherwise, set $\theta_{uv}=0$. Moreover, for each vertex $v\in V$, which is a part of the cycle or chain cover C, we have $\theta_v=1$; otherwise, $\theta_v=0$. Consider the output f for an arbitrary cycle or chain $C_i\in C$ and any set of two edges toward and going out of a common vertex. As per the definition of VDk^{-}CCCD$_O$, each cycle is of length at most k. So, the part of the function $m\cdot\sum_{\substack{u\in V:\theta_u=1\\\theta_i\geq 1}}\sigma(\theta_{uv}+\cdots+\theta_{zu}-k)$ will always be 0. Moreover, each chain is of length at most k. So, the part of the function $m\cdot\sum_{\substack{u\in V:\theta_u=1\\\theta_i\geq 1}}\sigma(\theta_{uv}+\cdots+\theta_{yz}-k)$ will always be 0. Furthermore, each cycle/chain $C_i\in C$ is disjoint. So, at most, two edges are associated with any vertex in the cover (one edge toward the vertex and another edge from the vertex) that can be 1. Thus, the other two parts of the function $n\cdot\sum_{\substack{(u,v)\in E\\(x,v)\in E}}\sigma(\theta_{uv}+\theta_{xv}-1)$ and $n\cdot\sum_{\substack{(v,w)\in E\\(v,y)\in E}}\sigma(\theta_{vw}+\theta_{vy}-1)$ are always 0. Furthermore, there are η vertices covered in the cycle cover C. So, for each vertex in the cycle cover, $f(\theta)$ contributes $-\frac{1}{4}$. Therefore, the minimum value of $f(\theta)=-\frac{\eta}{4}$.

Conversely, assume that the minimum value of the output function f is $f(\theta)=-\frac{\eta}{4}$. Then, it is clear that the θ value of η vertices is 1. We construct a VDk^{-}CCCD$_O$ solution C in G from f as follows: From the construction of the dNN, it is clear that, for each set of two edges (u,v) and (x,v) toward a common vertex v, $\theta_{uv}+\theta_{xv}\leq 1$. Furthermore, for each set of two edges (v,w) and (v,y) going out of a common vertex v, $\theta_{vw}+\theta_{vy}\leq 1$. Otherwise, f does not achieve its minimum value. To prove this, assume that $\theta_{uv}+\theta_{xv}>1$ and/or $\theta_{vw}+\theta_{vy}>1$. It follows that the function contributes $n \cdot (\theta_{uv} + \theta_{xv} - 1) > 0$ and/or $n \cdot (\theta_{vw} + \theta_{vy} - 1) > 0$ to the output $f(\theta)$. This is a contradiction to the fact that f achieves its minimum value. We can simply assign the value of θ_{uv} or θ_{xv} and/or θ_{vw} or θ_{vy} as zero and reduce the value of f further. So, it is clear that for any two edges (u, v) and (x, v) toward v and two edges (v, w) and (v, y) going out of v in G, the θ value of at most one of the two such edges can be 1. Each such edge, which θ value is one, is a part of a cycle/chain C_i. Furthermore, the length of any such cycle/chain C_i formed by using edges of θ value 1 is at most k. Otherwise, the function contributes $m \cdot \sigma(\theta_{uv} + \cdots + \theta_{zu} - k) > 0$ or $m \cdot \sigma(\theta_{uv} + \cdots + \theta_{yz} - k) > 0$ to the output $f(\theta)$. For each entry of θ with value 1, consider the corresponding edge and put the cycle/chain C_i in the VDk^-CCCD$_O$ solution C in which the edge is a part. It is clear that C is a VDk^-CCCD$_O$ solution in G. □

5 Conclusion

In this paper, we have explored various aspects of the Kidney Exchange problem as vertex-disjoint cycle cover variants, specifically VDk^-CCD, VDk^-CCD$_O$, and VDk^-CCCD$_O$. We devised differentiable functions tailored to these problem instances and established their correctness by demonstrating that these functions reach their minimum value when an exact solution is found. The derivation of these differentiable functions is based on the framework of dataless neural networks. For future work, we aim to conduct implementation-based experiments to test the effectiveness of our proposed approach, employing a diverse range of dataless neural networks. A crucial research direction involves devising a transformation procedure to convert arbitrary integer programs into dataless neural networks.

Funding This research was supported in part by the Defense Advanced Research Projects Agency through Grant HR001123S0001-FP-004.

Data availability We have no data associated with this research.

Declarations

Conflict of interest We declare that we have no Conflict of interest.

References

1. Abraham DJ, Blum A, Sandholm T (2007) Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In: Proceedings of the 8th ACM conference on Electronic commerce, pp 295–304
2. Akiba T, Iwata Y (2016) Branch-and-reduce exponential/FPT algorithms in practice: a case study of vertex cover. Theoret Comput Sci 609:211–225
3. Alkhouri Ismail R, Atia George K, Alvaro V (2022) A differentiable approach to the maximum independent set problem using dataless neural networks. Neural Netw 155:168–176
4. Ashlagi I, Fischer F, Kash IA, Procaccia AD (2015) Mix and match: a strategyproof mechanism for multi-hospital kidney exchange. Games Econom Behav 91:284–296
5. Ashlagi I, Gamarnik D, Rees MA, Roth AE (2012) The need for (long) chains in kidney exchange. Technical report, National Bureau of Economic Research
6. Ashlagi I, Roth AE (2014) Free riding and participation in large scale, multi-hospital kidney exchange. Theor Econ 9(3):817–863
7. Giorgio A, Pierluigi C, Giorgio G, Viggo K, Alberto M-S, Marco P (2012) Complexity and approximation: combinatorial optimization problems and their approximability properties. Springer, Cham
8. Bengio Y, Lodi A, Prouvost A (2021) Machine learning for combinatorial optimization: a methodological tour d'horizon. Eur J Oper Res 290(2):405–421
9. Blum A (2013) Algorithms: recitation notes. https://www.cs.cmu.edu/avrim/451f13/recitation/rec1016.txt
10. Boppana R, Halldórsson MM (1992) Approximating maximum independent sets by excluding subgraphs. BIT Numer Math 32(2):180–196
11. Jena SK, Subramani K, Velasquez A (2023) Differentiable discrete optimization using dataless neural networks. In: Weili W, Jianxiong G, (eds.), Combinatorial Optimization and Applications - 17th International Conference, COCOA 2023, Hawaii, HI, USA, December 15-17, 2023, Proceedings, Part II, vol 14462 of *Lecture Notes in Computer Science*, pp 3–15. Springer
12. Cornuejols G, Pulleyblank W (1980) A matching problem with side conditions. Discret Math 29(2):135–159
13. Cygan M, Fomin FV, Kowalik L, Lokshtanov D (2015) Dániel Marx. In: Michal P, Saket S (eds) Parameterized algorithms. Springer, Cham
14. Dickerson JP, Manlove DF, Plaut B, Sandholm T, Trimble J (2016) Position-indexed formulations for kidney exchange. In: Proceedings of the 2016 ACM Conference on Economics and Computation, pp 25–42
15. Ding Y, Ge D, He S, Ryan CT (2015) A non-asymptotic approach to analyzing kidney exchange graphs. In: Proceedings of the Sixteenth ACM Conference on Economics and Computation, pp 257–258
16. Drori I, Kharkar A, Sickinger WR, Kates B, Ma Q, Ge S, Dolev E, Dietrich B, Williamson DP, Udell M (2020) Learning to solve combinatorial optimization problems on real-world graphs in linear time. In: 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), pp 19–24
17. Festa P (2014) A brief introduction to exact, approximation, and heuristic algorithms for solving hard combinatorial optimization problems. In 2014 16th International Conference on Transparent Optical Networks (ICTON), pp 1–20
18. Flum Jörg GM (2006) Parameterized complexity theory. Texts in theoretical computer science. An EATCS series. Springer, Cham
19. Fomin FV, Kratsch D (2010) Exact exponential algorithms. Texts in theoretical computer science. An EATCS series. Springer, Cham
20. Garey MR, Johnson DS (1991) Computers and intractability: a guide to the theory of np-completeness. W. H. Freeman Company, San Francisco
21. Gaspers S (2010) Exponential time algorithms-structures, measures, and bounds. VDM

22. Hartvigsen D (1984) Extensions of matching theory. PhD thesis, Carnegie Mellon University
23. Hartvigsen D (1999) The square-free 2-factor problem in bipartite graphs. In: International Conference on Integer Programming and Combinatorial Optimization, pp 234–241. Springer
24. Jena SK, Subramani K, Velasquez A (2024) A differential approach for several NP-hard optimization problems. In: International Workshop on Combinatorial Image Analysis, pp 68–80. Springer
25. Lamm S, Sanders P, Schulz C, Strash D, Werneck RF (2016) Finding near-optimal independent sets at scale. In: 2016 Proceedings of the Eighteenth Workshop on Algorithm Engineering and Experiments (ALENEX), pp 138–150
26. Li Z, Chen Q, Koltun V (2018) Combinatorial optimization with graph convolutional networks and guided tree search. In: Advances in neural information processing systems, p 31
27. Mazyavkina N, Sviridov S, Ivanov S, Burnaev E (2021) Reinforcement learning for combinatorial optimization: A survey. Comput Oper Res 134:105400
28. Niedermeier R (2006) Invitation to fixed-parameter algorithms. Oxford University Press, Oxford
29. Roth AE, Sönmez T, Ünver MU (2004) Kidney exchange. Quart J Econ 119(2):457–488
30. Roth AE, Tayfun S, Utku ÜM (2005) Pairwise kidney exchange. J Econ Theor 125(2):151–188
31. Roth AE, Sönmez T, Utku Ünver M (2007) Efficient kidney exchange: coincidence of wants in markets with compatibility-based preferences. Am Econ Rev 97(3):828–851
32. Segundo PS, Rodríguez-Losada D, Jiménez A (2011) An exact bit-parallel algorithm for the maximum clique problem. Comput Oper Res 38(2):571–581
33. Schuetz Martin JA, Kyle Brubaker J, Katzgraber Helmut G (2022) Combinatorial optimization with physics-inspired graph neural networks. Nat Mach Intell 4(4):367–377
34. Toulis P, Parkes DC (2015) Design and analysis of multi-hospital kidney exchange mechanisms using random graphs. Games Econom Behav 91:360–382
35. William TT (1954) A short proof of the factor theorem for finite graphs. Can J Math 6:347–352
36. Wilder B, Dilkina B, Tambe M (2019) Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 33, pp 1658–1665