



HAL
open science

Maximum Expected Value Partial Failure Path Problem

Noam Goldberg, Michael Poss

► **To cite this version:**

Noam Goldberg, Michael Poss. Maximum Expected Value Partial Failure Path Problem. 2020. hal-02964433v2

HAL Id: hal-02964433

<https://hal.science/hal-02964433v2>

Preprint submitted on 13 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MIP Formulations for the Maximum Expected Value Partially Successful Path Problem

Noam Goldberg*

Michael Poss†

Abstract

Mathematical programming formulations are developed for a problem motivated by determining chains of organ-donation exchange pairs in a compatibility graph. The objective is to maximize the expected value where pairs are known to fail with given probabilities. Historically this NP-hard problem has been solved heuristically or exactly only for limited path lengths. Although the problem appears highly nonlinear, we formulate it as a mixed-integer linear program (MILP). A computationally tractable layered formulation that approximately solves larger instances is also presented.

Keywords: Longest path, layered formulation, kidney exchange, barter exchange, clearing algorithms

1 Introduction

Consider a setting in which each of m items has an associated profit and probability of success. In this paper we consider a maximum expected value partially successful (s, t) -path problem (PSPP), where each (s, v) -subpath of an (s, t) -path, up to some vertex v along the path, has a value of the product of probabilities of arcs along the (s, v) -subpath, times the sum of arc profits.

Our problem bears similarities with the probabilistic all-or-nothing problem, which is to select a subset of items so as to maximize the expected total profit – the product of probabilities of selected items times the sum of profits of these items. In particular, the all-or-nothing probabilistic path problem [13], has the subsets of items corresponding to paths in a graph, each of which has an expected value that is equal to the product of its underlying arc probabilities times the sum of its arc profits. Other all-or-nothing problems have been considered over all unrestricted subsets of a given ground set, and matchings in a graph [14].

While all-or-nothing path problems have applications in designing series-connected subsystems, among others, the partial failure variant studied in the current paper has important applications in live-donor organ exchange problems. In particular, paired-kidney donation graphs indicate donor-recipient compatibility using directed arcs. Each vertex in such a graph corresponds to a pair of a recipient (patient) and donor – typically a family member who is willing to donate a kidney in exchange for the related patient receiving one [1]. Because exchanges occur simultaneously, there is significant uncertainty with respect to the success of a particular donation to take place as a part of the exchange. For example, a particular donor may not be available on a given day due to illness or simply because reneging on the agreement. This setting with uncertainty gives rise to partially successful paths and the objective considered in this paper; see also the expected utility of chains formulated in [8] and the future work discussion in [16].

The idea of performing kidney-transplant exchanges in chains or paths of donor-recipient pairs, initiated by an altruistic donor, has been applied in practice and described in [18]. The overall kidney-exchange problem described in [6], [12], [17], [1] and [8] is a market clearing problem where all (recipient) vertices must be covered by paths or cycles. Cycles are initiated

*Department of Management, Bar-Ilan University, Ramat Gan 5290002, Israel noam.goldberg@biu.ac.il

†LIRMM, University of Montpellier, CNRS, France. michael.poss@lirmm.fr

by paired donors and paths are initiated by yet-to-be-paired altruistic donor vertices (vertices that have only outgoing arcs). The problem of determining a maximum expected-value PPSP can be considered as the pricing problem of the overall market clearing problem. Also, PPSP can be considered independently; as soon as an altruistic donor arrives he/she can be used to initiate a path of maximum expected value. The current paper is focused on improving and developing new mixed-integer programming formulations for this problem.

Related to our problem is the orienteering problem: an orienteering path problem is to maximize the number or weight of vertices visited subject to a time limit on all visits [10, 4, 5]. In particular a discounted orienteering problem is a special case with $p_e = P$ for all $e \in E$ and some $0 < P < 1$ (the problem is introduced in [4] for undirected graphs).

Next we formally describe the problem and relate our mixed-integer nonlinear programming (MINLP) formulation to the optimization problem as it appears in the literature. We then propose a linearization and reformulation of the MINLP, followed by an extended formulation based on a layered graph. Finally, we conclude with computational experiments on simulated kidney exchange data to evaluate the proposed formulations along with attempts to strengthen them.

2 Problem definition

We now formally state the PPSP problem. Let $G = (V, E)$ be a directed graph, $s, t \in V$ be the source and the sink, and let $\delta^+(u) = \{v \in V : (u, v) \in E\}$ and $\delta^-(u) = \{v \in V : (v, u) \in E\}$ be sets of direct successors and predecessors of node $u \in V$, respectively. The problem with multiple sources/sinks is also handled by constructing a supersource s and a supersink t that are connected to the multiple sources and sinks, respectively. For each edge $e = (u, v) \in E$, we are given a positive profit $c_e = c_{uv} \geq 0$ and a probability of success $p_e = p_{uv} \in [0, 1]$. Let $m = |E|$ and $n = |V|$. The maximum expected value PPSP problem is to find a path $\pi \in \mathcal{P}(t)$, where $\mathcal{P}(t) \subseteq 2^E$ is the set of all *elementary* paths from s to t , so that the objective function

$$z(\pi) = \sum_{(u,v) \in \pi} c_{uv} \prod_{f \in \pi(v)} p_f \quad (1)$$

is maximized. First, the following proposition relates the simplified form of the objective (3) to the objective function considered in for determining kidney exchange paths initiated by altruistic donors in [8].

Proposition 1. *The objective*

$$\sum_{(u,v) \in \pi} (1 - p_{uv}) \sum_{e \in \pi(u)} c_e \prod_{f \in \pi(u)} p_f + \sum_{e \in \pi} c_e \prod_{f \in \pi} p_f, \quad (2)$$

considered in [8] is equivalent to (1).

Proof. We rewrite (2) using straightforward algebra

$$\begin{aligned} (1) &= - \sum_{(u,v) \in \pi} p_{uv} \sum_{e \in \pi(u)} c_e \prod_{f \in \pi(u)} p_f + \sum_{(u,v) \in \pi} \sum_{e \in \pi(u)} c_e \prod_{f \in \pi(u)} p_f + \sum_{e \in \pi} c_e \prod_{f \in \pi} p_f \\ &= - \sum_{(u,v) \in \pi} \sum_{e \in \pi(u)} c_e \prod_{f \in \pi(v)} p_f + \sum_{(u,v) \in \pi} \sum_{e \in \pi(v)} c_e \prod_{f \in \pi(v)} p_f \\ &= \sum_{(u,v) \in \pi} c_{uv} \prod_{f \in \pi(v)} p_f \\ &= z(\pi) \end{aligned} \quad \square$$

Then, the maximum expected value PPSP problem is

$$\max\{z(\pi) : \pi \in \mathcal{P}(t)\}. \quad (3)$$

Evidently the problem (3) with $p_e = 1$ for all $e \in E$ is a longest path problem that is strongly NP-hard in general directed graphs [11].

3 MILP formulations

We now present mixed-integer linear programming (MILP) formulations to exactly and approximately solve (3).

3.1 Exact formulation

Although the problem (3) appears to be highly nonlinear in fact it can be formulated as a MILP. Given a graph $G = (V, E)$, with $s, t \in V$, let the path flow polytope be denoted by

$$\Pi = \left\{ x \in \mathbb{R}_+^{|E|} \mid \begin{array}{l} \sum_{u \in \delta^-(v)} x_{uv} = \sum_{u \in \delta^+(v)} x_{vu} \text{ for } v \in V \setminus \{s, t\} \\ \sum_{u \in \delta^+(s)} x_{su} = \sum_{u \in \delta^-(t)} x_{ut} = 1 \end{array} \right\}.$$

and consider the following MILP formulation.

$$\max \sum_{(u,v) \in E} c_{uv} q_{uv} \quad (4a)$$

$$\text{subject to: } \sum_{w \in \delta^-(u)} q_{wu} = \sum_{v \in \delta^+(u)} \frac{1}{p_{uv}} q_{uv} \quad u \in V \setminus \{s, t\} \quad (4b)$$

$$q_{uv} \leq M_{uv} x_{uv} \quad (u, v) \in E \quad (4c)$$

$$x \in \Pi \quad (u, v) \in E \quad (4d)$$

$$x_{uv} \in \{0, 1\}, q_{uv} \geq 0 \quad (u, v) \in E, \quad (4e)$$

where $0 < M_{uv} \leq 1$ is a sufficiently large probability constant. For each $(u, v) \in E$ the decision variable x_{uv} is used to indicate whether (u, v) is selected as a path edge, and the auxiliary variables q_{uv} is used to define $q_{uv} = \prod_{(u,v) \in \pi} p_{uv}$ as in $z(\pi)$. The constraints corresponding to (4d) are the $s - t$ path-flow conservation constraints. Constraint (4c) enforces that $q_{uv} = 0$ for each $(u, v) \in E$ with $x_{uv} = 0$. Finally, (4b) implements the recursion to accumulate the probability product for each pair of selected incident edges (w, u) and (u, v) with $x_{uw} = x_{vu} = 1$, it requires that $q_{wu} p_{uv} = q_{uv}$.

To strengthen the formulation (4) it is desirable to select M_{uv} for each $(u, v) \in E$ as small as possible. It can be observed it suffices to set $M_{uv} = \bar{p}(u) p_{uv}$ where $\bar{p}(v)$ denotes the maximum probability of a path from s to v for all $v \in S \setminus \{s\}$ and $\bar{p}(s) = 1$. It is known that \bar{p} can be determined in polynomial time by solving a maximum reliability problem (assuming exact arithmetic it is equivalent to a shortest path problem).

To prevent solutions that are feasible for (4) from containing cycles, cycle-breaking inequalities must be added. We consider two alternative ways of breaking the cycles. First, following [20], we consider *generalized cutset inequalities*,

$$\sum_{(u,v) \in \delta^+(S)} x_{uv} \geq \sum_{(w,v) \in \delta^+(w)} x_{wv} \quad S \subseteq V \setminus \{s, t\} : |S| \geq 2, w \in S. \quad (5)$$

These constraints break any cycle by enforcing that the number of arcs leaving S be not smaller than the number of arcs outgoing from any node k of S .

A second type of inequalities will prevent cycles if the probabilities are strictly less than unity (specifically their product along cycles). The sufficiency of the inequalities

$$\sum_{v \in \delta^+(u)} x_{uv} \leq 1 \quad u \in V \setminus \{t\}, \quad (6)$$

for prevention of cycles, given the condition that success is uncertain along cycles, is established by the following proposition.

Proposition 2. *The problem (4) with (6) has an acyclic optimal solution if $\prod_{f \in c} p_f < 1$ for each cycle $c \subseteq E$.*

Proof. Consider (x, q) that is a feasible solution of (4) with (6). Note that by the flow-conservation constraints and the integrality of x , the support of x is a union of one path π from s to t and a set of cycles $C \subset 2^E$. Suppose that there is some $u \in V$ that is incident to edges in two different cycles or an edge in a cycle and an edge in π . Then, by the integrality of x , $\sum_{v \in \delta^+(u)} x_{uv} > 1$, thereby violating (6). It follows that the cycles $c \in C$ and the path π are all vertex disjoint. Now further suppose for the sake of deriving a contradiction that (x, q) is an optimal solution with minimal $|C| > 0$ and consider a cycle $c \in C$. If $q_e = 0$ for each $e \in c$, then a new solution (x', q) can be defined (using (x, q)) by setting $x'_e = 0$ for each $e \in c$. The objective value of (x', q) is equal to that of (x, q) and has one less cycle, thereby establishing a contradiction. Otherwise, if $q_e > 0$ for some $e \in c$, constraints (4b) written for all $(u, v) \in c$ imply that $\prod_{f \in c} p_f = 1$. \square

3.2 Rounded formulation

The formulation from the previous section tends to have a weak linear programming relaxation, due the big- M coefficients involved in constraints (4c). We present in this section an alternative and stronger approach that is based on layered graphs, where multiple copies of graph vertices and edges are created and “stacked on top of each other”; see for example [15]. Specifically, we consider next a base $0 < \alpha < 1$ and define the rounded logarithms of the probabilities as $r_e = \lceil \log_\alpha p_e \rceil$, and accordingly, using an upper bound on r_e for all $e \in E$ of $U = n \max_{e \in E} r_e$, the set of rounded logarithms values is $R = \{0, \dots, U\}$. Define $V_0 = V \setminus \{s, t\}$ and let its induced subset of edges be denoted by $E_0 = E[V_0]$. Then, consider copies of V_0 , one for each for each $k \in K$, to be denoted by $V_0^k = V_0 \times K$; each member $(v, k) \in V^k$ is the k -th copy of $v \in V_0$ in \widehat{V} . The layered graph is then $\widehat{G} = (\widehat{V}, \widehat{E})$, where $\widehat{V} = \{(s, 0), (t, 0)\} \cup \bigcup_{k \in K} V_0^k$ and

$$\begin{aligned} \widehat{E} = & \{(u, k, v, k + r_{u,v}) \mid (u, v) \in E_0, k \in \{0, \dots, U - r_{u,v}\}\} \\ & \cup \{(s, 0, v, 0) \mid v \in \delta^+(s)\} \cup \{(u, k, t, 0) \mid u \in \delta^-(t), k \in \{0, \dots, U\}\}. \end{aligned}$$

Let $\widehat{\Pi}$ be the corresponding path flow polytope Π defined for the layered graph \widehat{G} in place of G , with $(s, 0)$ and $(t, 0)$ as the source and the sink nodes. Evidently, all paths in this graph correspond to extreme points of this polytope. Consider the formulation

$$\max \quad \sum_{e=(u,k,v,\bar{k}) \in \widehat{E}} \alpha^k c_{uv} x_e \quad (7a)$$

$$\text{subject to:} \quad \sum_{\substack{v \in V \setminus \{s\}: \exists \bar{k} \in K \\ e=(u,k,v,\bar{k}) \in \widehat{E}}} x_e \leq 1 \quad u \in V \setminus \{t\} \quad (7b)$$

$$x \in \widehat{\Pi} \quad (7c)$$

$$x_e \in \{0, 1\} \quad e \in \widehat{E}. \quad (7d)$$

Note that while any path $\widehat{\pi} = ((u_1, q_1) = (s_1, q_1), (u_2, q_2), \dots, (u_l, r_l) = (t_l, r_l))$ in \widehat{G} is elementary, its projection onto V , $\pi = (u_1 = s, u_2, \dots, u_l = t)$ might contain a cycle. We forbid such paths $\widehat{\pi}$ (whose projection contain one or more cycles) by limiting the number of outgoing arcs from any $u \in V \setminus \{s\}$ to at most one in constraints (7b). Also note that the rounding makes solving formulation (7) an inexact solution of (7). However, if α is chosen close enough to one, the worst-case performance of the optimal solution of (7) can be bounded from the theoretical viewpoint as formalized below.

Observation 1. *For $\alpha \leq (1 - \epsilon)^{1/n}$, then each solution that is optimal to (7) is an $(1 - \epsilon)$ -approximate solution to (4) with (6).*

4 Computational results

4.1 Kidney exchange graph experiments

In this subsection we experiment with kidney compatibility graphs that are randomly generated following the structure and data given in kidney exchange literature.

Graph topology simulated according to [3] (graph[n]a[A] datasets). We generate a “sparse-dense” graph with the node connectivity generated according to the panel reactive antibodies (PRA) levels as suggested in [3], and we make use of additional data given in [2, 8]. In particular we generate graph vertices with a highly sensitized recipient with probability 0.27 these recipients are connected with donors (corresponding to an incident incoming edge) with probability 0.03. The failure probabilities are assigned also based on PRA levels and highly sensitized patients are matched with a success probability of 0.5. Otherwise, low PRA recipients are compatible with donors with probability 0.5, 57% of them are matched with a success probability of 0.75 (cross-match failure rate of 0.25) and 43% of them are matched with a success probability of 0.95 (cross-match failure rate of 0.05). The values of matching PRA recipients are generated from a normal distribution with mean 10 and standard deviation 2. The values of matching highly sensitized recipients are a factor of 1.5 greater (so the mean is 15) as suggested in [8].

Graphs based on simulated donor data instances of [7] and [8] (MD[n] datasets). These instances were generated using a simulator based on [19]. Graph arcs were constructed based on blood-type compatibility and we set the weights to be either unitary in datasets MD[n]unitval or normally distributed with mean 10 and standard deviation 2 in datasets MD[n]-stochval. For patients that are considered highly sensitized with PRA levels exceeding 0.73 in the original data, the arc values considered were twice as much as initially generated. The success probabilities are set to 0.5 for highly sensitized patients (determined by the PRA levels). Otherwise, a success probability of 0.75 is set with probability 0.57 and success probability of 0.95 is set with probability 0.43 (similar to the graph[n]a[A] datasets).

4.2 Implementation details

All algorithms have been coded in Julia 1.3.1 on a platform using a (Intel i7-10510U) 1.80GHz CPU with 4 cores and an 8MB cache and 16GB of RAM memory. The mathematical programs are modeled using the JuMP v0.21.1 [9] and Gurobi v0.7.6 packages and solved using with the Gurobi 9.0.1 solver running 4 threads.

None of our instances contain an edge with a probability equal to 1. Although, if an instance has multiple altruistic donors, we then connect supersources and supersinks to all nodes with edges having probability 1, but this may not lead to cycles c such that $\prod_{f \in c} p_f = 1$. Hence, the condition of Proposition 2 holds so inequalities (6) are enough to guarantee the existence of optimal acyclic solutions. This being said, inequalities (5) are not dominated by inequalities (6), so separation of the former through callbacks might be further improve the overall performance of the formulation.

Following [20], we thus tested the separation of inequalities (5) at both fractional and integer solutions by checking whether they are satisfied on each strongly connected component of the graph induced by the positive components of x that is optimal to (a relaxation of) (4). Our results indicated a worsening of the overall performance: the relaxation improvement, often nonexistent, is too little to compensate for the time spent in the separation procedure. Therefore, we do not use inequalities (5) in the results presented next.

4.3 Description of the results

Table 1 shows for the particular kidney exchange graph instances that we experimented, statistics and characteristics including the number of nodes, edges, number of altruistic donors, length of the longest path from the supersource to any node in the graph, and the root relaxation gap (in %). Solution times are reported in seconds in Tables 2 and 3. Here T indicates that a time limit of 1800 seconds is reached and M indicates that the solver has exhausted the available memory. Several observations can be made from the results of these tables. First, we see that the extended formulation (7) scales reasonably well, as long as the memory limit is not exceeded. Some of the large MD instances, namely 84unitval, 84stochval, 125unitval, are solved faster than

Instance	$ V $	$ E $	$ S $	LP	Root gap
MD12stochval	18	96	1	9	9
MD12unitval	18	96	1	9	10
MD19stochval	18	150	1	14	7
MD19unitval	18	150	1	14	4
MD21stochval	18	114	2	10	7
MD21unitval	18	114	2	10	1
MD23stochval	18	121	2	13	17
MD23unitval	18	121	2	13	28
MD43stochval	34	402	1	19	14
MD43unitval	34	402	1	19	13
MD44stochval	34	337	1	17	15
MD44unitval	34	337	1	17	11
MD51stochval	34	405	3	19	6
MD51unitval	34	405	3	19	16
MD60stochval	34	371	3	19	10
MD60unitval	34	371	3	19	9
MD61stochval	34	422	4	19	16
MD61unitval	34	422	4	19	17
MD66stochval	34	403	4	22	20
MD66unitval	34	403	4	22	16
MD70stochval	34	332	4	18	13
MD70unitval	34	332	4	18	18
MD81stochval	66	2205	3	55	15
MD81unitval	66	2205	3	55	20
MD84stochval	66	1512	3	36	29
MD84unitval	66	1512	3	36	27
MD125stochval	130	6026	6	73	7
MD125unitval	130	6026	6	73	12

Instance	$ V $	$ E $	$ S $	LP	Root gap
graph20a1	22	178	1	20	25
graph30a2	32	329	2	30	31
graph40a2	42	681	2	41	30
graph50a3	52	981	3	51	23
graph60a3	62	1387	3	61	17
graph70a4	72	1851	4	71	13
graph80a4	82	2247	4	81	10

Table 1: Kidney exchange graph instance main characteristics. LP stands for longest path and the root gaps are in % with respect to formulation (4).

formulation (4). Second, the exact formulation (4) appears to be more efficiently solved with instances MD than with instances “graph”. To further investigate this behavior we refer to the graph instance characteristics in Table 1. The table data shows that graph30a2 and instances MD43 – MD70 have comparable numbers of nodes and edges. However, graph30a2 cannot be solved in 1800 seconds, while these MD instances are all solved within a few seconds, except instance 66unitval which requires roughly 72 seconds. A partial explanation of this behavior could lie in the strong root relaxation gap of (4) for these MD instances, ranging from 6 to 20% while for the graph30a2 instance it equals 31%. The strength of the relaxation appears to correlate with the length of paths in these graphs (directly impacting M_{uv}); for the MD instances with 34 nodes the length ranges from 17 to 22 hops, compared with 30 hops in graph30a2.

Next, the third group of gap columns of Tables 2 and 3 provide the best solution found by the extended formulation (7), relatively to the best solution found by the exact formulation (4) and expressed in %. All instances solved exactly by formulation (4) can only lead to negative values. Evidently, $\alpha < 0.8$ seems to lead to poor solutions, while setting α higher than 0.8 appears to yield only marginal improvements. Finally, we remark that for the instance MD125unitval that cannot be solved to optimality, the layered formulation (7) even provides slightly better solutions than the best solution found by the exact formulation (within the 3 hour time limit).

5 Conclusions and future work

In this paper we proposed a linear formulation for the PPSP problem. For this formulation we prove and evaluate cycle prevention inequalities that outperform standard longest path cycle prevention inequalities. We then evaluated a layered formulation that effectively solved the

Instance	α	CPU Seconds						Gap (%)				
		0.75	0.8	0.85	0.9	0.95	(4)	0.75	0.8	0.85	0.9	0.95
graph20a1		3	6	7	10	16	11	31	4	4	1	0
graph30a2		14	32	40	79	25	T	44	3	1	0	0
graph40a2		119	345	635	115	608	T	55	1	0	-1	-1
graph50a3		1198	128	407	590	114	T	49	0	-1	-1	-1
graph60a3		233	386	534	1456	216	T	71	0	-1	-1	-1
graph70a4		T	1521	1784	729	662	T	72	0	0	-1	-1
graph80a4		T	T	T	492	T	T	74	1	20	0	-

Table 2: Solution CPU seconds and and optimality gap for approximate MILP (7) for the “graph” instances.

Instance	α	CPU Seconds						Gap (%)				
		0.75	0.8	0.85	0.9	0.95	(4)	0.75	0.8	0.85	0.9	0.95
MD12unitval		1	1	1	1	4	1	9	0	0	0	0
MD12stochval		1	1	1	1	4	1	7	0	0	0	0
MD19unitval		1	1	1	2	6	1	27	0	0	0	0
MD19stochval		1	1	1	3	18	1	26	0	0	0	0
MD21unitval		0	1	1	2	6	1	1	0	1	0	0
MD21stochval		1	1	1	2	4	1	24	0	0	0	0
MD23unitval		1	1	1	2	3	1	46	0	0	0	0
MD23stochval		1	1	1	1	2	1	37	2	3	2	0
MD43unitval		8	28	43	57	37	2	43	0	0	0	0
MD43stochval		10	27	37	54	53	2	44	10	0	0	0
MD44unitval		4	10	13	17	25	3	62	0	0	0	0
MD44stochval		4	11	13	17	13	1	52	1	1	1	0
MD51unitval		15	38	49	107	46	2	37	0	0	0	0
MD51stochval		11	39	57	114	73	1	30	0	0	0	0
MD60unitval		8	26	34	81	55	2	48	0	0	0	0
MD60stochval		10	21	31	59	70	1	34	2	2	0	0
MD61unitval		15	33	36	72	54	1	54	0	0	0	0
MD61stochval		11	29	35	74	44	2	42	0	0	0	0
MD66unitval		12	30	38	72	39	73	24	0	0	0	0
MD66stochval		18	64	121	170	62	2	59	3	3	0	0
MD70unitval		7	19	28	60	27	7	58	0	0	0	0
MD70stochval		6	21	37	58	45	1	50	3	2	2	0
MD81unitval		564	731	210	207	557	T	77	0	0	0	0
MD81stochval		T	T	T	T	T	T	63	100	27	8	1
MD84unitval		367	1469	1524	T	T	T	54	1	0	1	1
MD84stochval		332	603	715	521	T	T	49	7	5	1	0
MD125unitval		409	1200	1343	M	M	T	64	-1	-1	-	-
MD125stochval		M	M	M	M	M	T	-	-	-	-	-

Table 3: Solution CPU seconds and optimality gap for approximate MILP (7) for the MD instances.

problem within a guaranteed approximation gap for larger instances. According to our experiments the approximation gap in practice usually appeared much smaller than the theoretical guarantee.

The problem considered in the current paper is strongly NP-hard in general graphs. Kidney-exchange and other organ donation compatibility graphs do not appear to have much of a special structure. In contrast, computing a path that maximizes the similar yet different all-or-nothing objective remains NP-hard in directed acyclic graphs (DAGs); see [13]. Although it is not difficult to extend the FPTAS suggested for the all-or-nothing objectives to the current problem, it is a question of theoretical interest for future work to determine whether the current problem remains NP-hard in a DAG.

References

- [1] R. Anderson, I. Ashlagi, D. Gamarnik, and A. E. Roth. Finding long chains in kidney exchange using the traveling salesman problem. *Proceedings of the National Academy of Sciences*, 112(3):663–668, 2015.
- [2] I. Ashlagi, D. Gamarnik, M. A. Rees, and A. E. Roth. The need for (long) chains in kidney exchange. Technical report, National Bureau of Economic Research, 2012.
- [3] I. Ashlagi, D. S. Gilchrist, A. E. Roth, and M. A. Rees. Nonsimultaneous chains and dominos in kidney-paired donation-revisited. *American Journal of Transplantation*, 11:984–994, 2011.
- [4] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff. Approximation algorithms for orienteering and discounted-reward tsp. *SIAM Journal on Computing*, 37(2):653–670, 2007.
- [5] C. Chekuri, N. Korula, and M. Pál. Improved algorithms for orienteering and related problems. *ACM Transactions on Algorithms (TALG)*, 8(3):23, 2012.
- [6] Y. Chen, Y. Li, J. D. Kalbfleisch, Y. Zhou, A. Leichtman, and P. X.-K. Song. Graph-based optimization algorithm and software on kidney exchanges. *IEEE Transactions on Biomedical Engineering*, 59(7):1985–1991, 2012.
- [7] J. P. Dickerson, A. D. Procaccia, and T. Sandholm. Optimizing kidney exchange with transplant chains: Theory and reality. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 711–718, 2012.
- [8] J. P. Dickerson, A. D. Procaccia, and T. Sandholm. Failure-aware kidney exchange. *Management Science*, 65(4):323–340, 2019.
- [9] I. Dunning, J. Huchette, and M. Lubin. Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.
- [10] M. Fischetti, J. J. S. González, and P. Toth. Solving the orienteering problem through branch-and-cut. 10:133–148, 1998.
- [11] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [12] K. M. Glorie, J. J. van de Klundert, and A. P. M. Wagelmans. Kidney exchange with long chains: An efficient pricing algorithm for clearing barter exchanges with branch-and-price. *Manufacturing & Service Operations Management*, 16:498–512, 2014.
- [13] N. Goldberg and M. Poss. Maximum probabilistic all-or-nothing paths. *European Journal of Operational Research*, 283:279–289, 2020.
- [14] N. Goldberg and G. Rudolf. On the complexity and approximation of the maximum expected value all-or-nothing subset. *Discrete Applied Mathematics*, 283:1–10, 2020.
- [15] L. E. N. Gouveia, M. Leitner, and M. Ruthmair. Layered graph approaches for combinatorial optimization problems. *Comput. Oper. Res.*, 102:22–38, 2019.
- [16] Y. Li, P. X.-K. Song, Y. Zhou, A. B. Leichtman, M. A. Rees, and J. D. Kalbfleisch. Optimal decisions for organ exchanges in a kidney paired donation program. 6:85–104, 2014.
- [17] D. F. Manlove and G. O’Malley. Paired and altruistic kidney donation in the UK: algorithms and experimentation. *ACM J. Exp. Algorithmics*, 19(1), 2014.
- [18] M. A. Rees, J. E. Kopke, R. P. Pelletier, D. L. Segev, M. E. Rutter, A. J. Fabrega, J. Rogers, O. G. Pankewycz, J. Hiller, A. E. Roth, T. Sandholm, M. U. Ünver, and R. A. Montgomery. A nonsimultaneous, extended, altruistic-donor chain. *The New England Journal of Medicine*, 360:1096–1101, 2009.
- [19] S. L. Saidman, A. E. Roth, T. Sönmez, M. U. Ünver, and F. L. Delmonico. Increasing the opportunity of live kidney donation by matching for two-and three-way exchanges. *Transplantation*, 81(5):773–782, 2006.
- [20] L. Taccari. Integer programming formulations for the elementary shortest path problem. *European Journal of Operational Research*, 252(1):122–130, 2016.