A polyhedral study of the cardinality constrained multi-cycle and multi-chain problem on directed graphs

Vicky Mak-Haua,1,*

School of Information Technology, Deakin University, Waurn Ponds, Geelong, Vic 3216, Australia.

Abstract

In this paper, we study the Cardinality Constrained Multi-cycle Problem (CCMcP) and the Cardinality Constrained Cycle and Chain Problem (CCCCP), both arose from kidney exchange or barter exchange optimisation. The CCMcP is defined on a directed graph, with a feasible solution allowing more than one cycle to exist on the digraph, not all vertices must be involved in the cycles, a vertex can be involved in no more than one cycle, and the cardinality of the cycles are restricted. The CCCCCP is an extension of the CCMcP, that it also allows multiple cardinality-constrained chains, with an additional set of vertices that can only serve, and are the only vertices that can serve, as the start of a chain.

This paper focuses on the polyhedral study of the arc-based formulations for the CCMcP and the CCCCCP, with the aim of filling the literature gap in polyhedral analysis of combinatorial optimization problems defined on directed or undirected graphs that concern cardinality-constrained or unconstrained single- or multi-cycle and/or chain problems. We prove that three classes of non-trivial constraints are facet-defining for the CCMcP polytope, propose four new classes of constraints and prove their validity. We then prove that the non-negativity constraints and the bound constraints are facet-defining for the CCCCCP polytope. We also experiment with two sets of simulated data, carry out preliminary numerical tests by using these data instances to demonstrate the strength of the constraints we have found, solve these instances to optimality, and discuss future research directions.

Keywords: integer programming, combinatorial optimisation, polyhedral analysis, cardinality constrained cycle and chains, kidney exchange, clearing barter exchange

1. Introduction

We first provide a mathematical description of the two combinatorial optimization problems under study in this paper. Consider a digraph $D = (V, A)$ with $V$ the set of vertices and $A$ the
set of arcs. Each arc \( a \in A \) is associated with a weight \( w_a \). A feasible solution to the Cardinality Constrained Multi-cycle Problem (CCMcP) may contain arcs forming more than one cycle, with each cycle involving a distinct set of vertices, and that there may be vertices not involved in any cycles. A solution that contains no cycles at all is also considered feasible. A \( k \)-cycle is a single-cycle that involves \( k \) vertices. The cycles are constrained in size, with cardinality not exceeding \( K \), for \( 2 \leq K \leq |V| \). The optimal solution to the CCMcP, however, is one that maximizes the total weight of arcs involved in all cycles. In the case when all arc weights are “1”, the objective function is then equivalent to maximizing the total number of arcs used. In comparison to the well-known Asymmetric Travelling Salesman Problem (ATSP), the two main differences are: in an ATSP, the Assignment Problem (AP) relaxation also allows multiple cycles (subtours), but these cycles are not constrained in size, and that all vertices must be visited.

Now consider again \( D = (V,A) \), where the set of vertices \( V \) is partitioned into: \( V = N \cup P \) and \( N \cap P = \emptyset \), and the set of arcs is given by: \( A = \{ (i,j) \mid i,j \in P, i \neq j \} \cup \{ (i,j) \mid i \in N, j \in P \} \). A feasible solution to the Cardinality Constrained Cycle and Chain Problem (CCCCCP), like that of a CCMcP, allows one or more cardinality constrained cycles, though a cycle can only involve arcs in the set \( \{ (i,j) \mid i,j \in P, i \neq j \} \). However, a CCCCP also allows one or more chains, where the first arc in a chain must be from the set \( \{ (i,j) \mid i \in N, j \in P \} \), but subsequent arcs from the set \( \{ (i,j) \mid i,j \in P, i \neq j \} \). The length of the chains is constrained to be no more than \( L \) vertices. An \( \ell \)-chain is a chain that involves \( \ell \) vertices. A vertex in \( P \) cannot be in more than one cycle or chain, and a vertex in \( N \) cannot be in more than one chain. In the literature, \( L = K \) is considered in e.g., Manlove and O’Malley [2]; \( L > K \) is considered in, e.g., Glorie et al. [3], Dickerson et al. [4] and Plaut et al. [5]; and \( L = \infty \) in, e.g., Anderson et al. [1]. Again, an optimal solution to the CCCCP is one that maximizes the total weight of arcs in all chains and cycles.

The CCMcP and the CCCCP have been studied in the context of Kidney Exchange Optimization (which is under the general umbrella of barter exchange) both in terms of mathematical modelling and solution methodologies. A review will be provided in Section 1.2. The contribution of this paper, however, is focused on the theoretical analysis of the arc-based formulations of the CCMcP and CCCCP, with the aim of enriching the literature of polyhedral analyses of arc-based formulations of constrained or unconstrained single- or multi-cycle problems defined on directed or undirected graphs. In Section 1.3, we review polyhedral results on a number of closely related combinatorial optimization problems. To the best of our knowledge, there has not been any polyhedral study of the CCMcP or the CCCCP, except for Mak-Hau [6].

### 1.1 The Kidney Exchange Problem

The Kidney Exchange-family of problems (KEPs) have attracted the attention of the combinatorial optimisation community. The KEP can be represented on a directed graph, with vertices representing the incompatible donor-patient pairs (PDPs)—by an incompatible pair, we mean a donor-patient pair (usually family or friends) such that the patient cannot accept the kidney of the donor due to ABO blood type incompatibility or positive serological cross match. A kidney exchange pool contains a large number of such PDPs. If the kidney of the donor in Pair A is a match for the patient in Pair B, then it can be represented as an arc from vertex A to vertex B on the digraph, and if an arc also exists in the opposite direction, then an exchange of kidneys can...
be carried out between PDPs A and B. Such an exchange is called a 2-cycle (a cycle involving 2 PDPs, hence 2 vertices on the digraph). An exchange can also involve more than two vertices. A 3-cycle may have the kidney of the donor in Pair A be donated to the patient in Pair B, that of the donor in Pair B donated to the patient in Pair C, and that of the donor in Pair C donated to the patient in Pair A. Naturally, a \( k \)-cycle will be a single cycle that involves \( K \) pairs of PDPs (and with no sub-cycles involved). As donors are not legally bounded to donate a kidney, in order to avoid donors quitting the program as soon as their partners receiving their kidneys, the exchanges involved in a \( k \)-cycle have to be performed simultaneously, hence the cardinality limit, \((K)\), of these kidney exchange cycles cannot be too large. In the context of kidney exchange, 2- and 3-cycles are very common, although the largest exchange cycle performed involved nine PDPs (see, e.g., SF Gate [7]). In a kidney exchange solution, multiple exchange cycles are involved, hence the underlying combinatorial optimisation problem is in fact a CCMcP.

In recent years, some kidney exchange pools have altruistic donors involved, a kidney exchange sequence that begins from an altruistic donor, who donates his/her kidney to a patient in a PDP, with the donor of this PDP in turn donating his/her kidney to the patient in a another PDP and so on, will eventually terminate at the deceased donor waiting list. Such a sequence of kidney exchanges is called a chain. The value \( L \) in a CCCCCP is called the cap size in the context of KEP. On a directed graph, we will use the set \( N \) to denote the set of altruistic donors, and \( P \) to denote the set of PDPs. A kidney donate chain that begins from an altruistic donor will form a chain on the directed graph. As both chains and cycles are expected to exist as a solution to a kidney exchange optimisation problem, the underlying combinatorial optimisation problem is a CCCCCP.

The objective of a kidney exchange optimisation problem is to maximise either the number of kidney exchanges, or a weighted sum of some metrics of the exchanges, and by weight, we mean a “score” for each transplant based on some prioritisation scheme. This means that in the underlying combinatorial optimisation problem, the objective function is to either maximise the total number of arcs used in the cycles (and chains) or to maximise the total weighted arc costs.

1.2 State-of-the-art solution methods

Integer programming models developed for the CCMcP and the CCCCCP, (which are mostly developed in the context of KEP), can be classified into three main branches: (a) arc-based models with a small number of variables, but exponentially many constraints (see, e.g., Roth et al. [8] for CCMcP and Mak-Hau [9] for CCCCCP); (b) cycle-based models with a small number of constraints, but exponentially many variables (see, e.g., Abraham et al. [10] and Roth et al. [8] for CCMcP and Anderson et al. [1] for CCCCCP); and (c) arc-based compact models that creates multiple clones of the directed graph, making it possible to have both variables and constraints be polynomial in size (see, e.g., Constantino et al. [11] and Dickerson et al. [4]). In terms of solution methodologies, branch-and-price based methods are applied with implementation details presented in, e.g., Abraham et al. [10], Glorie et al. [3], Klimentova et al. [12], Plaut et al. [5]. A summary review of the performances of Abraham et al. [10], Manlove and O’Malley [2], Constantino et al. [11], Glorie et al. [3], Klimentova et al. [12], Anderson et al. [1], and Mak-Hau [9] can be found in Mak-Hau [9], together with details on the sizes of problem tackled, and values \( K \) and \( L \) tested.

Recently, Dickerson et al. [4] presented a new polynomial size formulations for the CCMcP and
the CCCC with bounded $L$ wherein a binary variable is used to determine whether an arc is used in a particular position of a cycle in a particular copy of the digraph—a concept that is an extension to the extended edge formulation proposed by Constantino et al. [11] and with a stronger LP relaxation (LPR) bound. A polynomial size algorithm for variable elimination in preprocessing was discussed and implemented. The position-indexed edge formulation (PIEF) is developed for the CCMcP. For the CCCC, a similar idea is used for the chain variables, though for the cycles, a binary variable for each cycle is used. When $K$ is small, these cycles can be completely enumerated, but a pricing algorithm was proposed for a branch-and-price version of the model, called PICEF. Finally, a hybrid version is proposed that uses the position based binary variables for both cycles and chains (HPIEF). The authors extensively tested their methods on a number of large-scale problem instances, with $K = 3$ and $L$ varying from 2 to 12, and sizes as large as, e.g.: $|P| = 700$ with $|N| = 35$, and $|P| = 500$ with $|N| = 125$. From the experiments, it appears that PICEF and HPIEF outperformed all other methods tested. The paper also provided a proof that the LPR of PIEF is as strong as the cycle-formulation (see Abraham et al. [10] and Roth et al. [8]).

The models/methods listed above are mainly for single-objective implementation, multi-objective approaches has been discussed in Glorie et al. [3], and implemented in Manlove and O’Malley [2] and Manlove and O’Malley [13]. The method of Manlove and O’Malley [2] and Manlove and O’Malley [13] promote the use of shorter cycles e.g., 2-cycles as well as 3-cycles with a back arc (such that if one arc is broken, i.e., one transplant cannot move forward, the remaining two PDPs can still perform a kidney exchange). The reason is to reduce the chance of an exchange chain/cycle be broken due to unexpected events. The authors proposed a lexicographical approach to deal with multiple objectives one by one. The order of priorities are: 1. The number of 2-cycles be maximised. 2. The number of matches be maximised. 3. The number of back-arcs in the 3-cycles be maximised. 4. The overall arc weights be maximised. On the other hand, failure-awareness has also been addressed in, e.g., the model of Dickerson et al. [14] takes into consideration arcs probabilistically fail after algorithm match but before transplantation in modelling the weights on arcs. Heuristic approaches for multi-criteria objective function for the CCCC is also proposed in Mak-Hau and Nickhols [15]. It appears that as of this date, the state-of-the-art algorithms for the CCCC for $L = \infty$ is Anderson et al. [1] and that for bounded $L$ is Dickerson et al. [4].

### 1.3 A brief review of literature in polyhedral studies of related cycle and chain problems

In this section, we review the literature of polyhedral analysis of arc-based formulations of constrained or unconstrained single- or multi-cycle problems defined on directed or undirected graphs. First we provide background mathematics and explain why polyhedral study is of value.

The strength of the constraints in an integer program is one crucial aspect in the efficiency of an exact algorithm, and by a strong constraint (or, a strong cut), we mean one that cuts off as much infeasible fractional points as possible. A valid constraint is one that the half-space defined by the constraint contains all feasible points of the integer program. Let $a = (a_1, \ldots, a_n)$ and $a' = (a'_1, \ldots, a'_n)$, a constraint $a'x \leq b$ is stronger than $ax \leq b$, if $a'_j \geq a_j$, for all $j = 1, \ldots, n$, and there exists $j \in \{1, \ldots, n\}$ such that $a'_j > a_j$. Let $X$ be the set of all feasible integer points. The convex hull of $X$, denoted by $\text{conv}(X)$ contains the set of all feasible convex combinations of integer points in $X$. A facet-defining constraint is a valid constraint such that the dimension of
the intersection that contains the feasible points in $X$ and that satisfies the constraint at equality is one less than the dimension of $\text{conv}(X)$. For a bounded integer program, the $\text{conv}(X)$ will be a polytope bounded by facet-defining constraints. If a polytope is full dimensional, the complete polyhedral description of $\text{conv}(X)$—that is made up of the set of all facet-defining constraints—is unique, otherwise, there are infinitely many complete polyhedral descriptions of $\text{conv}(X)$. Nevertheless, if one can find just one such complete polyhedral description, the linear programming relaxation will return naturally integer solutions. The advantage is that even though cut generation and therefore separation algorithm may still be needed, no branching will be required. The search of facet-defining constraints for an integer program is therefore always a worthy exercise.

In ATSP, a large number of literature can be found in polyhedral results and exact methods based on facet-defining constraints. A one-dimensional mathematical induction proof framework was proposed in Fischetti [16] and proofs of facet-defining properties were developed for several classes of constraints. In Fischetti and Toth [17], separation algorithms were developed for these facet-defining constraints, and a branch-and-cut-and-price algorithm was implemented to obtain exact solutions for ATSP instances. In ATSP with side constraints, Ascheuer et al. [18] presented proofs of validity for a number of inequalities for the ATSP with time-windows, and described a lifting procedure for constructing new valid constraints. In Prize-collecting TSP, polyhedral results can be found in, e.g., Balas [19] and Balas [20]. In ATSP with Replenishment Arcs (RATSP), polyhedral results on facet-defining constraints and branch-and-bound (BNB) exact algorithm where an AP-based Lagrangean relaxation was solved in each node of the BNB tree can be found in, e.g., Mak and Boland [21] and Mak and Boland [22]. Polyhedral results for the Quadratic Selective TSP can be found in, e.g., Mak and Thomadsen [23]. In Black and White TSP, valid inequalities were developed in Ghiani et al. [24], separation algorithms and branch-and-cut were also discussed in the paper.

There is also a large number of literature in polyhedral results for Vehicle Routing-family of problems. See, e.g., Cornuejols and Harche [25] for the Capacitated Vehicle Routing Problem, and Mak and Ernst [26] for the Vehicle Routing Problem with Precedence and Time-windows Constraints. In other cycle or chain problems, the Cardinality Constrained Circuit Problem (CCCP) defined on undirected graphs concerns the finding of one single cycle with length no more than $K$, and an optimal solution is one that minimizes the total arc cost. The work of Bauer et al. [27] presented results on how facet-defining inequalities on related polytope can be transformed into facet-defining inequalities for the CCCP polytope and vice versa. Several classes of inequalities were proven to be facet-defining to the CCCP polytope, and separation algorithms were presented. The $k$-cycle Problem, however, is one that is defined on an undirected graph with $n$ vertices, where a feasible solution is a simple cycle with exactly $k$ vertices. Polyhedral results can be found in, e.g., Nguyen and Maurras [28]. The $p$-cycle Problem is similar to the $k$-cycle Problem, but defined on a directed graph instead. Relevant polyhedral work can be found in, e.g., Hartmann and Özlik [29].

Perhaps the closest combinatorial optimization problem to the CCMcP is the Cardinality Constrained Covering TSP (CCCTSP), see, e.g., Patterson and Rolland [30] where multiple cycles are allowed, the cardinality of these cycles are bounded by an upper bound (in the case of CCMcP, $K$), and a lower bound (in the case of CCCTSP, 0). However, the major difference is that all vertices must be used in a CCCTSP, and the objective is to minimize the total arc costs used in all cycles.
As far as we are aware, there is no polyhedral results for the CCCTSP.

1.4 Contributions and outline of paper

There has been no formal polyhedral study for either of the CCMcP and the CCCCP, except for a short communication in Mak-Hau [6]. The contribution of this paper is therefore to enrich the literature of polyhedral analyses of arc-based formulations of constrained or unconstrained single- or multi-cycle problems defined on directed or undirected graphs. The rest of the paper is outlined as follows. In Section 2, we present our polyhedral results for the CCMcP. We prove that the non-negativity constraints, the degree constraints, and three classes of cardinality violation elimination constraints are facet-defining, and that four other classes of constraints are valid for the CCMcP Polytope. In Section 3, we present the results on the dimension of the CCCCP Polytope, and that the non-negativity constraints and the degree constraints are facet-defining for the CCCCP Polytope. In Section 4, we present preliminary results on how the various constraints improved the computation time on simulated data sets. Finally, in Section 5, we conclude our findings and discuss directions for future research.

2. The Cardinality Constrained Multi-cycle Problem

In this section, we review the basic arc formulation for the CCMcP proposed in Roth et al. [8], propose a number of facet-defining and valid constraints, and present relevant polyhedral results, including: the dimension of the CCMcP polytope, the proofs of some trivial and non-trivial facets, and the validity proofs of some newly identified constraints.

2.1 The arc formulation

A number of studies in the literature have either implemented or used some alternative forms of the arc formulation that was first presented in Roth et al. [8], for example: Abraham et al. [10] showed that the basic arc formulation is not computationally effective (hence the motivation of our paper), Constantino et al. [11] (who compared extensively the arc formulation, the cycle formulation, and two polynomial-size formulation), Anderson et al. [1] (who proposed combined arc and cycle formulation for the CCCCP) and Dickerson et al. [4] (who also proposed polynomial size formulation for both of CCMcP and the CCCCP). Notice that cut generation is the dual form of column generation. In Anderson et al. [1], cut generation is performed in an untraditional way: relaxed problems are solved as integer programs, and violated cycle cardinality constraints are generated and added into the integer program, the resulting integer program is solved, and the procedure repeated. In the context of RATSP, an exact method that applies a Lagrangean relaxation (LR)-based branch-and-bound framework where at each node of the branch-and-bound tree, an Assignment Problem Relaxation (APR) is solved with violations dualised into the objective function and the dual multipliers optimised by subgradient optimisation, has demonstrated computational superiority against a branch-and-cut-and-price exact method (see Mak and Boland [21, 22], and compare with results in Boland et al. [31]). This is perhaps due to the fact that the APRs can be solved in polynomial time, and that facet-defining cuts were used. The aim of
this paper, therefore, is to find strong, valid and facet-defining cuts, motivated by the fact that at least for the CCMcP, when the cardinality constraints are removed, the relaxed problem is an Assignment Problem.

The arc formulation is given as follows. Let \( x_{ij} \), for all \((i,j) \in A\) be a binary decision variable such that \( x_{ij} = 1 \) if arc \((i,j)\) is used, and \( x_{ij} = 0 \) otherwise. Let \( K \) be the cardinality constraint for cycles, and \( \pi = (i_1, \ldots, i_{K+1}) \) be a path (with arc set \( \{(i_j, i_{j+1}) \mid j = 1, \ldots, K\} \)), we call \( \pi \) a minimal cardinality-violation path (MVP). By removing any of the vertices in \( \{i_1, \ldots, i_{K+1}\} \), the simple path formed by any permutation of the remaining \( K \) vertices will be a cardinality feasible path. Let \( \Pi \) be the set of all MVPs. Recall that \( w_{ij} \) is the weight of arc \((i,j)\).

**Model 1** (The Arc Formulation of Roth et al. [8]).

\[
\text{max} \quad \sum_{(i,j) \in A} w_{ij} x_{ij} \tag{1}
\]

\[
\text{s.t.} \quad \sum_{(i,j) \in A} x_{ij} \leq 1, \quad \forall i \in V \tag{2}
\]

\[
\sum_{(j,i) \in A} x_{ji} = \sum_{(i,j) \in A} x_{ij}, \quad \forall i \in V \tag{3}
\]

\[
\sum_{(i,j) \in \pi} x_{ij} \leq K - 1, \quad \forall \pi \in \Pi \tag{4}
\]

\[x_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A. \tag{5}\]

Constraint (2) ensures that each vertex is involved in no more than one cycle, and Constraint (3) preserves the balance of flow for all vertices. Constraint (4) eliminates all MVPs.

### 2.2 A polyhedral study of the CCMcP

For the sake of polyhedral analysis, we consider the CCMcP defined on a complete directed graph. This assumption is not restrictive as we can assign a weight of \(-\infty\) for arcs that do not exist. Let \( K_n \) be a complete digraph with \( n \) vertices and no loops. We have that \(|V| = n\) and \(|A| = n(n - 1)\), for \( V \) the set of vertices and \( A \) the set of arcs.

**Definition 1.** Let \( X = \{x_{ij} \in \{0, 1\}^{|A|} \mid (2) - (5) \text{ are satisfied} \} \). We define the polytope \( P_{n,K} \) to be the convex hull of \( X \). (Recall the definition of the convex hull of all integer feasible solutions of an integer program given in Section 1.3).

**Proposition 1** (Dimension of the CCMcP Polytope, Mak-Hau [6]). Let \( P_{n,K} \) be the CCMcP Polytope defined on \( K_n \) with cardinality \( K \). The dimension of \( P_{n,K} \) is:

\[
\text{dim}(P_{n,K}) = \begin{cases} 
\binom{n}{2}, & K = 2 \\
n^2 - 2n + 1, & 3 \leq K \leq n.
\end{cases}
\]
The proof can be found in Mak-Hau [6].

As the CCMcP comprises only binary decision variables, and that for any two distinct feasible points \( x', x'' \in \{0, 1\}^A \cap \mathcal{P}_{n,K} \), \( x = \alpha x' + \beta x'' \) must be fractional, if \( 0 < \alpha, \beta < 1 \) and \( \alpha + \beta = 1 \), the set of all extreme points for \( \mathcal{P}_{n,K} \) are in fact exactly the set of all feasible solutions for the CCMcP defined on \( \mathcal{K}_n \). We entered the entire set of feasible integer solutions for small \( n \) and \( K \), and used PANDA Lrwal and Reinelt [32] to obtain insights on the structure of some of the facet-defining constraints.

2.2.1 Polyhedral results for trivial constraints

Theorem 1. The trivial constraint \( x_a \geq 0 \), for all \( a \in A \), are facet defining for \( \mathcal{P}_{n,K} \) defined on \( \mathcal{K}_n \), for all \( 2 \leq K \leq n \).

Proof. Let \( \mathcal{F}_{n,K}^{x_a=0} = \{ x : x_a = 0 \} \cap \mathcal{P}_{n,K} \). We prove that \( \mathcal{F}_{n,K}^{x_a=0} \) is facet-defining for \( \mathcal{P}_{n,K} \) by showing that \( \dim(\mathcal{F}_{n,K}^{x_a=0}) = \dim(\mathcal{P}_{n,K}) - 1 \) (recall that \( \dim(\mathcal{P}_{n,K}) = n^2 - 2n + 1 \), see Proposition 1).

[Case \( K = 2 \)] This is trivial, as only one of the \( \binom{n}{2} \) affinely independent feasible single 2-cycles has \( x_a = 1 \). Given that all other solutions have \( x_a = 0 \), together with the 0-vector, \( \dim(\mathcal{F}_{n,2}^{x_a=0}) = \dim(\mathcal{P}_{n,2}) - 1 \).

[Case \( K \geq 3 \)] Trivial Case: \( \dim(\mathcal{F}_{3,3}^{x_a=0}) = 3 \). We first introduce a notation that we will use for the rest of the paper in all the proofs. We use \( e_a \), for \( a \in A \) to represent a \( |A| \)-dimensional binary unit vector, with a “1” in the element associated with arc \( a \), and a “0” in every other element (e.g., if the arc set \( A = \{(1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2)\} \), then \( e_{(1,3)} = (0, 1, 0, 0, 0, 0) \)). We now proceed with the rest of the proof. Without loss of generality (w.l.o.g.), let \( a = (1, 2) \). We have exactly 4 affinely independent feasible solutions: \( (0, e_{1,3} + e_{3,1}, e_{2,3} + e_{3,2}, e_{1,3} + e_{3,2} + e_{2,1}) \), so \( \dim(\mathcal{F}_{3,3}^{x_a=0}) = 3 \).

Induction by \( n \), with \( K = 3 \). Assuming \( \dim(\mathcal{F}_{\eta,3}^{x_a=0}) = \eta^2 - 2\eta \), for any arbitrary \( \eta \geq 3 \). When \( n = \eta + 1 \), we need exactly \( 2\eta - 1 \) new affinely independent feasible solutions with \( x_{12} = 0 \). Let \( \alpha \not\in V \) be the new vertex. We have \( n \) new single 2-cycles in the form of \( (\alpha, j, \alpha) \), one for each \( j \in V \), and \( \eta - 1 \) single 3-cycles in the form of \( (\alpha, i, 1, \alpha) \), one for each \( i \in V \setminus \{1\} \). None of these solutions uses the arc \( (1, 2) \). Now, any 3-cycles of the form: \( (\alpha, i, j, \alpha) \) for distinct \( i, j \in V \), \( (i, j) \neq (1, 2) \), can be obtained as a linear combination of the vectors we have found so far.

\[
e_{\alpha,i} + e_{i,j} + e_{j,\alpha} = (e_{\alpha,i} + e_{i,1} + e_{1,\alpha}) + (e_{i,j} + e_{j,\alpha}) + (e_{\alpha,j} + e_{j,\alpha}) + (e_{1,j} + e_{j,1}) - (e_{j,1} + e_{1,1}) = (e_{\alpha,j} + e_{j,\alpha}) + (e_{1,j} + e_{j,1})\]

Hence the dimension of \( \mathcal{F}_{\eta,3}^{x_a=0} \) is exactly \( \eta^2 - 2\eta \).

Induction by \( K \), for \( K \geq 3 \) with \( n \) unchanged. With \( n \) is held constant, assuming the proposition is true when \( K = \kappa \), where \( \kappa \leq n - 1 \), i.e., \( \dim(\mathcal{F}_{n,\kappa}^{x_a=0}) = n^2 - 2n \), we show that when \( K = \kappa + 1 \), \( \dim(\mathcal{F}_{n,\kappa+1}^{x_a=0}) \) is unchanged.

When \( K = \kappa + 1 \), we are simply required to show that all \((\kappa + 1)\)-cycles are linear combinations of some of the vectors that represent the \( n^2 - 2n + 1 \) affinely independent feasible solutions obtained
for $K = \kappa$. W.l.o.g., let the $(\kappa + 1)$-cycle be $(i_1, i_2, \ldots, i_\kappa, i_{\kappa+1}, i_1)$, for $(i_1, i_2) \neq (1, 2)$, we have:

$$
\left(\sum_{j=1}^{\kappa} e_{ij,ij+1} + e_{i_{n+1},i_1}\right) = \left(\sum_{j=1}^{\kappa-1} e_{ij,ij+1} + e_{i_n,i_1}\right) + (e_{i_n,i_{n+1}} + e_{i_{n+1},i_n}) + (e_{i_{n+1},i_1} + e_{i_1,i_n}) - (e_{i_1,i_{n+1}} + e_{i_{n+1},i_n} + e_{i_n,i_1})
$$

(6)

As we assumed that the arc $(1, 2)$ is not involved in any of the solution vectors in the right hand side of Equation (6), it will not be in the solution vector on the left hand side of the equation either. Furthermore, all multi-cycles are linear combinations of single-cycles. Hence, $\dim(F_{x_n=0}) = \dim(F_{x_3=0})$ for all $3 \leq K \leq n$.

**Definition 2.** Let $\delta^+(i) = \{(i, j) \in A \mid \forall j \in V \setminus \{i\}\}$; $\delta^-(i) = \{(j, i) \in A \mid \forall j \in V \setminus \{i\}\}$, for all $i \in V$; $x(\delta^+(i)) = \{x_a \mid \forall a \in \delta^+(i)\}$; and $x(\delta^-(i)) = \{x_a \mid \forall a \in \delta^-(i)\}$.

**Theorem 2.** The degree constraints

$$
\sum_{a \in \delta^+(i)} x_a \leq 1 \quad \forall i \in V.
$$

are facet defining for $\mathcal{P}_{n, K}$ defined on $K_n$, for all $3 \leq K \leq n$.

**Proof.** Let $F_{x_n=0}^{x(\delta^+(i))=1} = \{x : x(\delta^+(i)) = 1\} \cap P_{n, K}$. **Trivial case:** (when $n = 3$ and $K = 3$). Let $V = \{1, 2, 3\}$, and w.l.o.g., consider the outgoing arcs for vertex 1. We have the following 4 affinely independent feasible solutions: $(e_{12} + e_{21}, e_{13} + e_{31}, e_{12} + e_{23} + e_{31}, e_{13} + e_{32} + e_{21})$, so $\dim(F_{x_3=0}^{x(\delta^+(i))=1}) = 3$.

**Induction by $n$, with $K = 3$.** The proof technique is similar to that of the previous theorem. Let $V = \{1, \ldots, n\}$. Again, w.l.o.g., we consider the outgoing arcs from vertex 1. Assuming the inductive hypothesis is true for $n = \eta$, when $n = \eta + 1$, the $2\eta - 1$ new affinely independent feasible solutions needed are: a single 2-cycle $e_{1,\alpha} + e_{\alpha,1}$, again for $\alpha$ the new vertex; $\eta - 1$ pairs of new affinely independent single 3-cycles, $(1, \alpha, j, 1)$ and $(1, j, \alpha, 1)$, one for each $j \in V \setminus \{1\}$.

**Induction by $K$, for $K \geq 3$ with $n$ unchanged.** Again, assuming $\dim(F_{x_n=0}^{x(\delta^+(i))=1}) = n^2 - 2n + 1$, for any arbitrary $3 \leq \kappa \leq n$ all we need is show that when $K$ is increased to $\kappa + 1$, all $\kappa + 1$-cycles are obtained by linear combinations of cycles of sizes between 2 and $\kappa$. Let the $\kappa + 1$-cycles be of the form: $(1, i_2, \ldots, i_\kappa, i_{\kappa+1}, 1)$. We have:

$$
\left(e_{1,i_2} + \sum_{j=2}^{\kappa} e_{ij,ij+1} + e_{i_{n+1},1}\right) = \left(e_{1,i_2} + \sum_{j=2}^{\kappa-1} e_{ij,ij+1} + e_{i_n,1}\right) + (e_{i_n,i_{n+1}} + e_{i_{n+1},i_n}) + (e_{1,i_n} + e_{i_n,i_{n+1}} + e_{i_{n+1},1}) - (e_{1,i_n} + e_{i_n,i_{n+1}})
$$

(7)

From Equation (7), we can see that all incident vectors concerned contain an outgoing arc from Vertex 1. □
2.3 Polyhedral results for non-trivial constraints

In this section, we present the polyhedral results for non-trivial constraints. We first introduce a list of arc sets that we frequently use in the rest of this section.

**Definition 3.** Let $K_n = (V, A)$ be a complete directed graph with vertex set $V$ and arc set $A$. Let $\rho = (i_1, \ldots, i_r)$ be a simple path on $K_n$, we define:

\[ A(\rho) = \bigcup_{s=1}^{r-1} \{(i_s, i_{s+1})\}; \quad \mathcal{L}(\rho) = A(\rho) \cup \{(i_r, i_1)\}; \]
\[ \mathcal{F}(\rho) = \bigcup_{s=1}^{r-1} \bigcup_{t=s+1}^{r} \{(i_s, i_t)\}; \quad \mathcal{B}(\rho) = \bigcup_{s=3}^{r-1} \bigcup_{t=2}^{s-1} \{(i_s, i_t)\}; \]
\[ x(S) = \sum_{a \in S} x_a; \quad \forall S \in \{A(\rho), \mathcal{L}(\rho), \mathcal{F}(\rho), \mathcal{B}(\rho)\}; \]
\[ (i, S) = \bigcup_{j \in S} \{(i, j)\}; \quad (S, i) = \bigcup_{j \in S} \{(j, i)\} \]

If $r = 1$ then $A(\rho) = \mathcal{L}(\rho) = \mathcal{F}(\rho) = \mathcal{B}(\rho) = \emptyset$.

Whilst the left hand side (LHS) of the basic MVP violation elimination constraint (4) concerns only the path arcs $A(\rho)$, the facet-defining MVP-based constraints we have found consider more than just the path arcs, and they are hence stronger (recall the definition of a “stronger” constraint provided in Section 1.3).

**Lemma 1.** Let $\tau = (i_1, \ldots, i_K)$ be a maximally cardinality feasible path (MFP). The following MFP constraints are valid for $P_{n,K}$ defined on $K_n$, for all $K \geq 3$ and $n \geq K + 1$.

\[ x(\mathcal{F}(\tau)) - x_{i_K, i_1} \leq K - 2 \quad (8) \]

The proof is trivial: by degree constraints, $x(\mathcal{F}(\tau))$ is at most $K - 1$. As $x(\mathcal{F}(\tau)) = K - 1$ if and only if $x_a = 1$ for all $a \in A(\tau)$, in order to be cardinality-feasible, we have to have $x_{i_K, i_1} = 1$.

**Theorem 3.** Constraint (8) is facet-defining for $P_{n,K}$ defined on $K_n$, for all $K \geq 3$ and $|V| \geq K + 1$.

**Proof.** Again, we shall prove this by double induction.

**Trivial case:** (when $|V| = 4$ and $K = 3$). W.l.o.g., let $|V| = \{1, 2, 3, 4\}$, and let $\tau = (3, 4, 2)$. The corresponding constraint (8) is as follows.

\[ x_{34} + x_{42} + x_{32} \leq 1 + x_{23} \]

The dimension of all feasible solutions that satisfy (8) at equality is exactly $\dim(P_{4,3}) - 1$ (see below for the nine affinely independent feasible solutions, where elements in bold are arcs used for
the first time, e.g., in the second vector $e_{34} + e_{43}$, we have the arc $(3, 4)$ used for the first time, hence guaranteeing the second vector is affinely independent to the first vector).

\[
\begin{align*}
e_{24} + e_{32} + e_{43} \\
e_{34} + e_{43} \\
e_{24} + e_{42} \\
e_{14} + e_{21} + e_{42} \\
e_{13} + e_{21} + e_{32} \\
e_{13} + e_{34} + e_{41} \\
e_{13} + e_{31} + e_{24} + e_{42} \\
e_{23} + e_{34} + e_{42} \\
e_{12} + e_{21} + e_{34} + e_{43}
\end{align*}
\]

**Induction by $K$ (with $n = K + 1$).** Assuming that the theorem is true for $K = \kappa$, for any arbitrary $\kappa \geq 3$. We have that $n = \kappa + 1$, for $n$ the total number of vertices. W.l.o.g., let $\tau = \{i_1, \ldots, i_\kappa\}$, and let the only vertex in $V$ not involved in $\tau$ be $\omega$, our inductive hypothesis is that we have exactly $n^2 - 2n + 1$ affinely independent feasible solutions that satisfy:

\[
x(F(\tau)) - x_{i_\kappa,i_1} = \kappa - 2 \tag{9}
\]

We now prove that the theorem is true for $K = \kappa + 1$ by reconstructing those $n^2 - 2n + 1$ vectors and adding $2n - 1$ new, affinely independent feasible solutions that satisfy (9).

Let $\alpha$ be the newly introduced vertex. For each of the $n^2 - 2n + 1$ affinely independent solutions for $K = \kappa$, in our inductive hypothesis, the vertex $i_\kappa$ is either involved in a cycle, given by $(i_\gamma, \ldots, i_\beta, i_\kappa, i_\gamma)$, or not involved in a cycle (in other words, $\{i_\gamma, \ldots, i_\beta\} = \emptyset$). We modify these solutions according to the following rules so that Constraint (9) will be satisfied at equality:

\[
(i_\gamma, \ldots, i_\beta, i_\kappa, \alpha, i_\gamma), \text{ if } \{i_\gamma, \ldots, i_\beta\} \neq \emptyset; \tag{10}
\]

\[
(i_\kappa, \alpha, i_\kappa), \text{ otherwise.} \tag{11}
\]

Notice that in these solutions, the incoming arc to the new vertex $\alpha$ is fixed to be from $i_\kappa$, and that they use exactly $\kappa - 1$ arcs in the set $F(\tau) \cup \{(i_\kappa, \alpha)\}$. We now introduce $2n - 1$ new affinely independent feasible solutions that *satisfy Constraint (9) at equality*, i.e., either using $\kappa - 1$ arcs in $F(\psi)$, for $\psi = (i_1, \ldots, i_\kappa, \alpha)$ or $\kappa$ arcs in $F(\psi)$ together with the arc $(\alpha, i_1)$. For convenience of notation, we define $e(A) = \sum_{a \in A} e_a$, for any $A \subseteq A$.

1. First of all, in all previously introduced solutions, the new vertex $\alpha$ is involved with the arc $(i_\kappa, \alpha)$ fixed. Hence the solution $e(L(\tau'))$,

   for $\tau' = (i_1, \ldots, i_\kappa)$ is affinely independent to all previously introduced solutions and satisfies the constraint at equality.

2. Now, as all incoming arcs to $\alpha$ are fixed to be from $i_\kappa$ in the previous solutions, we add the following $n - 1$ new solutions each using an unused arc in $\delta^-(\alpha)$, whilst fixing the outgoing arc from $\alpha$ to be $(\alpha, \omega)$.

11
(a) Consider \((i_j, \alpha)\), for all \(j \in \{2, \ldots, \kappa - 1\}\), each having \(\delta^+(\alpha)\) fixed to \((\alpha, \omega)\). There are \(n - 3\) of them. We have:

\[
e(\mathcal{L}(\tau')) + e_{i_j, \alpha} + e_{\alpha, \omega} + e_{\omega, i_j}
\]

for \(\tau' = (i_1, \ldots, i_{j-1}, i_{j+1}, \ldots, i_\kappa)\). We then add the following two solutions:

(b) For \((i_1, \alpha)\), we have:

\[
e_{i_1, \alpha} + e_{\alpha, \omega} + e_{\omega, i_1} + e(\mathcal{L}(\tau'))
\]

for \(\tau' = (i_2, \ldots, i_\kappa)\).

(c) As for \((\omega, \alpha)\), we have:

\[
e(\mathcal{L}(\tau')) + e_{\omega, \alpha} + e_{\alpha, \omega}
\]

for \(\tau' = (i_1, \ldots, i_\kappa)\).

3. (a) We have \(n - 3\) solutions, each using a 2-cycle of the form \((i_j, \alpha, i_j)\), for \(j \in \{2, \ldots, \kappa - 1\}\), and that they cannot be obtained as a linear combination of any of the previously introduced solutions as the outgoing arc from \(\alpha\) was fixed to \(\omega\) in Case 1(a). These solutions are:

\[
e_{i_j, \alpha} + e_{\alpha, i_j} + e(\mathcal{L}(\tau'))
\]

for \(\tau' = (i_1, \ldots, i_{j-1}, i_{j+1}, \ldots, i_\kappa)\).

(b) The arc \((i_1, \alpha)\) was used only once previously as a 3-cycle in Case 2(b) with arc \((\alpha, \omega)\) fixed, and should there be a previously introduced solution with arc \((\alpha, j)\) used (in the inductive hypothesis), it has \((i_\kappa, \alpha)\) fixed, hence the following solution is affinely independent to all previously introduced solutions.

\[
e(\mathcal{L}(\tau')), \quad \text{for } \tau' = (i_1, \alpha, i_2, \ldots, i_\kappa).
\]

4. In all previously introduced solutions, the arc \((\alpha, i_\kappa)\) was only used as a 2-cycle in (11), and the arc \((i_\kappa, i_1)\), should it had been used before, must be used together with \(A(\tau)\). Hence, the solution:

\[
e(\mathcal{L}(\tau')), \quad \text{for } \tau' = (i_1, \alpha, i_2, \ldots, i_\kappa).
\]

**Induction by** \(n\) \((n = |V|)\) when \(K \leq n - 1\) is fixed. Assume that the theorem is true for \(n = \eta\), we now show that the theorem is true for \(n = \eta + 1\) whilst \(K \geq 3\) is held constant. Let \(V(\tau) = \{i_1, \ldots, i_K\}\) and \(\overline{V(\tau)} = V \setminus V(\tau) = \{i_{K+1}, \ldots, i_\eta\}\). Again, let \(\alpha\) be the new index in the inductive process. We need \(2\eta - 1\) new affinely independent feasible solutions that satisfy (9) at equality by the inductive assumption.

1. We first introduce the \(\eta\) new affinely independent feasible solutions, each using a distinct arc in \(\delta^-(\alpha)\) while having the outgoing arc from \(\alpha\) fixed to be \((\alpha, i_\eta)\). These solutions are listed below. We have:
(a) \( \eta - K \) solutions in the the form of:

\[
\eta - K + 1, \quad \alpha + \alpha, i_\eta, i_{i_j} + e_{i_\eta,i_j},
\]

one for each \( j \in \{ K, \ldots, \eta - 1 \} \) where \( \tau' = (i_1, \ldots, i_{K-1}) \);

(b) \( e(\mathcal{L}(\tau')) + e_{i_\eta, i_j} + e_{\alpha, i_j} \),

where \( \tau' = (i_1, \ldots, i_{K-1}) \);

(c) \( e(\mathcal{L}(\tau')) + e_{i_j, \alpha} + e_{\alpha, i_\eta} + e_{i_\eta, \alpha} \),

where \( \tau' = (i_2, \ldots, i_K) \);

(d) \( K - 2 \) solutions, one for each \( j \in \{ 2, \ldots, K - 1 \} \), in the form of:

\[
e(\mathcal{L}(\tau_1')) + e(\mathcal{L}(\tau_2'))\]

for \( \tau_1' = (i_j, \alpha, i_\eta, i_1) \) and \( \tau_2' = (i_2, \ldots, i_{j-1}, i_{j+1}, \ldots, i_K) \).

2. Now, we introduce the remaining \( \eta - 1 \) new affinely independent feasible solutions needed, each with an arc from \( \delta^+(\alpha) \setminus \{ (\alpha, i_\eta) \} \).

(a) \( e(\mathcal{L}(\tau')) + e_{i_K, \alpha} + e_{\alpha, i_K} \),

where \( \tau' = (i_1, \ldots, i_{K-1}) \);

(b) \( e(\mathcal{L}(\tau')) + e_{i_1, \alpha} + e_{\alpha, i_1} \),

where \( \tau' = (i_2, \ldots, i_K) \);

(c) \( K - 2 \) solutions in the form of:

\[
e(\mathcal{L}(\tau')) + e_{i_j, \alpha} + e_{\alpha, i_j},
\]

one for each \( j \in \{ 2, \ldots, K - 1 \} \) where \( \tau' = (i_1, \ldots, i_{j-1}, i_{j+1}, \ldots, i_K, i_\eta) \);

(d) \( \eta - K - 1 \) solutions in the form of:

\[
e(\mathcal{L}(\tau')) + e_{i_j, \alpha} + e_{\alpha, i_j},
\]

one for each \( j \in \{ K + 1, \ldots, \eta - 1 \} \) where \( \tau' = (i_1, \ldots, i_{K-1}) \).

\[ \square \]

Some strong, facet-defining minimal infeasible path elimination constraints has been found in the context of RATSP and ATSP/VRP with time windows/precedence constraints, see [21, 26] respectively, but they in general eliminate subtours as well, and is thus invalid for the CCMcP. For a CCMcP, “subtours” are allowed, as long as the sizes are not larger than the cardinality restriction \( K \). In this section, we introduce a couple of facet-defining constraints that are based on MVPs.

**Lemma 2.** The constraints:

\[
x(\mathcal{F}(\pi)) \leq K - 1,
\]

for all minimal cardinality violation path \( \pi \in \Pi \), are valid.
Proof. If \((x_{i_1,i_2} = 0) \lor (x_{i_K,i_{K+1}} = 0)\), then by degree constraint, at most \(K - 1\) arcs in \(\mathcal{F}(\pi)\) can be used. Otherwise, the problem is reduced to a MVP \(\pi' = (i_2, \ldots, i_K)\) with cardinality \(K - 2\), where at most \(K - 3\) arcs in \(\mathcal{F}(\pi')\) can be used. Repeating the same argument, we have two cases below.

1. For \(K\) odd: we are left with arc \([\lceil K/2 \rceil, \lceil K/2 \rceil + 1]\) and a cardinality limit of 1, and clearly no arcs can be used.
2. For \(K\) even: we are left with arcs \((K/2, K/2 + 1), (K/2, K/2 + 2), (K/2 + 1, K/2 + 2)\) and a cardinality limit of 2, and clearly at most 1 arc can be used.

Theorem 4. Constraints (12) are facet defining for \(P_{n,K}\) defined on \(K_n\), for \(K \geq 3\) and \(|V| \geq K + 2\).

Proof. We prove this by directly finding \(n^2 - 2n + 1\) affinely independent feasible vectors that satisfy the constraint at equality. We do so in two steps. In Step 1, we construct \((n - 1)(n - 2)\) such vectors, and in Step 2, we construct \(n - 1\) such vectors. Again, w.l.o.g., let \((i_1, \ldots, i_{K+1})\) be a MVP.

Step One We consider all pairs of distinct vertices \(i_\alpha, i_\beta \in V \setminus \{i_1\}\). By permutation, we have \((n - 1)(n - 2)\) such pairs, and we construct a solution for each pair. We have four cases.

1. Consider \(\alpha, \beta \in \{2, \ldots, K + 1\}\).
   a. When \(\beta > \alpha\). The general solutions take the form:
      \[e(\mathcal{L}(\tau_1)) + e(\mathcal{L}(\tau_2)),\]
      for \(\tau_1 = (i_1, i_\alpha, i_\beta)\), and \(\tau_2 = (i_2, \ldots, i_{\alpha-1}, i_{\alpha+1}, \ldots, i_{\beta-1}, i_{\beta+1}, \ldots, i_{K+1})\), with the exception of the following general cases.
      i. \(\beta = \alpha + 1\), where \(\tau_2 = (i_2, \ldots, i_{\alpha-1}, i_{\alpha+1}, \ldots, i_{K+1})\);
      ii. \(\alpha = 2\), where
         \[\tau_2 = \begin{cases} (i_3, \ldots, i_{\beta-1}, i_{\beta+1}, \ldots, i_{K+1}), & \text{if } \beta > \alpha + 1 \\ (i_{\beta+1}, \ldots, i_{K+1}), & \text{otherwise}; \end{cases}\]
      iii. \(\beta = K + 1\), where
         \[\tau_2 = \begin{cases} (i_2, \ldots, i_{\alpha-1}, i_{\alpha+1}, \ldots, i_K), & \text{if } \beta > \alpha + 1 \\ (i_2, \ldots, i_{K-1}), & \text{otherwise}. \end{cases}\]
   b. When \(\beta < \alpha\). All cases in Case 1(a) repeated, with \(\tau_1\) the same, but \(\alpha\) and \(\beta\) interchanged in \(\tau_2\).

2. Consider \(\alpha \in \{2, \ldots, K + 1\}\), and \(\beta \in \{K + 2, \ldots, n\}\). Again, the general solutions take the form:
   \[e(\mathcal{L}(\tau_1)) + e(\mathcal{L}(\tau_2)),\]
   for \(\tau_1 = (i_1, i_\alpha, i_\beta)\), and
   \[\tau_2 = \begin{cases} (i_2, \ldots, i_{\alpha-1}, i_{\alpha+1}, \ldots, i_{K+1}), & \text{if } 3 \leq \alpha \leq K \\ (i_3, \ldots, i_{K+1}), & \text{if } \alpha = 2 \\ (i_2, \ldots, i_K), & \text{if } \alpha = K + 1 \end{cases}\]
3. Consider $\beta \in \{2, \ldots, K + 1\}$, and $\alpha \in \{K + 2, \ldots, n\}$. The general solutions take the form.

$$e(\mathcal{L}(\tau_1)) + e(\mathcal{L}(\tau_2)),$$

for $\tau_1 = (i_\alpha, i_\beta)$ and

$$\tau_2 = \begin{cases} (i_1, \ldots, i_{\beta-1}, i_{\beta+1}, \ldots, i_{K+1}), & \text{if } \beta \leq K \\ (i_1, \ldots, i_K), & \text{o.w.} \end{cases}$$

4. Consider $\alpha, \beta \in \{K + 2, \ldots, n\}$. We have

$$e(\mathcal{L}(\tau_1)) + e(\mathcal{L}(\tau_2)),$$

for $\tau_1 = (i_1, i_\alpha, i_\beta)$, and $\tau_2 = (i_2, \ldots, i_{K+1})$.

**Step Two** Now we consider a solution for each $i_\alpha \in V \setminus \{i_1\}$, that induces a 2-cycle $(i_1, i_\alpha, i_1)$, hence guaranteeing the affine independency. The solutions are given by:

$$e(\mathcal{L}(\tau_1)) + e(\mathcal{L}(\tau_2)),$$

for $\tau_1 = (i_1, i_\alpha)$, and

$$\tau_2 = \begin{cases} (i_3, \ldots, i_{K+1}), & \text{if } \alpha = 2 \\ (i_2, \ldots, i_{\alpha-1}, i_{\alpha+1}, \ldots, i_{K+1}), & \text{if } 3 \leq \alpha \leq K \\ (i_2, \ldots, i_K), & \text{if } \alpha = K + 1 \\ (i_2, \ldots, i_{K+1}), & \text{o.w.} \end{cases}$$

**Definition 4.** Let $\rho = (i_1, \ldots, i_r)$ be a simple path defined on $K_n$, for all $3 \leq K \leq n$, and let the arc set of $\rho$ be $A(\rho)$ (as defined in Definition 3). We further define:

1. $\mathcal{H}(\rho)$ to be

$$\mathcal{H}(\rho) = \{(i_j, i_{j+2}) \mid j = 1, \ldots, r - 2\};$$

2. $\mathcal{R}(\rho)$ to be

$$\mathcal{R}(\rho) = \{(i_j, i_{j-1}) \mid j = 3, \ldots, r - 1\}.$$

**Axiom 1.** Let $\rho = (i_1, \ldots, i_r)$ be a simple path defined on $K_n$, for all $3 \leq K \leq n$, by degree constraints, $x(A(\pi)) + x(\mathcal{H}(\pi)) \leq r - 1$.

**Lemma 3.** Let $\pi = (i_1, \ldots, i_{K+1})$ be a MVP, defined on $K_n$, for all $3 \leq K \leq n$. The following constraint is valid for $P_{n,K}$.

$$x(A(\pi)) + x(\mathcal{H}(\pi)) + x(\mathcal{R}(\pi)) \leq K - 1 \quad (13)$$

**Proof.** First of all, as we shall see later that (12) is valid, and since $A(\pi) \cup \mathcal{H}(\pi) \subseteq F(\pi)$,

$$x(A(\pi)) + x(\mathcal{H}(\pi)) \leq K - 1 \quad (14)$$
is clearly valid as well. We then obtain (13) by sequential lifting. First, we obtain:

$$x(A(\pi)) + x(H(\pi)) + x(\pi_j) + x(\pi_j) + \alpha_{i_j,i_{j-1}} x_{i_j,i_{j-1}} \leq K - 1.$$ (15)

When $x_{i_{K-1},i_K} = 1$, if $x_{i_{K-2},i_K} = 1$, then we have a two-cycle, together with $A(\tau) \cup H(\tau)$, for $\tau = (i_1,\ldots,i_{K-2})$. By Axiom 1, no more than $K - 3$ arcs will be used in $A(\tau) \cup H(\tau)$. If $x_{i_{K-1},i_K} = 0$, and exactly $K - 3$ arcs are used in $A(\tau) \cup H(\tau)$, then at most one of $(i_{K-2},i_K)$ or $(i_{K-1},i_{K+1})$ can be used in order to be cardinality feasible. Hence $\alpha_{i_{K-1},i_K} = 1$.

In the general step of the sequential lifting, we attempt to lift in $x_{i_j,i_{j-1}}$, for $j \in \{3,\ldots,K\}$, and a similar argument can be used. Let

$$\mathcal{R}^*(\pi_j) = \{ (i_{K-\ell}, i_{K-\ell-1}) \mid \ell = 0, \ldots, K - j - 1 \}$$

$x(A(\pi)) + x(H(\pi)) + x(\pi_j) + x(\mathcal{R}^*(\pi_j)) + \alpha_{i_j,i_{j-1}} x_{i_j,i_{j-1}} \leq K - 1.$ (16)

When $x_{i_j,i_{j-1}} = 1$, and $x_{i_{j-1},i_j} = 1$, then at most $j - 3$ arcs can be used in $H(\tau_1)$, for $\tau_1 = (i_1,\ldots,i_{j-2})$, and by degree constraint, at most $K - j$ arcs can be used in $A(\tau_2) \cup H(\tau_2) \cup \mathcal{R}^*(\pi_j)$ for $\tau_2 = (i_{j+1},\ldots,i_{K+1})$, hence no more than $K - 1$ arcs will be used in total. On the other hand, if $x_{i_{j-1},i_j} = 0$, and if exactly $j - 3$ arcs are used in $A(\tau_1) \cup H(\tau_1)$ and exactly $K - j$ arcs are used in $A(\tau_2) \cup H(\tau_2) \cup \mathcal{R}^*(\pi_j)$, then at most one of $(i_{j-2},i_j)$, $(i_{j-1},i_{j+1})$, and $(i_{j+1},j)$ can be used, for using both of $(i_{j-2},i_j)$, and $(i_{j-1},i_{j+1})$ will induce a cardinality violation. Using both of $(i_{j-1},i_{j+1})$, and $(i_{j+1},j)$, however, implies that by degree constraint, only $K - j - 1$ arcs in $A(\tau_2) \cup H(\tau_2) \cup \mathcal{R}^*(\pi_j)$ can be used. Hence $\alpha_{i_j,i_{j-1}} = 1$, and we have proved the validity of Constraint (13).

\textbf{Theorem 5.} Let $\pi = (i_1,\ldots,i_{K+1})$ be a MVP. Constraint (13) is facet-defining for $P_{n,K}$ defined on $K_n$, for all $K \geq 3$ and $|V| \geq K + 2$.

We prove by double induction. First we define more notation.

\textbf{Definition 5.} Let $\rho = (i_1,\ldots,i_r)$, for $r \geq 4$ be a simple path, we define:

$$\Delta(\rho) = \begin{cases} 
\{(i_{2j+1},i_{2j+2}), (i_{2j+2},i_{2j+1}) \mid j = 0, \ldots, \frac{r}{2} - 1\}, & \text{if } r \text{ is even;} \\
\{(i_{2j+1},i_{2j+2}), (i_{2j+2},i_{2j+1}) \mid j = 0, \ldots, \frac{r-3}{2} - 1\} \\
\cup \{(i_{r-2},i_{r}), (i_{r},i_{r-1}), (i_{r-1},i_{r-2})\}, & \text{otherwise,}
\end{cases}
$$

and $x(\Delta(\rho)) = \sum_{a \in \Delta(\rho)} x_a$.

\textbf{Trivial Case:} (with $|V| = 5, K = 3$). W.l.o.g., we assume that $\pi = (4,5,3,1)$, we have the following 16 affinely independent feasible solutions that satisfies the constraint at equality by using exactly 2 arcs from the set $\{(4,5),(5,3),(3,1),(4,3),(5,1),(3,5)\}$.
Induction by $K$ (with $n = K + 2$). Assuming that the theorem is true for $K = \kappa$, for any arbitrary $\kappa \geq 3$, $n = \kappa + 2$, we now show that the theorem is true for $K = \kappa + 1$. W.l.o.g., let $\pi = (i_1, \ldots, i_{\kappa + 1})$ be a minimal violation path, and $\omega$ be the only vertex on the digraph not involved in $\pi$. Again, let $\alpha$ be the new vertex.

First of all, the $n^2 - 2n + 1$ affinely independent feasible solutions obtained for $K = \kappa$ can be obtained by modification similar to (10) and (11). We now introduce the $2n - 1$ new affinely independent feasible solutions in the following sequence, each satisfying Constraint (13) at equality.

1. First we consider the $n - 1$ solutions each with an arc that leaves the new vertex $\alpha$ and enters a node in the set $\{i_1, \ldots, i_\kappa, \omega\}$, each having the outgoing arc from $\alpha$ be fixed to $\omega$.
   (a) For $j = 1$, we have:
   $$e_{i_1, \alpha} + e_{\alpha, \omega} + e_{\omega, i_1} + e(\Delta(\rho)),$$
   where $\rho = (i_2, \ldots, i_{\kappa + 1})$.
   (b) For $j \in \{2, \ldots, \kappa - 1\}$, we have:
   $$e(\mathcal{L}(\tau)) + e(\Delta(\rho)),$$
   where $\tau = (i_1, \ldots, i_j, \alpha, \omega)$ and $\rho = (i_{j+1}, \ldots, i_{\kappa + 1})$.
   (c) For $j = \kappa$, we have:
   $$e(\mathcal{L}(\tau_1)) + e(\mathcal{L}(\tau_2))$$
   where $\tau_1 = (i_1, \ldots, i_{\kappa - 1})$, and $\tau_2 = (i_{\kappa + 1}, i_\kappa, \alpha, \omega)$.
   (d) For $j = \omega$, we have:
   $$e(\mathcal{L}(\tau)) + e_{\alpha, \omega} + e_{\omega, \alpha},$$
   where $\tau = (i_1, \ldots, i_{\kappa + 1})$.

2. Next, we have $n - 2$ solutions, each using an arc that leaves $\alpha$ and enters a node $i_j$ for $j \in \{1, \ldots, \kappa - 1\}$. As the arc $(i_\kappa, i_{\kappa + 1})$ is not used in these solutions, they are affinely independent to those $n^2 - 2n + 1$ obtained in the modification step.
(a) For $j = \kappa$, we have:

$$e(\mathcal{L}(\tau)) + e_{i_\kappa, \alpha} + e_{\alpha, i_\kappa}$$

where $\tau = (i_1, \ldots, i_{\kappa-1}, i_{\kappa+1})$.

(b) For $j \in \{2, \ldots, \kappa-1\}$, we have:

$$e(\mathcal{L}(\tau_1)) + e(\mathcal{L}(\tau_2))$$

where $\tau_1 = (i_1, \ldots, i_{j-1})$, and $\tau_2 = (i_j, \ldots, i_{\kappa-1}, i_\kappa, \alpha)$.

(c) For $j = 1$, we have:

$$e_{i_1, \alpha} + e_{\alpha, i_1} + e(\Delta(\rho))$$

where $\rho = (i_2, \ldots, i_{\kappa+1})$.

3. We have 1 solution with arc $(\alpha, i_{\kappa+1})$:

$$e(\mathcal{L}(\tau_1)) + e(\mathcal{L}(\tau_2))$$

where $\tau_1 = (i_1, \ldots, i_{\kappa-1})$ and $\tau_2 = (i_\kappa, \alpha, i_{\kappa+1})$.

4. We have 1 solution as follows that was not feasible for $K = \kappa$ but feasible for $K = \kappa + 1$.

$$e(\mathcal{L}(\tau))$$

where $\tau = (i_1, \ldots, i_{n+1})$.

*Induction by $n$ ($n = |V|$) when $K \leq n - 2$ is fixed.* Assume that the theorem is true for $n = \eta$, we now show that the theorem is true for $n = \eta + 1$.

Again, w.l.o.g., let $V = \{i_1, \ldots, i_\eta\}$ and $\pi = (i_1, \ldots, i_{K+1})$ be a MVP. Let $\alpha$ be the new vertex in the inductive process. We now introduce $2\eta - 1$ new affinely independent feasible solutions that uses exactly $K - 1$ arcs in the set $\mathcal{A}(\pi) \cup \mathcal{H}(\pi) \cup \mathcal{R}(\pi)$.

1. We first introduce the $\eta$ new affinely independent feasible solutions, each using a distinct arc in $\delta^{-}(\alpha)$ while having the outgoing arc from $\alpha$ fixed to be $(\alpha, i_\eta)$. These solutions are listed below. We have:

   (a) $\eta - K - 1$ solutions in the the form of:

$$e(\mathcal{L}(\pi')) + e_{i_j, \alpha} + e_{\alpha, i_\eta} + e_{i_\eta, i_1}$$

one for each $j \in \{K + 1, \ldots, \eta - 1\}$ where $\pi' = (i_1, \ldots, i_K)$;

(b) $e(\mathcal{L}(\pi')) + e_{i_\eta, \alpha} + e_{\alpha, i_\eta}$, where $\pi' = (i_1, \ldots, i_K)$;

(c) $e(\mathcal{L}(\pi')) + e_{i_1, \alpha} + e_{\alpha, i_\eta} + e_{i_\eta, i_1}$, where $\pi' = (i_2, \ldots, i_{K+1})$;

(d) $K - 1$ solutions, one for each $j \in \{2, \ldots, K\}$, in the form of:

$$e(\mathcal{L}(\pi'_1)) + e(\mathcal{L}(\pi'_2))$$

for $\pi'_1 = (i_j, \alpha, i_\eta)$ and $\pi'_2 = (i_1, \ldots, i_{j-1}, i_j, i_{j+1}, \ldots, i_{K+1})$. 18
2. Now, we introduce the remaining \( \eta - 1 \) new affinely independent feasible solutions needed, each with an arc from \( \delta^+(\alpha) \setminus \{\alpha, \eta\} \).

(a) \( \eta - K - 2 \) solutions in the form of:

\[
\text{one for each } j \in \{K + 2, \ldots, \eta - 1\} \text{ where } \pi' = (i_1, \ldots, i_K);
\]

\[
e(\mathcal{L}(\pi')) + e_{i_j,\alpha} + e_{\alpha,i_j},
\]

(b) \( \eta - K - 2 \) solutions in the form of:

\[
\text{one for each } j \in \{K + 2, \ldots, \eta - 1\} \text{ where } \pi' = (i_1, \ldots, i_K);
\]

\[
e(\mathcal{L}(\pi')) + e_{i_{K+1},\alpha} + e_{\alpha,i_{K+1}},
\]

(c) \( \eta - K - 2 \) solutions in the form of:

\[
\text{one for each } j \in \{K + 2, \ldots, \eta - 1\} \text{ where } \pi' = (i_1, \ldots, i_K);
\]

\[
e(\mathcal{L}(\pi')) + e_{i_1,\alpha} + e_{\alpha,i_1},
\]

(d) \( K - 1 \) solutions in the form of:

\[
\text{one for each } j \in \{2, \ldots, K\} \text{ where } \pi' = (i_1, \ldots, i_{j-1}, i_{j+1}, \ldots, i_K+1).
\]

2.4 Valid constraints and proofs of validity

In this section, we present a number of classes of valid constraints for the CCMcP that we have recently identified.

**Definition 6.** Let \( \pi = (i_1, \ldots, i_{K+1}) \) be a MVP, let \( L_j = \{1, \ldots, j - 1, j + 1, \ldots, K - 1\} \), and \( \Lambda_j(\ell) = \{\ell + 2, \ldots, K + 1\} \setminus \{j + 1\} \), we define:

\[
A^*(\pi) = \{(i_\ell, i_\lambda) \mid \ell \in L_j \text{ and } \lambda \in \Lambda_j(\ell)\}
\]

**Lemma 4.** Let \( \pi = (i_1, \ldots, i_{K+1}) \) be a MVP, \( \pi_1 = (i_1, \ldots, i_j) \) and \( \pi_2 = (i_{j+1}, \ldots, i_{K+1}) \), for any arbitrary \( j \in \{2, \ldots, K - 1\} \), and \( S = V \setminus V(\pi) \), the following constraint is valid for \( P_{\eta,K} \):

\[
x(\mathcal{A}(\pi_1)) + x(\mathcal{A}(\pi_2)) + x(A^*) \leq K - 2 + x_{i_j,i_1} + x(i_j, S).
\]

**Proof.** First of all, if all arcs in \( \mathcal{A}(\pi_1) \cup \mathcal{A}(\pi_2) \) are used, then we have \( K - 1 \) arcs, and therefore the arc \( (i_j, i_{j+1}) \) must not be used, otherwise the cardinality constraint will be violated. Hence the path \( \mathcal{A}(\pi_1) \) must exit from vertex \( i_j \) either to the first vertex of the path or to a vertex in \( V \setminus V(\pi) \). Hence we have the following trivial constraint.

\[
x(\mathcal{A}(\pi_1)) + x(\mathcal{A}(\pi_2)) \leq K - 2 + x_{i_j,i_1} + x(i_j, S).
\]

To lift in \( x_{i_\ell,i_\lambda} \), for any arbitrary \( (i_\ell, i_\lambda) \in A^*(\pi) \), if \( x_{i_\ell,i_\lambda} = 1 \), then

\[
\sum_{\eta=\{1,\ldots,j-1\}\setminus\{\ell\}}\sum_{a \in \delta^+(i_\eta)} x_a = j - 2,
\]
as \( i_j \) has no outgoing arcs. Further,
\[
\sum_{\eta=\{j+1,\ldots,K-1\}\setminus\{i\}} x_\eta = K - j - 2.
\]

Now if \( x(\delta^+(i_{K-1})) = 1 \), then it must be that the arc \((i_{K-1}, i_{K+1})\) is used and therefore the arc \((i_K, i_{K+1})\) cannot be used. The lifting of all arcs in \( A^*(\pi) \) is order independent as the argument above concerns degree constraints only.

\[\square\]

**Lemma 5.** Let \( K = 3 \), and \( \pi = (i_1, i_2, i_3, i_4) \) be a MVP. The following two classes of constraints are valid for \( P_{n,3} \):
\[
\begin{align*}
x(A(\pi)) + 3x_{i_4,i_1} + 2x_{i_2,i_1} + 2x_{i_4,i_2} + 2x_{i_3,i_1} & \leq 4. \quad (20) \\
x(A(\pi)) + 3x_{i_4,i_1} + 2x_{i_4,i_3} + 2x_{i_4,i_2} + 2x_{i_3,i_1} & \leq 4. \quad (21)
\end{align*}
\]

We will only provide the proof for Constraint (20), as that of Constraint (21) is similar.

**Proof.** By degree constraint, we have
\[
x_{i_1,i_2} + x_{i_2,i_3} + x_{i_3,i_4} + x_{i_4,i_1} \leq 4.
\]
However, when \( x_{i_4,i_1} = 1 \), \( x_{i_1,i_2} + x_{i_2,i_3} + x_{i_3,i_4} \leq 1 \), otherwise the cardinality violation is induced. Hence, the coefficient of \( x_{i_4,i_1} \) can be lifted to 3, and we have:
\[
x_{i_1,i_2} + x_{i_2,i_3} + x_{i_3,i_4} + 3x_{i_4,i_1} \leq 4.
\]
Now consider lifting in the variable \( x_{i_2,i_1} \): when \( x_{i_2,i_1} = 1 \), \( x_{i_2,i_3} = x_{i_4,i_1} = 0 \). As \( x_{i_1,i_2} + x_{i_3,i_4} \leq 2 \): we can lift in \( x_{i_4,i_2} \) with a coefficient of 2, and obtain:
\[
x_{i_1,i_2} + x_{i_2,i_3} + x_{i_3,i_4} + 3x_{i_4,i_1} + 2x_{i_2,i_1} \leq 4.
\]
When \( x_{i_4,i_3} = 1 \), \( x_{i_4,i_1} = x_{i_1,i_2} = 0 \), as we cannot have \((i_3, i_4), (i_4, i_2)\), and \((i_2, i_1)\) at the same time, we can lift in \( x_{i_4,i_2} \) with a coefficient of 2, and obtain:
\[
x_{i_1,i_2} + x_{i_2,i_3} + x_{i_3,i_4} + 3x_{i_4,i_1} + 2x_{i_2,i_1} + 2x_{i_4,i_2} \leq 4.
\]
Last of all, when \( x_{i_3,i_1} = 1 \), \( x_{i_3,i_4} = x_{i_4,i_1} = x_{i_2,i_1} = 0 \), and we cannot have \((i_4, i_2), (i_2, i_3), (i_3, i_1)\) at once, hence we can lift in \( x_{i_3,i_1} \) with a coefficient of 2 as well. Therefore, we obtain:
\[
x_{i_1,i_2} + x_{i_2,i_3} + x_{i_3,i_4} + 3x_{i_4,i_1} + 2x_{i_2,i_1} + 2x_{i_4,i_2} + 2x_{i_3,i_1} \leq 4.
\]
\[\square\]

**Lemma 6.** Given any distinct vertices \( i_\alpha, i_\beta \in V \), the constraint
\[
x(i_\alpha, S_1) + x_{i_\alpha,i_\beta} + x_{i_\beta,i_\alpha} + x(S_2, i_\beta) \leq 2, \quad (22)
\]
for any \( S_1 \subset V, 1 \leq |S| \leq |V| - 3 \), \( S_2 \subset V \setminus \{S_1 \cup \{i_\alpha, i_\beta\}\} \), and \( 1 \leq |S_2| \leq |V| - |S_1| - 2 \), is valid for \( P_{n,3} \).
Obviously no more than two arcs among \((i_\alpha, S_1), (i_\beta, i_\alpha),\) and \((S_2, i_\beta)\) can be used, and, if \(x_{i_\alpha, i_\beta} = 1,\) \(x(i_\alpha, S_1) + x(S_2, i_\beta) = 0.\)

**Lemma 7.** Given any distinct vertices \(i_\alpha, i_\beta, i_\gamma \in V,\) the constraint

\[
x(i_\alpha, S_1) + 2x_{i_\alpha, i_\beta} + 2x_{i_\beta, i_\alpha} + x(S_2, i_\gamma) + x_{i_\alpha, i_\gamma} + x_{i_\beta, i_\gamma} \leq 4 \tag{23}
\]

for any \(S_1 \subset V, 1 \leq |S| \leq |V| - 3, S_2 \subset V \setminus \{S_1 \cup \{i_\alpha, i_\beta, i_\gamma\}\},\) and \(1 \leq |S_2| \leq |V| - |S_1| - 2,\) is valid for \(P_{n,3}.\)

We begin with adding the two trivial constraints \(x(i_\alpha, S_1) + x_{i_\gamma, i_\beta} + x_{i_\beta, i_\alpha} \leq 2\) and \(x_{i_\gamma, i_\beta} + x_{i_\beta, i_\alpha} + x(S_2, i_\gamma) \leq 2.\) When \(x_{i_\alpha, i_\beta} = 1,\) we have \(x_{i_\gamma, i_\beta} = x(i_\alpha, S_1) = 0,\) hence it can be lifted in with a coefficient of 1. Now, when \(x_{i_\beta, i_\gamma} = 1,\) we have \(x_{i_\beta, i_\alpha} = x(S_2, i_\gamma) = 0.\) If \(x_{i_\alpha, i_\beta} = 1,\) \(x_{i_\beta, i_\gamma}\) can have a coefficient of 3, but if \(x_{i_\gamma, i_\beta} = 1,\) \(x_{i_\beta, i_\gamma}\) can only have a coefficient of 1. Hence we obtain (23).

### 3. The Cardinality Constrained Cycles and Chains Problem

The Cardinality Constrained Cycles and Chains Problem (CCCCP) is more complicated. It arose from the variation of the KEP where there exists altruistic donors. Integer programming models are proposed in Glorie et al. [3], Anderson et al. [1], and [9]. The digraph \(G = (V, A')\) is defined as follows. We have that \(V = N \cup P,\) with \(N \cap P = \emptyset, A = \{(i, j) \mid i, j \in P\},\) and \(A' = A \cup \{(i, j) \mid i \in N, j \in P\}.\) A cycle on \(G\) involves only vertices in \(P.\) A chain on \(G,\) however, always begins with a vertex in \(N,\) travels through some vertices in \(P,\) and ends with a vertex in \(P.\)

The cardinality restriction of chains is represented by \(L,\) which is generally considered to be not less than \(K.\) We use \(\Delta\) to represent the set of MVP for chains. Let: \(y_{ij} \in \{0, 1\},\) for all \((i, j) \in A',\) with \(y_{ij} = 1\) if arc \((i, j)\) is used as part of a chain, and 0 otherwise; \(x_{ij} \in \{0, 1\},\) for all \((i, j) \in A,\) with \(x_{ij} = 1\) if arc \((i, j)\) is used as part of a cycle, and 0 otherwise. The IP model for the CCCCCP is given as below.
Model 2. The exponential-size SPLIT formulation

\[
\begin{align*}
\max z &= \sum_{(i,j) \in A'} w_{ij} y_{ij} + \sum_{(i,j) \in A} w_{ij} x_{ij} \\
\text{s.t.} \\
&\quad \sum_{j : (i,j) \in A'} y_{ij} \leq 1, \quad \forall i \in N \quad (24) \\
&\quad \sum_{j : (i,j) \in A} (y_{ji} + x_{ji}) \leq 1, \quad \forall i \in P \quad (25) \\
&\quad \sum_{j : (i,j) \in A} y_{ij} \leq \sum_{j : (j,i) \in A'} y_{ji}, \quad \forall i \in P \quad (26) \\
&\quad \sum_{j : (i,j) \in A} x_{ij} = \sum_{j : (j,i) \in A} x_{ji}, \quad \forall i \in P \quad (27) \\
&\quad \sum_{j : (i,j) \in A} y_{ij} \leq |S| - 1, \quad \forall S \subset P, \ 2 \leq |S| \leq |P| - 2 \quad (28) \\
&\quad \sum_{(i,j) \in \delta} y_{ij} \leq L - 1, \quad \forall \delta \in \Delta \quad (29) \\
&\quad \sum_{(i,j) \in \pi} x_{ij} \leq K - 1, \quad \forall \pi \in \Pi. \quad (30)
\end{align*}
\]

Notice that the number of \(x\)- and \(y\)-variables are \(|P|(|P| - 1)\) and \(|N||P| + |P|(|P| - 1)\) respectively. Constraints (25) and (26) are the flow and degree constraints. Constraint (28) eliminates the formation of subtours from vertices in \(P\) induced by the \(y\) variables, and Constraints (29) and (30) eliminate the formation of MVPs in chains and cycles respectively. Notice that instead of Constraints (29) for the elimination of MVPs, some strong constraints in the context of RATSP or CVRP can be implemented, given that those constraints eliminates subtours as well. Notice also that the model is similar to that of the exponential-size SPLIT model presented in Mak-Hau [9], thought the chain size violation elimination is polynomial in size therein as a Miller-Tucker-Zemlin Miller et al. [33] type constraint is used for subtour elimination for the chain variables, however the constraints (28) used in this paper is exponential in size. For convenience of polyhedral analysis, we do not use a terminal node to represent the end of a chain, but a chain ends at a vertex in \(P\) (the set of incompatible pairs).

We now expand the incident vectors \(e_a\) to contain \(e_a^y\) and \(e_a^x\), for all \(a \in A\), as well as \(e_a^y\), for all \(a \in A' \setminus A\) to represent a vector of zeros except for the arc \(a\) being used in a chain (with superscript \(y\)) or a cycle (with superscript \(x\)).

**Proposition 2** (Dimension of the CCCCP polytope). Let \(|N| = n, \ |N| \geq 2, \ |P| = p, \ |P| \geq 2; \ and let \(Q_{n,p,K,L}\) be the CCCP Polytope defined on a directed graph where: (1) vertices in \(P\) defines a complete graph; (2) arc \((i,j)\) exists for all \(i \in N\) and all \(j \in P\); (3) \(K\) (for \(K \geq 2\) is the cardinality restriction for cycles; and (4) \(L\) (for \(L \geq 2\) is the cardinality restriction for chains. The dimension of \(Q_{n,p,K,L}\), is:

\[
\dim(Q_{n,p,K,L}) = \begin{cases} 
\binom{p}{2} + np + p(p - 1), & K = 2 \\
p^2 - 2p + 1 + np + p(p - 1), & 3 \leq K \leq n.
\end{cases}
\]

22
Proof. The original \((\mathcal{P})\) two-cycles in the case of \(K = 2\) and the \(p^2 - 2p + 1\) two- and three-cycles in the case of \(K \geq 3\), together with the null vector, are still feasible solutions for the CCCC\textsuperscript{P}. We are now required to construct at least \(np + p^2 - p\) feasible solutions that involves only chains. As there are \(n\) nodes in \(N\), each connecting to every vertices is \(P\), we have \(np\) arcs, and the corresponding incident vectors \(e^y_{s,i}\), for all \(s \in N\) and all \(i \in P\) are affinely independent. Now, we can consider paths that starts from any of the vertices in \(N\), and visits two distinct vertices in \(P\), e.g., \(i\) and \(j\), forming incident vectors \(e^y_{s,i} + e^y_{i,j}\), for any arbitrary \(s^* \in N\). We have \(p^2 - p\) such affinely independent incident vectors, hence the proposition is proved. \(\square\)

**Theorem 6.** Let \(|N| = n\), \(|P| = p\), \(|P| \geq 2\); and let \(Q_{n,p,K,L}\) be the CCCC Polytope defined on a directed graph where: (1) vertices in \(P\) defines a complete graph; (2) arc \((i,j)\) exists for all \(i \in N\) and all \(j \in P\); (3) \(K\) (for \(K \geq 2\)) is the cardinality restriction for cycles; and (4) \(L\) (for \(L \geq 2\)) is the cardinality restriction for chains. The trivial constraints \(x_a \geq 0\), for all \(a \in A\), and \(y_a \geq 0\), for all \(a \in A'\), for \(K \geq 2\) and \(L \geq 2\) are facet defining for \(Q_{n,p,K,L}\).

**Proof.** [Case \(x_a = 0\), for all \(a \in A\)]. We first modify the incident vectors obtained in Theorem 1, whilst holding \(y_a = 0\), for all \(a \in A'\). We then add the \(np\) vectors given by \(e^y_{s,i}\), for all \(s \in N\) and all \(i \in P\) and the \(p(p-1)\) vectors given by \(e^y_{s,i} + e^y_{i,j}\), for \(s^*\) arbitrarily chosen in \(N\), and \(i,j\) distinct in \(P\).

[Case \(y_a = 0\), for all \(a \in A\)]. We first modify the incident vectors obtained in Proposition 1 and set \(y_a = 0\), for all \(a \in A'\). We need \(np + p(p-1) - 1\) more affinely independent vectors. These can be obtained from the new vectors obtained in Proposition 2 except for, w.l.o.g., assuming \(a = (i^*, j^*)\), the incident vector \(e^y_{s^*,i^*} + e^y_{i^*,j^*}\).

[Case \(y_a = 0\), for all \(a \in \{(i,j) \mid i \in N, j \in P\}\)]. In this case, we obtain the incident vectors in a similar way as the previous case. Again, w.l.o.g., assume \(a = (s^*, i^*)\), we will not have the incident vector \(e^y_{s^*,i^*}\), and for all vectors in the form of \(e^y_{s^*,i^*} + e^y_{i^*,j^*}\), they will be replaced by \(e^y_{s,i} + e^y_{i,j}\), for any \(s^* \in N \setminus \{s^*\}\), as we have \(|N| \geq 2\). \(\square\)

We now examine the non-trivial constraints.

**Theorem 7.** Let \(|N| = n\), \(|P| = p\), and let \(Q_{n,p,K,L}\) be the CCCC Polytope defined on a directed graph where: (1) vertices in \(P\) defines a complete graph; (2) arc \((i,j)\) exists for all \(i \in N\) and all \(j \in P\); (3) \(K\) (for \(K \geq 2\)) is the cardinality restriction for cycles; and (4) \(L\) (for \(L \geq 2\)) is the cardinality restriction for chains. The trivial constraints \(\{2\}\) is facet-defining for \(Q_{n,p,K,L}\) under the conditions that \(|N| \geq 2\) and \(|P| \geq 3\) for \(K = 2\), and \(|P| \geq 4\) for \(K \geq 3\).

**Proof.** We need to construct \((\mathcal{P})\) \(+ np + p(p-1)\) for \(K = 2\) and \(p^2 - 2p + 1 + np + p(p-1)\) for \(K \geq 3\) affinely independent feasible vectors. First of all, \(np + p(p-1)\) of them can be obtained in the following sequence. W.l.o.g., we consider the out-degree constraint \((24)\) for the vertex \(s^* \in N\). Let \(j^*\) be an arbitrary vertex in \(P\). Bold font indicates the incident vector with that particular arc is used for the first time.

[Case 1] We have \(e^y_{s^*,j}\), one for each \(j \in P\). There are \(p\) such vectors.

[Case 2] We have \(e^y_{s^*,j} + e^y_{j,k}\), one for each arc \((j,k)\) for distinct \(j, k \in P\). There are \(p(p-1)\) such vectors.
[Case 3] We have $e_{s^*,j'}^y + e_{n,j^*}^y$, one for each $s \in N \setminus \{s^*\}$, for some $j' \in P \setminus \{j^*\}$. There are $n - 1$ such vectors.

[Case 4] We have $e_{s^*,j'}^y + e_{s,j^*}^y$, one for each $s \in N \setminus \{s^*\}$ and each $j \in P \setminus \{j^*\}$. There are $(n - 1)(p - 1)$ such vectors.

We then have the following vectors that involves both chains and cycles.

[Case 5] We have $e_{s^*,j'}^y + e_{[p_{p-1,K}]}^x$. By $e_{[p_{p-1,K}]}^x$ we mean the supporting vectors for the full polytope of $P_{p-1,K}$, in which case, there are $(\binom{p-1}{2})$ such 2-cycles for $K = 2$ and $(p-1)^2 + 2(p-1) + 1$ such 2- or 3-cycles for $K \geq 3$.

Thus, for $K = 2$, we are required to find $p - 1$ more incidence vectors that are affinely independent to the previously found ones, and for $K \geq 3$, $2p - 3$ such vectors.

[Case 6a] For $K = 2$, we have $e_{s^*,j'}^y + e_{j^*,k}^x + e_{k,j^*}^x$, one for each $k \in P \setminus \{j^*\}$, (there are $p - 1$ such vectors), with $j \in P \setminus \{j^*,k\}$.

[Case 6b] For $K \geq 3$, we have the following sub-cases, each containing a chain as well as a cycle. Let $j'$ be arbitrary in $P \setminus \{j^*\}$.

- We have $e_{s^*,j'}^y + e_{j^*,k}^x + e_{k,j^*}^x$, one for each $k \in P \setminus \{j^*,j'\}$. There are $p - 2$ such vectors.
- We have $e_{s^*,j'}^y + e_{j^*,k'}^x + e_{k',j^*}^x + e_{k,j^*}^x$, for some arbitrary $k^* \in P \setminus \{j^*,j'\}$, one for each $k \in P \setminus \{j^*,j',k^*\}$. There are $p - 3$ such vectors. (These 3-cycles are affinely independent to the 2-cycles introduced in the previous case).
- Two vectors: $e_{s^*,j'}^y + e_{j^*,k}^x + e_{k,j^*}^x + e_{k',j^*}^x$, and $e_{s^*,j'}^y + e_{j^*,k}^x + e_{j',k^*}^x + e_{k',j^*}^x$ for some arbitrary $k^* \in P \setminus \{j^*,j'\}$ and $j'' \in P \setminus \{j^*,j',k^*\}$. (These are affinely independent to the previous vectors as $j''$ was used in the chain $(s^*,j')$ therein, and never used as part of a cycle).

**Theorem 8.** Let $|N| = n$, $|P| = p$, and let $Q_{n,p,K,L}$ be the CCCC Polytope defined on a directed graph where: (1) vertices in $P$ defines a complete graph; (2) arc $(i,j)$ exists for all $i \in N$ and all $j \in P$; (3) $K$ (for $K \geq 2$) is the cardinality restriction for cycles; and (4) $L$ (for $L \geq 3$) is the cardinality restriction for chains. The degree constraint (25) is facet-defining for $Q_{n,p,K,L}$ under the conditions that $|N| \geq 2$ and $|P| \geq 3$.

**Proof.** From the result of Theorem 2, using only cycles with $i^*$ involved, we have the $(\binom{p}{2})$ (in the case of $K = 2$) and $p^2 - 2p + 1$ (in the case of $K = 3$) affinely independent vectors we need, hence all we need is $np + p(p-1)$ new affinely independent feasible incident vectors. They can be constructed in the following manner. Let $i^*,j^*$ be two distinct arbitrary vertices in $P$, and $s^*$ be any arbitrary vertex in $N$. W.l.o.g., we consider the out-degree constraint (25) from vertex $i^*$.

Notice that in the proof below, the arc presented in bold are used for the first time, hence making the vector affinely independent to all previously introduced vectors.

[Case 1] First we have solutions using a single chain and a single 2- or 3-cycle.
(1a) We have $c_{s^*, j^*}^x + c_{j^*, j^*}^x + e_{s, k}^y$ for each $s \in N$ and each $k \in P \setminus \{i^*, j^*\}$. There are $n(p - 2)$ such vectors.

(1b) We have the vector $e_{s^*, i^*}^y + c_{i^*, j^*}^x + c_{j^*, i^*}^x$, for distinct $i', j' \in P \setminus \{i^*\}$.

(1c) We have the vector $e_{s^*, j^*}^y + e_{j^*, j'}^x + c_{j', i^*}^x$, for $j' \in P \setminus \{j^*\}$.

The next four cases contains only chains.

**Case 2** We have $c_{s^*, j^*}^x + e_{i^*}^y$ for all $j \in P \setminus \{i^*\}$. There are $p - 1$ such vectors.

**Case 3** We have $c_{s^*, i^*}^x + c_{i^*, j}^y + e_{j, k}^y$ for all $j, k \in P \setminus \{i^*\}$ and $j \neq k$. There are $(p - 1)(p - 2)$ such vectors. (Notice that linear combination of Case 2 and Case 3 instances can produce 2-cycles, however they do not involve the vertex $i^*$, and therefore the linear indepenedency is preserved).

**Case 4** We have $e_{s^*, j}^y + e_{j, i^*}^y$, for each $j \in P \setminus \{i^*\}$. There are $p - 1$ such vectors.

**Case 5**

(5a) We have $e_{s^*, i^*}^y + e_{s, j^*}^y$, for each $s \in N \setminus s^*$, with an arbitrary $i' \in P \setminus \{i^*, j^*\}$. There are $n - 1$ such vectors.

(5b) We have $e_{s^*, i^*}^y + e_{s, j^*}^y$, for each $s \in N \setminus s^*$. There are $n - 1$ such vectors.

4. Preliminary numerical results

We have conducted preliminary experiments to test the strengths of the facet-defining and valid constraints presented in this paper by comparing the run time for obtaining optimal solutions. First we describe how the problem instances are generated. The instances were simulated by using the parameters described in the Anderson et al. [1] wherein 25 problem cases were presented with $|N|$, $|P|$, and number of edges $|A|$ specified. We estimated the arc density of each case using the formula: arc density = $\frac{|A|}{|N|(|P| + |P|(|P| - 1))}$. To generate data instances with a certain arc density, we give each arc a specific chance of existence, e.g., if the arc density is 14%, then a random number is generated with this probability to determine if the arc “exists”. For each of the 25 distinct problem cases, we ran the generator five times to generate five distinct instances, and all numerical results reported in this section are obtained as an average out of the five instances. Data are available for download at Full Data Set [34]. We tested the same parameter settings used in Anderson et al. [1], that is, we tested $K = 3$ and $L = \infty$ (in practice though, we have $L = |P| + 1$) in these instances. In our implementation, we simply completely enumerated all cycle-cardinality violation elimination constraints for each constraint type and incorporated them into Model 6 of Mak-Han [9]. The model is coded and solved using IBM ILOG CPLEX Optimization Studio v12.6. All tests are carried out on an iMac with a 2.93GHz Intel Core i7 processor, and 32GB RAM.

From Table 1 (wherein the original computation times are provided) and Table 2 (wherein the computation times of other columns are calculated as a fraction over that of the column for (4), hence all values in the column for (4) are “1”), we can see that the maximally cardinality feasible path constraint, (8), provided the strongest improvements on the computation time when compared with the basic minimal-cardinality violation elimination constraint (4) and the improvement is by
Table 1 A comparison of computation time for exact solutions using Model 6 of Mak-Hau [9] with different constraints for eliminating cardinality violations for cycles. We set $K = 3$ and $L = \infty$ (i.e., $|P| + 1$ in practice). The data are generated using the same arc density calculated from the Table S3 presented in Anderson et al. [1]. Computation times for solving the problems to optimality are in seconds.

| $|N|$ | $|P|$ | Arc density | (4)    | (8)   | (12)   | (13)   | (18)   | (20)   | (21)   |
|-----|-----|------------|-------|-------|--------|--------|--------|--------|--------|
| 1   | 310 | 4.64%      | 285.23| 6.156 | 466.748| 444.484| 702.61 | 141.732| 119.662|
| 3   | 199 | 6.45%      | 61.922| 2.012 | 60.898 | 65.67  | 139.66 | 29.178 | 25.896 |
| 3   | 202 | 11.42%     | 1190.3| 14.034| 1061.844| 927.522| 1974.758| 3221.788| 1557.36|
| 4   | 289 | 4.15%      | 977.2 | 3.188 | 256.568| 410.894| 363.456| 55.652 | 52.652 |
| 7   | 198 | 12.09%     | 517.422| 10.15 | 488.298| 542.11 | 1187.244| 258.734| 205.488|
| 7   | 255 | 3.59%      | 51.85 | 1.456 | 44.148 | 47.562 | 57.822 | 11.456 | 4.084  |
| 7   | 291 | 4.36%      | 201.172| 2.992 | 585.762| 240.086| 360.924| 80.476 | 63.232 |
| 8   | 275 | 4.07%      | 85.504| 2.116 | 88.84  | 91.578 | 149.406| 36.036 | 28.736 |
| 10  | 152 | 4.60%      | 4.702 | 0.326 | 4.944  | 4.774  | 6.244  | 1.152  | 0.718  |
| 10  | 156 | 4.31%      | 5.174 | 0.318 | 4.558  | 5.01   | 5.362  | 0.934  | 0.854  |
| 10  | 255 | 3.79%      | 65.456| 1.444 | 54.304 | 55.924 | 120.17 | 17.74  | 6.174  |
| 10  | 256-1| 3.55%    | 46.278| 1.356 | 49.506 | 48.994 | 70.548 | 10.128 | 4.014  |
| 10  | 256-2| 3.46%    | 46.278| 1.67  | 37.712 | 328.842| 85.834 | 8.004  | 3.604  |
| 10  | 257 | 3.60%      | 139.72 | 1.372 | 46.206 | 61.608 | 105.558| 14.588 | 5.488  |
| 11  | 257 | 3.65%      | 45.598| 1.432 | 43.976 | 43.678 | 64.498 | 12.778 | 4.602  |

a significant margin and is consistent. It is likely that the superiority of (8) in computation time is due to the fact that it only needed to completely enumerate a 3-permutation of $n$ vertices, whereas the others constraints needed to enumerate all 4-permutations. Constraints (20) and (21) also showed superiority when compared with (4), however the improvement is not as strong as that of (8). The performances of Constraints (12), (13), and (18) are inconsistent, and sometimes did worse than (4).

In Table 3, we further tested a full set of randomly generated problem instances based on all 25 problem cases described in Anderson et al. [1] using Constraint (8), this time comparing the two objectives: maximizing the total weight (priority) of the matches versus maximizing the total number of matches. Again, we used $K = 3$ and $L = \infty$ in these runs. Five distinct instances for the 25 problem cases were tested, and the average and standard deviation of the computation times are reported. Notice that for some problem cases, the variation of solution times across five distinct problem instances are rather large–this is not unusual in integer programming. We can see that the computation time for maximizing number of matches is much higher than that of maximizing weighted sum. This is due to the fact that the objective of maximizing number of matches contains a great deal of symmetry as there may exists a large number of “equivalent” solutions with the same objective value.

We further generated a set of 20 problem instances by using the Australian blood type distribu-
The data instances tested are the same as those presented in Table 1. The computation times taken to solve the problem to optimality is calculated as a proportion against the basic constraint (4), e.g., for the instance \( |N| = 1 \) and \( |P| = 310 \) the time taken to solve the instances to optimality using Constraint (8) is only 2.2% of that of using the basic constraint (4).

| \(|N|\) | \(|P|\) | Arc density | (4)  | (8)  | (12) | (13) | (18) | (20) | (21) |
|------|------|------------|------|------|------|------|------|------|------|
| 1    | 310  | 4.64%      | 1.0022 | 1.636 | 1.555 | 2.463 | 0.497 | 0.42 |
| 3    | 199  | 6.45%      | 1.032  | 0.983 | 1.061 | 2.255 | 0.471 | 0.418|
| 3    | 202  | 11.42%     | 1.012  | 0.892 | 0.779 | 1.659 | 2.707 | 1.308|
| 3    | 269  | 3.62%      | 1.006  | 0.385 | 0.316 | 0.786 | 0.048 | 0.062|
| 4    | 289  | 4.15%      | 1.003  | 0.263 | 0.42  | 0.372 | 0.057 | 0.054|
| 7    | 198  | 12.09%     | 1.02   | 0.944 | 1.048 | 2.295 | 0.5   | 0.397|
| 7    | 255  | 3.59%      | 1.028  | 0.851 | 0.917 | 1.115 | 0.221 | 0.079|
| 7    | 291  | 4.36%      | 1.015  | 2.912 | 1.193 | 1.794 | 0.4   | 0.314|
| 8    | 275  | 4.07%      | 1.025  | 1.039 | 1.071 | 1.747 | 0.421 | 0.336|
| 10   | 152  | 4.60%      | 1.069  | 1.051 | 1.015 | 1.328 | 0.245 | 0.153|
| 10   | 156  | 4.31%      | 1.069  | 0.881 | 0.968 | 1.036 | 0.181 | 0.165|
| 10   | 255  | 3.79%      | 1.022  | 0.83  | 0.854 | 1.836 | 0.271 | 0.094|
| 10   | 256-1| 3.55%      | 1.029  | 1.07  | 1.059 | 1.524 | 0.219 | 0.087|
| 10   | 256-2| 3.46%      | 1.046  | 1.039 | 0.906 | 2.365 | 0.221 | 0.099|
| 10   | 257  | 3.60%      | 1.01  | 0.331 | 0.441 | 0.755 | 0.104 | 0.039|
| 11   | 257  | 3.65%      | 1.031  | 0.964 | 0.958 | 1.414 | 0.28  | 0.101|

Data are available at Australian Blood Type [35]. Each donor/recipient is given a randomly generated blood type, with probability distribution following the blood type distribution of the Australian population using the ABO distribution by country data presented in ABOAustralia [36]. An arc will be created if the kidney of the donor of a vertex in \( P \) or an altruistic donor in \( N \) is a match for a recipient in a vertex in \( P \). We did not allow any loops, hence assuming no transplants can be carried out between the donor and the recipient in an incompatible PDP pair. The runtime for finding the optimal solution is presented in Table 4. The values for \(|N|\), and \(|P|\) are also presented therein. In general, the model is capable of solving problems with up to 10 altruistic donors and 240 PDPs in the exchange pool and provide an optimal solution for maximizing the total weighted matches. Australia is a relatively sparsely populated country, the sizes of the kidney exchange pool is not expected to be much larger than the instances tested.

5. Future research directions

To obtain better computational results, we propose to implement a Lagrangean relaxation-based branch-and-bound (BNB) method. At least for the CCMcP, when Constraint (4) are relaxed, we get an Assignment Problem (AP) relaxation at each node of the BNB tree. AP relaxations are polynomially solvable. To improve the bounds, we can dualise, e.g., the actively violated cardinality constraints (e.g., (8)) in the objective function, and optimize the Lagrangean dual by using subgradient optimization. AP relaxation also returns natural integer solutions, hence
identifying violated cardinality constraints is straightforward. As for CCCCP, more work on identifying and proving strong valid or facet-defining cuts should be conducted. As solutions that contain both cycles and chains on the same digraph have not be widely studied in the past, the exploration of various relaxation and decomposition methods is an interesting and challenging problem in its own right.

Acknowledgement To the α and the ω. And to the reviewers whose comments and suggestions have greatly improved this paper.

A special thanks To my husband Dr David Hau for his help in installing PANDA [32] on all my machines, making the results of this paper possible.

References


Table 3 Exact solution times for randomly generated problem instances using $|N|$ and $|P|$ and the arc density approximate from $|A|$ in Anderson et al. [1]. We used $K = 3$ and $L = \infty$ (i.e., $|P| + 1$ in practice) in these runs. Run times are in seconds, taken average over 5 instances for each problem case.

| $|N|$ | $|P|$ | Arc density | Weighted Sum | MaxMatches |
|------|------|-------------|--------------|------------|
|      |      | Avg. | Time (sec) | St Dev (sec) | Avg. | Time (sec) | St Dev (sec) |
| 1    | 310  | 4.64%| 6.156 | 1.2 | 207.624 | 333.14 |
| 2    | 389  | 5.50%| 17.404 | 4.62 | 1694.252 | 538.14 |
| 3    | 199  | 6.45%| 2.012 | 0.19 | 35.708 | 38.41 |
| 2    | 202  | 11.42%| 14.034 | 7.4 | 246.984 | 26.03 |
| 3    | 269  | 3.62%| 3.342 | 1.31 | 21.456 | 15.2 |
| 4    | 289  | 4.15%| 3.188 | 0.55 | 43.484 | 29.12 |
| 5    | 284  | 12.38%| 526.932 | 63.02 | 3729.684 | 2117.38 |
| 6    | 261  | 12.84%| 299.628 | 39.25 | 1234.886 | 987.73 |
| 6    | 263  | 12.68%| 333.892 | 56.86 | 1855.51 | 1007.51 |
| 6    | 312  | 13.19%| 927.31 | 185.01 | 10910.26 | 2131.53 |
| 6    | 324  | 12.36%| 911.208 | 103.13 | 14869.744 | 16810.81 |
| 6    | 328  | 12.55%| 1161.334 | 261.47 | 10513.004 | 1788.81 |
| 6    | 330  | 12.12%| 980.294 | 161.38 | 9333.478 | 1177.15 |
| 7    | 198  | 12.09%| 10.15 | 0.53 | 189.05 | 20.49 |
| 7    | 255  | 3.59%| 1.456 | 0.23 | 4.816 | 2.3 |
| 7    | 291  | 4.36%| 2.992 | 1.01 | 40.338 | 25.24 |
| 8    | 275  | 4.07%| 2.116 | 0.22 | 5.352 | 2.43 |
| 10   | 152  | 4.60%| 0.326 | 0.1 | 0.618 | 0.1 |
| 10   | 156  | 4.31%| 0.318 | 0.06 | 0.776 | 0.35 |
| 10   | 255  | 3.79%| 1.444 | 0.11 | 4.202 | 3.48 |
| 10   | 256-1| 3.55%| 1.356 | 0.33 | 5.216 | 2.55 |
| 10   | 256-2| 3.46%| 1.67 | 0.92 | 3.048 | 1.12 |
| 10   | 257  | 3.60%| 1.372 | 0.05 | 4.672 | 2.24 |
| 11   | 257  | 3.65%| 1.432 | 0.08 | 4.73 | 2.17 |
Table 4 Exact solution times for randomly generated problem instances based on the Australian blood type distribution. We used $K = 3$ and $L = \infty$, i.e., $|P| + 1$ in practice.

<table>
<thead>
<tr>
<th>Problem Instances</th>
<th>Optimal Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td>98</td>
</tr>
<tr>
<td>2</td>
<td>98</td>
</tr>
<tr>
<td>2</td>
<td>98</td>
</tr>
<tr>
<td>2</td>
<td>98</td>
</tr>
<tr>
<td>2</td>
<td>98</td>
</tr>
<tr>
<td>5</td>
<td>145</td>
</tr>
<tr>
<td>5</td>
<td>145</td>
</tr>
<tr>
<td>5</td>
<td>145</td>
</tr>
<tr>
<td>5</td>
<td>145</td>
</tr>
<tr>
<td>5</td>
<td>145</td>
</tr>
<tr>
<td>4</td>
<td>196</td>
</tr>
<tr>
<td>4</td>
<td>196</td>
</tr>
<tr>
<td>4</td>
<td>196</td>
</tr>
<tr>
<td>4</td>
<td>196</td>
</tr>
<tr>
<td>4</td>
<td>196</td>
</tr>
<tr>
<td>10</td>
<td>240</td>
</tr>
<tr>
<td>10</td>
<td>240</td>
</tr>
<tr>
<td>10</td>
<td>240</td>
</tr>
<tr>
<td>10</td>
<td>240</td>
</tr>
<tr>
<td>10</td>
<td>240</td>
</tr>
</tbody>
</table>