Approximations of the Convex Hull of Hamiltonian Cycles for Cubic Graphs

Asghar Moeini Korbekandi

School of Computer Science, Engineering and Mathematics

Faculty of Science and Engineering

Flinders University

This dissertation submitted for the degree of

Doctor of Philosophy (Mathematics)

February 2016

To my family

for unconditionally providing their love, support, guidance and encouragement.

Contents

List of Figures

List of Tables

Summary

The now classical Travelling Salesman Problem (TSP) constitutes a famous challenge for operations researchers, mathematicians and computers scientists. In particular, suppose there are N cities, and a traveling salesman is going to start from the home city, pass through all the other cities exactly once and return to the home city. Such a travel path is called a tour or a Hamiltonian cycle (HC). The distance between each pair of cities is given, and so for any tour, the tour length is the sum of distances travelled. Hence TSP can be simply thought of as the optimisation problem of identifying the tour of shortest length.

A simple case of TSP is the Hamiltonian Cycle Problem (HCP). In particular, given a graph, we are asked to determine whether it contains at least one HC or not. With respect to this property - Hamiltonicity - graphs possessing HC are called Hamiltonian, and graphs not possessing a HC are called non-Hamiltonian. Hamiltonian cycle problem is known to be an NP-Complete problem. Indeed, HCP is already NP-complete for cubic graphs, namely, undirected graphs with exactly three edges emanating from each vertex.

An inherent difficulty in studying either HCP or TSP is the discreteness of the solution space of these problems. One approach to analysing these problems in continuous and convex domains stems from the embedding of the Hamiltonian Cycle problem, on a symmetric graph, in a discounted Markov decision process. The embedding allows us to explore the space of occupational measures corresponding to that decision process.

In this thesis we consider a convex combination of a Hamiltonian cycle and its reverse. We show that this convex combination traces out an interesting "H-curve"

in the space of occupational measures. Since such an H-curve always exists in Hamiltonian graphs, its properties may help in differentiating between graphs possessing Hamiltonian cycles and those that do not. Our analysis relies on the fact that the resolvent-like matrix induced by our convex combination can be expanded in terms of finitely many powers of probability transition matrices corresponding to that Hamiltonian cycle. We derive closed form formulae for the coefficients of these powers which are reduced to expressions involving the classical Chebyshev polynomials of the second kind.

An important aim of the methods for TSP and HCP which are designed based on integer programing is to construct a good approximation for the polytope $\mathcal{Q}_c$, whose extreme points are all posible HCs. However, in most real-world TSP applications and also in the case of HCP, we deal with non-complete graphs. Therefore, we would ideally like to approximate $\mathcal{Q}(G)$, the polytope whose extreme points are all HCs that belong to a given graph G. To the best of our knowledge, there is no standard approximation for $\mathcal{Q}(G)$, and $\mathcal{Q}_c$ often plays that role. However, $\mathcal{Q}_c$ can be a very poor approximation for $\mathcal{Q}(G)$, especially when the graph G is sparse.

In this thesis we developed an approximation for $\mathcal{Q}(G)$, where G is a given cubic graph. Before constructing this approximation, we introduce a generic technique to generate unidentified equality constraints that can be used to refine feasible regions of LP-relaxations of integer programming problems. Consequently, we exploit this technique and introduce a method for identifying structural equality constraints through embedding of cubic graphs in a suitably constructed universal graph.

An indirect method of tackling HCP is by identifying which cubic graphs are non-Hamiltonian. In recent contributions other researchers developed a formulation that we call the parameter-free model. The latter characterizes a polytope containing the ideal polytope whose extreme points are Hamiltonian cycles (if any) of the given graph. Thus, if that polytope is empty, the graph is non-Hamiltonian. Unfortunately, while the polytope associated with the parameter-free model is successful at so identifying all bridge graphs and approximately 18% of non-Hamiltonian non-bridge graphs, it fails on the remaining 82% of the latter. The strength of our approximations for $\mathcal{Q}(G)$ is such that its use allows us to refine the parameter-free model for cubic graphs to achieve 100% success rate with identifying all non-Hamiltonian instances in all tested cases; in particular, when the number of vertices is 18 or less.

Declaration

I certify that this thesis does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

Asghar Moeini Korbekandi B.Sc., M.Sc.

Acknowledgements

I am very grateful to Professor Jerzy Filar for being a great friend and a fantastic supervisor (and a patient one, as well!). Working with him was a very joyful experience.

I would also like to thank my associate supervisor, Dr. Michael Haythrope. He has helped me a lot in proofreading my PhD thesis. I have also learned a lot from him about how to write good MATLAB code.

I wish to thank Dr. Ali Eshragh for being a great teacher. I have had many good teachers thus far, but he had the greatest influence. His teaching methods persuaded me to engage in research with great enthusiasm. I further wish to thank him for involving me in this interesting research project.

I am always thankful to my friends for some great moments in my life from time to time, and for sharing stories with me (vice versa), not only literally but also practically. To name a few: Leila, Ghazal, Hamed, Amir Mahdi, Davoud, Hamid, Amir, Pouya, Amin, Serguei, Mahdi, Masoud, Smaeil, Nasim, Amelia, Brianna, Alex, and Dale.

I also very much appreciate Associate Professor V. Ejov, Dr. D. Glynn, Mr. S. Rossomakhine, Mr. P. Baniasadi, Mr. K. Clancy, and Mr. A. Newcombe for their company and valuable comments made in our weekly meetings.

Last but not least are my parents. They have always supported me as I have pursued my studies. I am eternally grateful to them.

Chapter 1

Introduction

1.1 The Hamiltonian cycle problem

Combinatorial optimisation refers to problems where optimal solutions lie among a possibly very large number of discrete alternative solutions. There are many combinatorial optimization problems in literature, with the *Travelling Salesman Problem* (TSP) arguably the most famous. Although TSP is an NP-Hard problem, its appealing interpretation has attracted many researchers.

There are various versions of this problem in the literature (*e.g.*, see [2, Chapter 1]). However, in this thesis the standard form will be considered. In particular, suppose there are N cities, and a traveling salesman is going to start from the home city, pass through all the other cities exactly once and return to the home city. Such a travel path is called a *tour* or a *Hamiltonian cycle*[1] (HC). The distance between each pair of cities is given, and so for any tour, the *tour length* is the sum of distances travelled. Then TSP can be simply thought of as the problem of identifying the tour of shortest length.

A special case of TSP is the *Hamiltonian Cycle Problem* (HCP). In particular, given a graph G, we are asked to determine whether it contains at least one HC or

[1] It should be noted that terms "tour" and "Hamiltonian cycle" will be used, interchangeably.

not. With respect to this property - *Hamiltonicity* - graphs possessing HC are called Hamiltonian[2], and those graphs not possesing HC are called non-Hamiltonian. HCP is known to be an NP-Complete problem (*e.g.*, see [24]). In order to show that HCP is a special case of the TSP, we first add some auxillary arcs to the graph G to make it a complete graph. If we assign length 0 to each original arc and 1 to each auxillary arc, the resultant instance of TSP is equivalent to HCP, in the following sense. If there exists a tour with length of zero in such a complete graph, then the graph G is Hamiltonian and otherwise, it is non-Hamiltonian. Although HCP seems to be a special case of the TSP, some authors presume that the underlying difficulty of the TSP is, perhaps, hidden in HCP [18]. Hence, HCP can sometimes be considered as a "haupt problem" for solving TSP.

1.2 Background and outline

In 1994, Filar and Krass [20] embedded HCP in a perturbed Markov decision process (MDP). In this way, they converted the deterministic HCP to a particular average-reward MDP and then exploited some of the properties of such a process. That model was the motivation for a new line of research continued by a growing group of researchers around the world (*e.g.*, see [17], [8], [13], [15], [9], [14] and [29]). A review of many findings resulting from these investigations is given in [18].

In 2000, Feinberg [17] converted HCP to a special class of MDPs, the so-called, "weighted discounted MDP". In the process, he introduced $\mathcal{H}_\beta$, a new polytope corresponding to a given graph G. Among other results, he showed that if the graph G is Hamiltonian, then corresponding to each HC in G, there exists an extreme point of polytope $\mathcal{H}_\beta$. This polytope is contained in the polytope $\mathcal{X}(\beta)$ of all discounted occupational measures induced by stationary policies, where $\beta \in (0, 1)$ is the discount parameter. This model has been recently extended to a parameter-free model that

---

[2] The name stems from Sir William Hamilton's investigations of such cycles on the dodecahedron graph around 1856 but Leonhard Euler studied the famous "knight's tour" on a chessboard as early as 1759.

has a feasible region $\mathcal{P}$ (in $\mathcal{O}(N^4)$ dimensions) [6], [19]. Many other approaches to TSP have focused on properties of feasible regions $\mathcal{LF}$ resulting from LP relaxations of various *Integer Linear Programming* (ILP) formulations. There are literally hundreds of papers devoted to those approaches. The reader is referred to the excellent text Applegate et al [2] for summary of the latter.

This thesis contains three main contributions:

1. An analysis of Hamiltonian curves in the polytope $\mathcal{X}(\beta)$ and the equality constraints implied by these curves in $\mathcal{P}$.

2. A generic method to discover and generate new equality constraints that can be used to refine feasible regions $\mathcal{LF}$.

3. An embedding of cubic graphs in a suitably constructed universal graph is introduced as a useful tool in identifying new structural equality constraints.

The outline of this thesis is as follows. Chapter 2 presents a brief overview of MDPs and explains the space of discounted occupational measures corresponding to a given graph G. We then formally introduce a refined polyhedral domain for HCP, namely, $\mathcal{H}_\beta$. We also review the parameter-free model $\mathcal{P}$ introduced by Filar et al. [6], [19] and then refine this model by removing a number of redundant constraints. At the end of the chapter, we prove that only $\mathcal{O}(N^2)$ integer variables are required to ensure that the feasible region of $\mathcal{P}$ corresponds precisely to the set of HCs.

In Chapter 3 we study the embedding of the Hamiltonian Cycle problem, for undirected graphs, in a discounted Markov decision process. In particular, we consider a convex combination of a Hamiltonian cycle and its reverse. We show that this convex combination traces out an interesting "H-curve" in the space of occupational measures $\mathcal{X}(\beta)$. Since such an H-curve always exists in Hamiltonian graphs, its properties may help in differentiating between graphs possessing Hamiltonian cycles and those that do not. Our analysis relies on the fact that the resolvent-like matrix induced by our convex combination can be expanded in terms of finitely many powers of the probability transition matrix corresponding to that Hamiltonian cycle. We derive closed form formulae for the coefficients of these powers which are reduced to expressions involving the classical Chebyshev polynomials of the second kind. For regular graphs, we also define a function that is the inner product of points on the H-curve with a suitably defined center of the space of occupational measures and show that, despite the nonlinearity of the inner-product, this function can be expressed as a linear function of variables in the parameter-free model.

In Chapter 4 we consider the smallest dimension polytope containing all integer solutions of an integer programming problem. Frequently, this polytope is characterized by identifying linear equality constraints that all integer solutions must satisfy. Typically, some of these constraints are readily available but others need to be discovered by more technical means. We develop a method to obtain such equality constraints, and derive a set of new equality constraints for the parameter-free model. Subsequently, exploiting these results, some techniques are proposed to tighten several integer program formulations of both TSP and HCP. Finally, we strengthen the relaxation of a widely used TSP formulation with help of the newly discovered equality constraints, and analyse the improvement obtained by doing so.

In Chapter 5 we discuss the concept of universal graphs, and construct a new universal graph for cubic graphs. We then extract a number of new structural equality constraints by exploiting this universal graph and results from Chapter 4. We compare the performance of this new model with that of the original parameter-free model in terms of their ability to correctly identify all non-Hamiltonian graphs, and demonstrate that the new model is successful in identifying non-Hamiltonian cubic graphs up to size 18. These models use infeasibility as an indicator that a graph is non-Hamiltonian.

We conclude this thesis in Chapter 6 by summarising the results discovered, and presenting future directions for work in this line of research. The most promising of these is to construct a family of graphs which are universal for all cubic graphs, in a generalised sense. More precisely, any given cubic graph is a sub-graph of at least one of the graphs in this family. Based on this family of generalised universal graphs, we derive structural equality constraints for TSP and HCP on cubic graphs using only $\mathcal{O}(N^2)$ variables.

Chapter 2

Preliminary Results and Notation

2.1 Introduction to Markov decision process

We consider a sequential decision making problem. More precisely, there is a system that evolves randomly over time. At each time t, called a *stage*, the system is in an observable state S_t that takes values in the finite set $\mathcal{S} = \{1, \ldots, N\}$. At each stage t a decision maker looks at the state of the system and chooses an action, A_t. It is assumed that for each state $i \in \mathcal{S}$, there is a set $\mathcal{A}(i)$ that comprises a finite number of corresponding actions. If the decision maker selects action $a \in \mathcal{A}(i)$, he will receive the reward $r(i, a)$, immediately, and the state of the system changes to some state $j \in \mathcal{S}$ with respect to a stochastic transition rule. If such a rule follows a Markov transition law, that is,

$$\begin{aligned}
\Pr(S_{t+1} = j \mid S_0 = s_0, A_0 = a_0, \cdots, S_{t-1} = s_{t-1}, A_{t-1} = a_{t-1}, S_t = i, A_t = a) \\
= \Pr\left(S_{t+1} = j \mid S_t = i, A_t = a\right) \\
= p(j|i, a)\,,
\end{aligned}$$

then this sequential decision making process is called a *Markov Decision Process* (MDP). One might be interested in finding a sequence of decision rules f_t that choose actions with respect to the pair (t, S_t). Such a sequence is called *policy*. Thus, a typical policy f is made up of decision rules $f_t, t = 0, 1, 2, \ldots,$ where $f_t(i, \cdot)$ is a probability distribution function over action set $\mathcal{A}(i)$; that is,

$$\begin{cases} \displaystyle\sum_{a \in \mathcal{A}(i)} f_t(i,a) = 1, & \\ & \forall\; i \in \mathcal{S}\,. \\ f_t(i,a) \geq 0, \quad \forall\; a \in \mathcal{A}(i) & \end{cases}$$

A policy f is called *stationary* if it is independent of the stages (that is, $f_t(i,\cdot) = f_{t'}(i,\cdot)$ for all possible values of t, t' and i). A *stationary deterministic policy* is one that has a degenerate distribution, that is, $f(i,a) = f_t(i,a) \in \{0,1\}$ for all possible values of t, i and a. To simplify notation, this type of policy is denoted by $f(i)$ which determines the exact action one should choose whenever the system is in state i. Since we will only be interested in stationary policies, we will take "policy" to imply "stationary policy" throughout this thesis.

Note that every stationary policy f uniquely defines the probability transition matrix $P(f) = (p_{ij}(f))_{i,j=1}^{N,N}$ where

$$p_{ij}(f) = \sum_{a \in \mathcal{A}(i)} p(j|i,a) f(i,a), \quad i,j = 1,\ldots,N. \tag{2.1}$$

Here, for a fixed policy f, and where no confusion can arise we suppress the argument f and write $p_{ij} = p_{ij}(f)$.

Now, if f is a deterministic stationary policy and $P(f)$ a permutation matrix comprising a single cycle, then $P(f)$ will be called a Hamiltonian matrix. Note also that $P(f)$ has period N. That is $P^N(f) = I$ and $P^r(f) \neq I, \;\; r = 1,\ldots,N-1$.

One might be interested in finding an *optimal policy*, that is, a policy that optimizes a function (often called a "criterion") which aggregates a sequence of rewards. A criterion which is extensively discussed in literature (*e.g.*, see [22, Chapter 2]) is given below:

Discounted Markov Decision Process. Suppose the real-valued function v_f is defined by

$$v_f(s_0) \;\; := \;\; \sum_{t=0}^{\infty} \beta^t E_f\left[r(S_t, A_t) | S_0 = s_0\right] \,,$$

where β is a fixed discount factor chosen from $[0,1)$ and E_f denotes the expected reward value with respect to the probability transition matrix $P(f)$ induced by a policy f. Hence, the subscript f in v_f determines that the decision maker selects actions based on the policy f. The MDP associated with the optimisation problem

$$v^*(s_0) := \max_f \{v_f(s_0)\} \ , \tag{2.2}$$

is called *Discounted Markov Decision Process* (DMDP).

It can be proved that there exists a stationary policy that is optimal for the optimisation problem (2.2) (*e.g.*, see [22, Chapter 2]). Now, consider the following primal and dual linear programs:

- *Primal*

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N} \gamma_i \, y_i \\ \text{subject to} \quad & y_i \geq r(i,a) + \beta \sum_{j=1}^{N} p(j|i,a) y_j, \quad \forall \, i \in \mathcal{S}, \ a \in \mathcal{A}(i) \, . \end{aligned}$$

- *Dual*

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{N} \sum_{a \in \mathcal{A}(i)} r(i,a) x_{ia} \\ \text{subject to} \quad & \sum_{i=1}^{N} \sum_{a \in \mathcal{A}(i)} \left(\delta_{ij} - \beta p(j|i,a)\right) x_{ia} = \gamma_j, \quad \forall \, j \in \mathcal{S} & (2.3) \\ \text{and} \quad & x_{ia} \geq 0, \quad \forall \, i \in \mathcal{S}, \ a \in \mathcal{A}(i) \, , & (2.4) \end{aligned}$$

where $\gamma_i > 0$ is the probability that the process starts at state i, and δ_{ij} is the Kronecker delta, that is,

$$\delta_{ij} \quad = \quad \begin{cases} 1 \ , \text{ if } i = j \\ 0 \ , \text{ if } i \neq j \end{cases} \quad .$$

The optimal values of primal variables y_i are in fact equal to $v^*(i)$ in equation (2.2) and the optimal values of dual variables x^*_{ia} can be interpreted as the total discounted probability of being in (state-action)-pair (i, a), when an optimal policy, say f^*, is applied, that is,

$$\begin{aligned} x^*_{ia} &= \sum_{j=1}^{N} \sum_{t=0}^{\infty} \beta^t \Pr_{f^*} (S_t = i, A_t = a | S_0 = j) \Pr(S_0 = j) \\ &= \sum_{j=1}^{N} \sum_{t=0}^{\infty} \beta^t \Pr_{f^*} (S_t = i, A_t = a | S_0 = j) \, \gamma_j \, . \end{aligned}$$

Thus, one can interpret x_{ia} as the summation of total discounted times that the (state, action)-pair (i, a) is occupied. For this reason, $\boldsymbol{x} = \{x_{ia} | i \in \mathcal{S},\, a \in \mathcal{A}(i)\}$ is called a *discounted occupational measure*. Accordingly, the polytope defined by linear constraints (2.3)–(2.4) is called *the space of discounted occupational measures*. We also define $x_i := \sum_a x_{ia}$ and $\boldsymbol{x}_f(\beta) := (x_1, \ldots, x_N)$.

The following theorem is the main result that connects DMDP with linear programming. (*e.g.*, see [22, Chapter 2] and [33, Chapter 6]):

Theorem 2.1.1. *The following hold:*

(i) The primal-dual linear programs mentioned above possess finite optimal solutions.

(ii) Let $\{y^*_1, \ldots, y^*_N\}$ be an optimal solution of the primal linear program, then $y^*_i = v^*(i), \;\; \forall\, i \in \mathcal{S}$.

(iii) Let the occupational measure $\boldsymbol{x^*} = \{x^*_{ia} | i \in \mathcal{S},\; a \in \mathcal{A}(i)\}$ be an optimal solution of the dual linear program and define $x^*_i := \sum_{a \in \mathcal{A}(i)} x^*_{ia}$ for each $i \in \mathcal{S}$; then $x^*_i > 0$, and the stationary policy f^* defined by

$$f^*(i, a) := \frac{x^*_{ia}}{x^*_i}, \quad \forall\, i \in \mathcal{S},\; a \in \mathcal{A}(i),$$

is an optimal stationary policy for the related DMDP optimization problem proposed in (2.2).

There are extensive discussions of MDPs in [27] and [33]. The book [22] could be considered as a comprehensive manuscript for stochastic games, a competitive form of MDPs, but it also deals with MDPs. Although only one chapter in [22] deals mainly with MDPs, the connections with linear programming are discussed there in a concise way.

2.2 Hamiltonian cycles via controlled Markov chains

An important contribution stemming from the research direction described in Section 1.2, is due to Feinberg [17]. As mentioned in that section, Filar and Krass [20] converted HCP to an average-reward Markov decision process and developed a new model based on the extensive theory of MDPs. In a similar way, Feinberg converted HCP to a class of DMDPs, the so-called *Weighted Discounted Markov Decision Processes* (WDMDP). He proved the following theorem which shows that HCP can be reduced to finding a feasible deterministic policy for a DMDP with constraints. In the upcoming theorem, we will use the following terminology and notation.

Directed graph G. Let G be a directed graph on N ($N > 3$) vertices with no self-loops. Suppose $\mathcal{V} = \{1, 2, \ldots, N\}$ is the set of all vertices and $\mathcal{A}$ is the set of all arcs in this graph. For each vertex i, we can define two subsets $\mathcal{A}(i) = \{a \in \mathcal{V} \,|\, (i, a) \in \mathcal{A}\}$ and $\mathcal{B}(i) = \{b \in \mathcal{V} \,|\, (b, i) \in \mathcal{A}\}$.

Theorem 2.2.1. *[17] Consider the following linear constraints corresponding to the graph described above:*

$$\sum_{a \in \mathcal{A}(i)} x_{ia} - \beta \sum_{b \in \mathcal{B}(i)} x_{bi} = \delta_{i1}, \quad \forall\, i \in \mathcal{V} \tag{2.5}$$

$$\sum_{a \in \mathcal{A}(1)} x_{1a} = \frac{1}{1 - \beta^N} \tag{2.6}$$

$$x_{ia} \geq 0, \quad \forall\, i \in \mathcal{V},\ a \in \mathcal{A}(i), \tag{2.7}$$

where β is a fixed discount factor chosen arbitrarily from interval $(0, 1)$. The graph G is Hamiltonian if and only if the linear constraints (2.5)–(2.7) have at least one feasible solution $\boldsymbol{x}$ corresponding to a deterministic policy, a so-called "Hamiltonian *policy". That is, if for each $i \in \mathcal{V}$, there is exactly one $a \in \mathcal{A}(i)$ such that, $x_{ia} > 0$. In other words,*

$$f(i,a) := \frac{x_{ia}}{\sum\limits_{j \in \mathcal{A}(i)} x_{ij}} \in \{0,1\}, \quad \forall\, i \in \mathcal{V},\ a \in \mathcal{A}(i)\,.$$

If such a solution is found, then the corresponding deterministic policy f traces out a Hamiltonian cycle in G.

Example 2.2.2. *We illustrate Constraints* (2.5)–(2.7) *for the following given graph:*

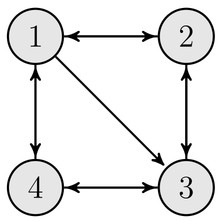

Figure 2.1: A directed 4-vertex Hamiltonian graph

$$\begin{pmatrix} 1 & 1 & 1 & -\beta & 0 & 0 & 0 & -\beta & 0 \\ -\beta & 0 & 0 & 1 & 1 & -\beta & 0 & 0 & 0 \\ 0 & -\beta & 0 & 0 & -\beta & 1 & 1 & 0 & -\beta \\ 0 & 0 & -\beta & 0 & 0 & 0 & -\beta & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_{12} \\ x_{13} \\ x_{14} \\ x_{21} \\ x_{23} \\ x_{32} \\ x_{34} \\ x_{41} \\ x_{43} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \frac{1}{1-\beta^4} \end{pmatrix}$$

$$x_{ia} \geq 0 \ \ ; \quad i = 1,2,3,4\ , \quad a \in \mathcal{A}(i)$$

It is easy to show that

$$\boldsymbol{x}^T \ = \ \frac{1}{1-\beta^4}(1,0,0,0,\beta,0,\beta^2,\beta^3,0).$$

is a feasible solution of above constraints. Moreover, this solution is associated with the following Hamiltonian policy

$$\begin{cases} f(1,2) = 1\,,\ f(2,3) = 1\,,\ f(3,4) = 1\,,\ f(4,1) = 1 \\ f(1,3) = f(1,4) = f(2,1) = f(3,2) = f(4,3) = 0 \end{cases} \quad ,$$

which traces out the tour "$1 \to 2 \to 3 \to 4 \to 1$".

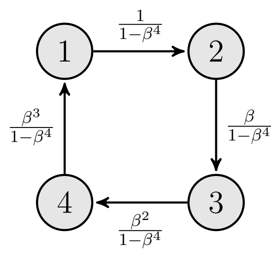

Figure 2.2: The values of entries of the vector $\boldsymbol{x}$ on the above graph

The interesting point is that these values of vertices of a feasible vector of (2.5)–(2.7) *can be achieved only for Hamiltonian cycles. Hence, this may clarify the crucial role of Constraint* (2.6)*.*

One important matrix in the context of MDPs is β*-resolvent* matrix where β is the discount factor. This matrix, for a policy f, is defined as follows:

$$R(\beta) \quad := \quad (I - \beta P(f))^{-1}\,. \tag{2.8}$$

where $P(f)$ is the probability transition matrix induced by a policy f.

The following proposition provided in [6] shows $R(\beta)$ is equal to a finite sum of powers of matrix P when P is a Hamiltonian transition matrix. For completeness we supply that proof here.

Proposition 2.2.3. *[6] If P is a Hamiltonian transition matrix, then for any value of discount factor $\beta \in (0,1)$, the following holds:*

$$R(\beta) = \frac{1}{1-\beta^N} \sum_{r=0}^{N-1} \beta^r P^r \, .$$

Proof. By exploiting the fact that N is the smallest power such that $P^N = I$, we have

$$\begin{aligned} R(\beta) = \sum_{i=0}^{\infty} \beta^i P^i \; &= \; \sum_{d=0}^{\infty} \sum_{r=0}^{N-1} \beta^{dN+r} P^{dN+r} \\ = \sum_{r=0}^{N-1} \sum_{d=0}^{\infty} \beta^{dN+r} P^r &= \; \sum_{r=0}^{N-1} P^r \sum_{d=0}^{\infty} \beta^{dN+r} \\ = \sum_{r=0}^{N-1} P^r \frac{\beta^r}{1-\beta^N} \; &= \; \frac{1}{1-\beta^N} \sum_{r=0}^{N-1} \beta^r P^r \, . \end{aligned}$$

$\blacksquare$

2.3 Parameter-free polytope

As mentioned in Section 2.2, corresponding to each given graph G, we can construct a DMDP. It is well known that we can associate the following system of linear equations with each stationary policy f in DMDP (*e.g.*, see [22, Chapter 2] and [33, Chapter 6]):

$$\boldsymbol{x}_f(\beta)^T (I - \beta P(f)) := \boldsymbol{\gamma}^T \, , \tag{2.9}$$

or, equivalently

$$\sum_{i=1}^{N} x_i(\delta_{ij} - \beta p(j|i,a)) = \gamma_j, \;\; j = 1, \ldots, N \, , \tag{2.10}$$

where $\boldsymbol{\gamma}$ is the initial distribution. Accordingly, by assuming $\boldsymbol{x}_f(\beta) > \mathbf{0}$ and defining $x_{ia} \; := \; x_i \, f(i,a)$, we can rewrite the left hand side of (2.10) with the help of (2.2) as

$$\begin{aligned}
\sum_{i=1}^{N} x_i(\delta_{ij} - \beta\ p(j|i,a)) &= \sum_{i=1}^{N} x_i \sum_{a\in\mathcal{A}(i)} (\delta_{ij} - \beta\ p(j|i,a))f(i,a) \\
&= \sum_{i=1}^{N} x_i \sum_{a\in\mathcal{A}(i)} (\delta_{ij} - \beta\ p(j|i,a))\frac{x_{ia}}{x_i} \\
&= \sum_{i=1}^{N} \sum_{a\in\mathcal{A}(i)} (\delta_{ij} - \beta\ p(j|i,a))x_{ia} \\
&= \sum_{a\in\mathcal{A}(j)} x_{ja} - \beta \sum_{b\in\mathcal{B}(j)} x_{bj}.
\end{aligned}$$

In the above we had assumed that $x_i > 0$ for every i. In fact this need not be the case in general but it always holds when f is an HC. Thus, (2.10) is simplified as below:

$$\sum_{a\in\mathcal{A}(j)} x_{ja} - \beta \sum_{b\in\mathcal{B}(j)} x_{bj} = \gamma_j,\ j = 1,\dots,N\,. \tag{2.11}$$

Clearly, if we set the initial distribution $\boldsymbol{\gamma}$ to a degenerate distribution on vertex 1, constraints (2.5) will coincide with (2.11).

Suppose that the Hamiltonian graph G on N vertices is given and P is a probability transition matrix that is a Hamiltonian matrix. Similarly to Feinberg's approach [17] to derive constraints (2.5)–(2.7), we can define the following set of constraints:

$$\boldsymbol{x}^k(\beta)^T(I - \beta P) := (1-\beta^N)\boldsymbol{e}_k^T,\ k = 1,\dots,N\,, \tag{2.12}$$

where $\boldsymbol{e}_k$ is a column vector with a unit in the k^{th} element, and zeros elsewhere. From (2.7) and Proposition 2.2.3, equation (2.12) can be rewritten as

$$\boldsymbol{x}^k(\beta)^T = (1-\beta^N)\boldsymbol{e}_k^T R(\beta) \ = \ \boldsymbol{e}_k^T I + \beta\boldsymbol{e}_k^T P + \beta^2\boldsymbol{e}_k^T P^2 + \cdots + \beta^{N-1}\boldsymbol{e}_k^T P^{N-1}\,.$$

Now, one may define new vectors $(\boldsymbol{x}_r^k)^T := \boldsymbol{e}_k^T P^r$ for $r = 0,1,2,\dots,N-1$. Since P is a permutation matrix, all components of vector $\boldsymbol{x}_r^k$ will be 0 except for single entry which will be equal to 1. This unique element identifies the r^{th} vertex visited on a Hamiltonian cycle starting from vertex k. Hence, we can rewrite $\boldsymbol{x}^k(\beta)$ in terms of vectors $\boldsymbol{x}_r^k$ as follows:

$$\boldsymbol{x}^k(\beta) = \boldsymbol{x}_0^k + \beta\boldsymbol{x}_1^k + \beta^2\boldsymbol{x}_2^k + \cdots + \beta^{N-1}\boldsymbol{x}_{N-1}^k\,. \tag{2.13}$$

Example 2.3.1. *In the case of the Hamiltonian cycle* $1 \to 2 \to 3 \to 4 \to 1$ *in example 2.2.2, the corresponding Hamiltonian policy* f *induces the Hamiltonian matrix*

$$P = P(f) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

and

$$R(\beta) = \frac{1}{1-\beta^4} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} + \frac{\beta}{1-\beta^4} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$+ \frac{\beta^2}{1-\beta^4} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} + \frac{\beta^3}{1-\beta^4} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Now it is easy to verify that (2.13) *corresponds to*

$$\boldsymbol{x}^k(\beta) = \frac{1}{1-\beta^4}(1,0,0,0,0,0,0,0,0) + \frac{\beta}{1-\beta^4}(0,0,0,0,1,0,0,0,0) +$$

$$\frac{\beta^2}{1-\beta^4}(0,0,0,0,0,0,1,0,0) + \frac{\beta^3}{1-\beta^4}(0,0,0,0,0,0,0,1,0).$$

It is readily seen that (2.13) supports the generic structure of solutions corresponding to Hamiltonian policies given in Theorem 2.2.1. Next, we consider constraints (2.12) with the right side of (2.13) in place of $\boldsymbol{x}^k(\beta)$ to obtain

$$(\boldsymbol{x}_0^k + \beta \boldsymbol{x}_1^k + \beta^2 \boldsymbol{x}_2^k + \cdots + \beta^{N-1} \boldsymbol{x}_{N-1}^k)^T (I - \beta P) := (1-\beta^N)\boldsymbol{e}_k^T, \quad k = 1, \ldots, N.$$

Or, equivalently,

$$\begin{aligned} &\boldsymbol{x}_0^k + \beta(\boldsymbol{x}_1^k - P^T \boldsymbol{x}_0^k) + \beta^2(\boldsymbol{x}_2^k - P^T \boldsymbol{x}_1^k) + \cdots \\ &\quad + \beta^{N-1}(\boldsymbol{x}_{N-1}^k - P^T \boldsymbol{x}_{N-2}^k) - \beta^N P^T \boldsymbol{x}_{N-1}^k = \boldsymbol{e}_k - \beta^N \boldsymbol{e}_k. \end{aligned} \tag{2.14}$$

Now, by equating coefficients of powers of β in both sides of (2.14), we obtain the following set of linear constraints which is free of the parameter β:

$$\boldsymbol{x}_r^k - P^T \boldsymbol{x}_{r-1}^k = 0, \quad r = 2, \ldots, N-1. \tag{2.15}$$

Now the above equation can be expanded as follows:

$$\sum_{a \in \mathcal{A}(i)} x_{r,ia}^k \quad = \quad \sum_{b \in \mathcal{B}(i)} x_{r-1,bi}^k, \ i,k = 1, \ldots, N, \ r = 1, \ldots, N-1. \tag{2.16}$$

Based on the way we have defined vectors $\boldsymbol{x}_r^k$, we now have an interesting interpretation for components $x_{r,ia}^k$. More precisely, if these vectors indeed came from a Hamiltonian transition matrix P, then we may demand that $x_{r,ia}^k$ be equal to 1 if arc (i,a) is visited at the r^{th} step of a Hamiltonian cycle starting from vertex k, and otherwise be equal to 0. New constraints which can be derived by the use of this interpretation are listed below. (see [6])

(i) If a vertex i is visited at r^{th} step of a Hamiltonian cycle starting from vertex k, then on the same tour, vertex k should be visited at $N-r$ steps starting from vertex i:

$$\sum_{a \in \mathcal{A}(i)} x_{r,ia}^k = \sum_{a \in \mathcal{A}(k)} x_{N-r,ka}^i, \ i,k = 1, \ldots, N, \ i \neq k, r = 1, \ldots, N-1\,; \tag{2.17}$$

(ii) If an arc (i,a) belongs to a particular Hamiltonian cycle, then it should be visited from any starting vertex on that tour, in particular from starting vertices k and j:

$$\sum_{r=0}^{N-1} x_{r,ia}^k = \sum_{r=0}^{N-1} x_{r,ia}^j, i,k,j = 1, \ \ldots, N, \ j < k, \ a \in \mathcal{A}(i)\,; \tag{2.18}$$

(iii) If an arc (i,a) belongs to a particular Hamiltonian cycle, then arc (i,a) will be traversed in any given step $0, \ldots, N-1$ for precisely one starting vertex. $0, 1, \ldots, N-1$ on that tour:

$$\sum_{k=1}^{N} x_{r,ia}^k = \sum_{k=1}^{N} x_{t,ia}^k, \ i = 1, \ldots, N, \ t = 0, \ldots, N-1, \ a \in \mathcal{A}(i)\,; \tag{2.19}$$

(iv) Starting from vertex k, we must visit vertex i at some step in the next $N-1$ steps:

$$\sum_{r=0}^{N-1} \sum_{a \in \mathcal{A}(i)} x_{r,ia}^k = 1,\; i, k = 1, 2, \ldots, N\,; \tag{2.20}$$

(v) Starting from vertex k, we must visit exactly one vertex at r^{th} step:

$$\sum_{i=1}^{N} \sum_{a \in \mathcal{A}(i)} x_{r,ia}^k = 1,\; k = 1, \ldots, N,\; r = 0, 1, \ldots, N-1\,; \tag{2.21}$$

(vi) The probability that a vertex is departed at the very start of a Hamiltonian cycle starting at a different vertex is 0:

$$x_{0,ia}^k = 0,\; i, k = 1, \ldots, N,\; i \neq k,\; a \in \mathcal{A}(i)\,. \tag{2.22}$$

(vii) Finally, all variables should be non-negative:

$$x_{r,ia}^k \geq 0,\; i, k = 1, \ldots, N,\; r = 0, 1, \ldots, N-1,\; a \in \mathcal{A}(i)\,. \tag{2.23}$$

2.4 Removing redundancies of parameter-free model

Let $\mathcal{P}$ denote the parameter-free model constructed by the constraints (2.16)–(2.23). We shall also use $\mathcal{P}$ to denote the polytope corresponding to the feasible region of that model. This polytope was comprehensively discussed in [6], and [19]. However, we will show that this model includes a number of redundancies. We will see that it is beneficial to remove some of these redundancies. To write the constraints of the parameter-free model for a given graph G, we can begin by constructing the parameter-free polytope $\mathcal{P}_c$ associated with a complete graph. Then we remove variables corresponding to the edges present in the complete graph but not in the given graph G. This means that a redundant constraint for the complete graph is redundant for any given graph G as well.

There are many variables which always have zero value in any HC. One kind of these is mentioned in the constraints (2.22), but there are four other kinds. We extend constraints (2.22) as follows.

$$x^k_{r,ia} = 0 \quad \begin{cases} i \neq k; r = 0 \\ a \neq k; r = N-1 \\ i = k; r \neq 0 \\ a = k; r \neq N-1 \\ r < \mu^k_i \end{cases} \tag{2.24}$$

One should note that μ^k_i in constraints (2.24) is the shortest path between vertices k and i in terms of the number of steps required to reach i from k. Obviously the last branch of (2.24) arises because a tour cannot begin at vertex k and reach vertex i in fewer than μ^k_i steps. When $k = i$, constraints (2.16) are equivalent to $0 = 0$ by constraints (2.24). Therefore, we only need to consider these constraints for $k \neq i$.

In the set of constraints (2.17) we only need to consider $k < i$, because for $k > i$, each of these constraints coincides with the analogous $k < i$ constraint when r is replaced by $N - r$. When $k = i$, constraints (2.17) are equivalent to $0 = 0$ by constraints (2.24).

Since constraints (2.18) (respectively, (2.19)) hold for all pairs of indices $j \neq k$ (respectively, $r \neq t$) it is sufficient to fix one of these indices. In particular, it is sufficient to set $j = 1$ (respectively, $t = 0$).

Lemma 2.4.1. *If the following single constraint is added to* (2.16)–(2.24)

$$\sum_{k=1}^{N} \sum_{r=0}^{N-1} \sum_{i=1}^{N} \sum_{a \in \mathcal{A}(i)} x^k_{r,ia} = N^2, \tag{I}$$

then it follows that constraints (2.20) *and* (2.21) *become redundant.*

Proof.

(i) First we show that constraints (2.20) are redundant. Recall constraint (2.18) has been replaced by $\sum_{r=0}^{N-1} x^1_{r,ia} = \sum_{r=0}^{N-1} x^k_{r,ia}, \ \forall \ k, i, a$. By taking the sum of the latter over i and a we have

$$\sum_{i=1}^{N} \sum_{a\in\mathcal{A}(i)} \sum_{r=0}^{N-1} x_{r,ia}^{1} = \sum_{i=1}^{N} \sum_{a\in\mathcal{A}(i)} \sum_{r=0}^{N-1} x_{r,ia}^{k}, \quad \forall \;\; k.$$

Therefore we can write $\sum_{i=1}^{N} \sum_{a\in\mathcal{A}(i)} \sum_{r=0}^{N-1} x_{r,ia}^{k} = c, \; \forall \, k$, where c is a constant. By substituting, the above equation in the constraint (I) we have

$$\sum_{k=1}^{N} \left(\sum_{r=0}^{N-1} \sum_{i=1}^{N} \sum_{a\in\mathcal{A}(i)} x_{r,ia}^{k} \right) \quad = \quad \sum_{k=1}^{N} c = N^2.$$

That is, $c = N$ and we can write

$$\sum_{r=0}^{N-1} \sum_{i=1}^{N} \sum_{a\in\mathcal{A}(i)} x_{r,ia}^{k} = N, \;\; \forall \;\; k. \tag{I$'$}$$

If in constraints (2.18) we take the sum over a of both sides, we can write

$$\sum_{a\in\mathcal{A}(i)} \sum_{r=0}^{N-1} x_{r,ia}^{1} = \sum_{a\in\mathcal{A}(i)} \sum_{r=0}^{N-1} x_{r,ia}^{k}, \;\; \forall \;\; k, i,$$

and conclude

$$\sum_{a\in\mathcal{A}(i)} \sum_{r=0}^{N-1} x_{r,ia}^{1} = \sum_{a\in\mathcal{A}(i)} \sum_{r=0}^{N-1} x_{r,ia}^{2} = \ldots = \sum_{a\in\mathcal{A}(i)} \sum_{r=0}^{N-1} x_{r,ia}^{N} = d_i, \;\; \forall \;\; i, \tag{II}$$

where d_i depends only on i.

The constraints (2.17) are $\sum_{a\in\mathcal{A}(k)} x_{r,ia}^{k} = \sum_{a\in\mathcal{A}(i)} x_{N-r,ka}^{i}, \;\forall \; k, i, r$. By taking the sum over r of both sides of these constraints, we will have

$$\sum_{r=0}^{N-1} \sum_{a\in\mathcal{A}(i)} x_{r,ia}^{k} = \sum_{r=0}^{N-1} \sum_{a\in\mathcal{A}(k)} x_{N-r,ka}^{i} = \sum_{r=0}^{N-1} \sum_{a\in\mathcal{A}(k)} x_{r,ka}^{i}, \;\; \forall \;\; k, i,$$

where the last equality follows from the fact that the index $N - r$ takes the same set of values as the index r. Then by exploiting the above equation and (II) we arrive at

$$d_i = \sum_{r=0}^{N-1} \sum_{a\in\mathcal{A}(i)} x_{r,ia}^{k} = \sum_{r=0}^{N-1} \sum_{a\in\mathcal{A}(k)} x_{r,ka}^{i} = d_k, \;\; \forall \;\; k, i, \tag{III}$$

and as (III) can be written for any i and k, we can conclude that $d_i = d_k = d, \ \forall \ i, k$. Then (II) can be rewritten as follows

$$\sum_{r=0}^{N-1} \sum_{a \in \mathcal{A}(i)} x_{r,ia}^{k} = d, \ \ \forall \ \ k, i. \tag{II$'$}$$

Then by substituting (II$'$) into (I$'$) we have $\sum_i d = N, \ \forall \ \ k$. Therefore $d = 1$ and we can rewrite (II$'$) as follows

$$\sum_{r=0}^{N-1} \sum_{a \in \mathcal{A}(i)} x_{r,ia}^{k} = 1, \ \ \forall \ \ k, i,$$

which is equivalent to constraints (2.20). That is, we have found that constraints (2.20) are implied by (2.17),(2.18),(I). This proves that constraints (2.20) are redundant and can be removed after (I) is added.

(ii) We now can prove that the constraints (2.21) are also redundant. From the redundancy of constraints (2.20) we know that the following constraints are redundant

$$\sum_{a \in \mathcal{A}(i)} x_{0,ia}^{i} = 1, \ \ \forall \ \ i,$$

because they can be derived from constraints (2.20) by setting $i = k$ and using (2.24). Also, by taking sum over a, from both sides, of constraints (2.19) (with $t = 0$) and using (2.22) we have

$$\sum_{a \in \mathcal{A}(i)} \sum_{k=1}^{N} x_{r,ia}^{k} = \sum_{a \in \mathcal{A}(i)} x_{0,ia}^{i} = 1 \ \ \forall \ \ r, i.$$

This shows that constraints (2.21) are also redundant and should be removed from $\mathcal{P}$.

$\blacksquare$

In view of the above, the parameter-free model (2.16)-(2.23) can be replaced by the following, simpler, model defined by constraints (2.25)-(2.31) below which we shall call the refined parameter-free model and denote it by $\bar{\mathcal{P}}$:

$$\sum_{a\in\mathcal{A}(i)} x^k_{r,ia} = \sum_{a\in\mathcal{B}(i)} x^k_{r-1,ai} \qquad k \neq i = 1,\dots,N;\; r = 1,\dots,N-1 \tag{2.25}$$

$$\sum_{a\in\mathcal{A}(i)} x^k_{r,ia} = \sum_{a\in A(k)} x^i_{N-r,ka} \qquad k > i = 1,\dots,N;\; r = 1,\dots,N-1 \tag{2.26}$$

$$\sum_{r=0}^{N-1} x^k_{r,ia} = \sum_{r=0}^{N-1} x^1_{r,ia} \qquad k = 2,\dots,N;\; (i,a) \in G \tag{2.27}$$

$$x^i_{0,ia} = \sum_{k=1}^{N} x^k_{r,ia} \qquad r = 1,\dots,N-1;\; (i,a) \in G \tag{2.28}$$

$$\sum_{k=1}^{N}\sum_{r=0}^{N-1}\sum_{i=1}^{N}\sum_{a\in\mathcal{A}(i)} x^k_{r,ia} = N^2 \tag{2.29}$$

$$x^k_{r,ia} = 0 \qquad \begin{cases} r = 0; i \neq k = 1,\dots,N; (i,a) \in G \\ i = k = 1,\dots,N; r = 1,\dots,N-1; (i,a) \in G \\ a = k; r = 0,\dots,N-2; (i,a) \in G \\ r < \mu^k_i; k,i = 1,\dots,N; (i,a) \in G \end{cases} \tag{2.30}$$

$$x^k_{r,ia} \geq 0 \qquad k = 1,\dots,N;\; r = 0,\dots N-1;\; (i,a) \in G \tag{2.31}$$

We are now assuming that the underlying graph is undirected. It will be shown that if there exist a feasible point in $\bar{\mathcal{P}}$ such that every $x^k_{r,ia}$ is binary then this feasible point defines a HC in the original graph.

Note that, for cubic graphs, refined parameter-free model $\bar{\mathcal{P}}$ has $\mathcal{O}(N^2)$ constraints in $\mathcal{O}(N^3)$ variables. We are not claiming that all the redundant constraints of the parameter-free model have been eliminated, as we do not wish to sacrifice the easy interpretability of the constraints.

2.5 Only $\mathcal{O}(N^2)$ integer variables are sufficient

Often, having an Integer Program with fewer integer variables is preferable to one with more variables. One might expect that to restrict the set of feasible solutions of $\bar{\mathcal{P}}$ to be equal to the set of HCs, we would need to demand that all variables be integer-valued. However, in this section it will be proved that only $\mathcal{O}(N^2)$ variables are required to be binary.

Proposition 2.5.1. *Consider a feasible point* $\mathbf{x} = \{x^k_{r,ia}\} \in \bar{\mathcal{P}}$ *such that the initial* $\mathcal{O}(N^2)$ *variables are binary, that is,*

$$x^k_{0,ka} \in \{0,1\},\ k = 1,\dots,N,\ a \in \mathcal{A}(k).$$

Then all the other variables will be binary as well.

$$x^k_{r,ia} \in \{0,1\},\ k = 1,\dots,N,\ r = 1,\dots,N-1,\ i = 1,\dots,N,\ a \in \mathcal{A}(k).$$

Proof. Let variables $x^i_{0,ia} \in \{0,1\},\ \forall (i,a) \in G$, and recall that all other variables are non-negative and continuous. Then summing constraint (2.28) over r $(\neq 0)$ we obtain

$$(N-1)x^i_{0,ia} = \sum_{r=1}^{N-1} x^i_{0,ia} = \sum_{r=1}^{N-1}\sum_{k=1}^{N} x^k_{r,ia},\ \forall (i,a) \in G. \tag{$*$}$$

By taking sum of $(*)$ over i and a, we see that

$$\sum_{i=1}^{N}\sum_{a\in\mathcal{A}(i)} (N-1)x^i_{0,ia} - \sum_{k=1}^{N}\sum_{r=1}^{N-1}\sum_{i=1}^{N}\sum_{a\in\mathcal{A}(i)} x^k_{r,ia} = 0.$$

Next, rewrite constraint (2.29), with the help of (2.30), as

$$\sum_{i=1}^{N}\sum_{a\in\mathcal{A}(i)} x^i_{0,ia} + \sum_{k=1}^{N}\sum_{r=1}^{N-1}\sum_{i=1}^{N}\sum_{a\in\mathcal{A}(i)} x^k_{r,ia} = N^2,$$

and add to the preceding to obtain

$$N\sum_{i=1}^{N}\sum_{a\in\mathcal{A}(i)} x^i_{0,ia} = N^2,$$

or, equivalently

$$\sum_{i=1}^{N}\sum_{a\in\mathcal{A}(i)} x^i_{0,ia} = N. \tag{$\dagger$}$$

Also, from $(*)$

$$x^i_{0,ia} = \frac{1}{N-1}\sum_{r=1}^{N-1}\sum_{k=1}^{N} x^k_{r,ia}. \tag{‡}$$

From the above equation we see that

$$\text{if } x^i_{0,ia} = 0, \text{then } x^k_{r,ia} = 0,\ \forall\, k, r. \tag{§}$$

According to (†) and the hypothesis of the proposition, exactly N variables among $x^i_{0,ia}$ with $(i,a) \in G$ must have value 1. Moreover, if $x^i_{0,ia} = 1$, then from constraints (2.25) and (2.30) we see $x^i_{0,ia} = \sum_{j\in\mathcal{A}(a)} x^i_{1,aj} = 1$. Now we show that $x^i_{1,aj}$ has value one for only one $j \in A(a)$, and the rest have to be zero. By contradiction, assume two have positive value. Without loss of generality suppose $x^i_{0,ia} = x^i_{1,ab} + x^i_{1,ac}$ where $x^i_{1,ab}, x^i_{1,ac} > 0$. By exploiting constraints (2.28) we can write

$$x^a_{0,ab} = x^i_{1,ab} + \sum_{\substack{k=1\\k\neq i}}^{N} x^k_{1,ab} > 0,$$

$$x^a_{0,ac} = x^i_{1,ac} + \sum_{\substack{k=1\\k\neq i}}^{N} x^k_{1,ac} > 0.$$

Since $x^a_{0,ab}, x^a_{0,ac} \in \{0,1\}$ and the two RHSs are strictly larger than 0 by assumption, we can conclude that $x^a_{0,ab} = x^a_{0,ac} = 1$. Note that (†) and $x^i_{0,ij} \in \{0,1\}\ \forall (i,j) \in G$ ensure that exactly N variables among $x^i_{0,ij}\ \forall (i,j) \in G$ have value one and the rest have value zero. Also, since two variables $x^a_{0,ab}, x^a_{0,ac}$ among $x^a_{0,aj}, j \in A(a)$ have value one, there must exist at least a vertex, say d, from which no flow emanates. That is, $x^d_{0,dj} = 0,\ j \in A(d)$. Then by using constraints (2.25), with $r = 1$, we have

$$x^d_{0,dj} = 0 = \sum_{m\in\mathcal{A}(j)} x^d_{1,jm},\quad j \in A(d).$$

By taking sum over j, we have

$$0 = \sum_{j\in A(d)} x^d_{0,dj} = \sum_{j\in A(d)}\sum_{m\in\mathcal{A}(j)} x^d_{1,jm}.$$

Therefore by (§) we can write $x^d_{1,jm} = 0 \ \forall \ \ j, m$. We then can continue by using (2.25) as follows

$$0 = \sum_{m=1}^{N} \sum_{j=1}^{N} x^d_{1,jm} = \sum_{m=1}^{N} \sum_{i=1}^{N} x^d_{2,mi},$$

implying that $x^d_{2,mi} = 0 \ \forall m, i$. By iteratively using constraints (2.25) we can conclude that

$$x^d_{r,mj} = 0, \ \forall r, m, j. \tag{$**$}$$

Then, by setting $k = d$ in constraints (2.27) and exploiting ($**$) we have

$$\sum_{r=0}^{N-1} x^1_{r,ia} = \sum_{r=0}^{N-1} x^d_{r,ia} = 0, \ \forall \ (i, a) \in G,$$

so $x^1_{r,ia} = 0, \ \forall \ r, i, a$. Considering constraint (2.27) we see

$$\sum_{r=0}^{N-1} x^1_{r,ia} = \sum_{r=0}^{N-1} x^k_{r,ia} = 0, \ \forall (i, a) \in G, \ k = 2, \dots, N,$$

which forces $x^k_{r,ia} = 0, \ \forall \ k = 2, \dots, N, \ (i, a) \in G$, that is, $x^k_{r,ia} = 0, \ \forall \ k, r, i, a$ and this is obviously a contradiction to constraint (2.29). Therefore our assumption is wrong and exactly one arc emanates from each vertex when the solutions are traced out by $x^i_{0,ia} \in \{0, 1\} \ \forall \ (i, a) \in G$. Let $x^i_{1,ab} = 1$ and $x^i_{1,aj} = 0, \ \forall \ j \neq b \in A(a)$. It then follows that only one of $x^i_{2,bj}$ has value 1 for $j \in A(b)$ and the rest have to be zero. To prove this, by contradiction assume that two or more of them have positive values, so we have $x^i_{1,ab} = x^i_{2,bc} + x^i_{2,bd}$ with $x^i_{2,bc}, x^i_{2,bd} \geq 0$ and by using constraints (2.28) we obtain

$$x^b_{0,bc} = x^i_{2,bc} + \sum_{\substack{k=1 \\ k \neq i}}^{N} x^k_{2,bc} > 0,$$

$$x^b_{0,bd} = x^i_{2,bd} + \sum_{\substack{k=1 \\ k \neq i}}^{N} x^k_{2,bd} > 0.$$

Arguing as before, since $x^b_{0,bc}, x^b_{0,bd} \in \{0, 1\}$ and they are both strictly larger than 0, they both must be 1 and it contradicts constraints (2.29) as shown earlier.

Utilising analogous arguments guarantees that when $x^i_{0,ia} = 1$, then for each $r = 1, \dots, N-1$ exactly one term in $\sum_{j=1}^{N} \sum_{b \in \mathcal{A}(j)} x^i_{r,jb}$ takes value 1 and all other terms are equal to 0. As N elements of $x^i_{0,ia}$ $\forall(i,a) \in G$ must have value 1, we will have N^2 variables with value 1 and the rest of the variables will have zero value. This proves that all the variables are either 0 or 1. ■

Theorem 2.5.2. *Let $\mathbf{x} \in \bar{\mathcal{P}}$ be a binary feasible point. Consider the arcs $(i,a) \in G$ such that $x^i_{0,ia} = 1$. These arcs form a Hamiltonian cycle in G.*

Proof. We know from the proof of Proposition 2.5.1 that exactly N of the $x^i_{0,ia}$ variables take value 1. Also, exactly one of these, say x^i_{0,ia_i}, emanates from each vertex i. Therefore, the matrix $\mathbf{X}_0 = \left(x^i_{0,ia_i}\right)^{N,N}_{i,a_i=1}$ is a 0-1 matrix with all rows summing to 1. Recall that, by construction, $x^i_{0,ia} = 0$ whenever $(i,a) \notin G$. Hence the positive values of $\mathbf{X}_0$ trace out a sub-graph G_0 of G, consisting of all N vertices and exactly N arcs. That sub-graph can be (i) a Hamiltonian cycle, (ii) a noose cycle, or (iii) a union of disjoint cycles. A noose cycle arises when some column j of $\mathbf{X}_0$ sums to zero. Hence there exists a vertex j corresponding to column j of $\mathbf{X}_0$ such that $x^i_{0,ij} = 0 \;\forall\; i = 1,\ldots,N$. Then, (§) in the proof of Proposition 2.5.1 implies that $x^k_{r,ij} = 0, \;\forall\; k,r,i$. By taking the sum over $r(\neq 0)$ in constraints (2.25) we can write

$$0 = \sum_{r=1}^{N-1}\sum_{i=1}^{N} x^k_{r-1,ij} = \sum_{r=1}^{N-1}\sum_{i=1}^{N} x^k_{r,ji},$$

so $x^k_{r,ji} = 0, \;\forall\; k,i,r = 1,\ldots,N$ and then by taking sum over i in (‡) we have

$$\sum_{k=1}^{N}\sum_{i=1}^{N}\sum_{r=1}^{N-1} x^k_{r,ji} = (N-1)\sum_{i=1}^{N} x^j_{0,ji} = 0.$$

Therefore, $x^j_{0,ji} = 0, \;\forall\; i$, which contradicts the fact that j^{th} row of $\mathbf{X}_0$ must sum to 1. Hence G_0 cannot be a noose cycle and case (ii) cannot hold.

Next, suppose case (iii) holds. That is, G_0 is a union of two or more disjoint cycles. That is, there are p cycles $C_1, C_2, \ldots, C_p$ forming a partition of the vertices of G. Suppose now that distinct vertices j and a belong to two disjoint cycles in that partition. It then follows that $x^j_{0,ja} = 0$. Hence, by (§), it also follows that

$$x^k_{r,ja} = 0, \;\forall\; k,r. \tag{2.32}$$

Now, without loss of generality, suppose that vertex $1 \in C_1$ and $j \in C_2$. Since C_2 is traced out by positive entries of the form $x^i_{0,ia} = 1$, there exists a vertex $m \in C_2$ such that $x^j_{0,jm} = 1$. Now by (2.25) with $r = 1$ we have

$$1 = \sum_{a \in A(j)} x^j_{0,ja} = x^j_{0,jm} = \sum_{a \in A(m)} x^j_{1,ma}, \tag{2.33}$$

where the second equality follows from (2.30). Since all variables are binary the right side of (2.33) contains exactly one variable taking value one. Suppose that $x^j_{1,mz} = 1$ and $x^j_{1,ma} = 0$ if $a \neq z$. Note that z lies in C_2 because otherwise we would have contradiction to (2.32). Now, applying (2.25) as before, we obtain

$$1 = x^j_{1,mz} = \sum_{a \in A(z)} x^j_{2,za}. \tag{2.34}$$

Again, exactly one of the terms in the above summation takes value one. Suppose $x^j_{2,zl} = 1$ and $x^j_{2,za} = 0$ if $a \neq l$. Again note that $l \in C_2$ because otherwise we would have a contradiction to (2.32).

Continuing in this fashion we deduce that for $j \in C_2$ there are exactly N variables of the form

$$x^j_{r,k_r a_r} = 1 \;\; ; \;\; r = 0, \ldots, N-1, \tag{2.35}$$

where $(k_0, a_0) = (j, m), (k_1, a_1) = (m, z), (k_2, a_2) = (z, l)$ and so on. Furthermore, j and $k_r \in C_2$ for every $r = 0, 1, \ldots, N-1$. Since the same argument can be represented for any vertex, irrespective of which cycle it lies in, we have constructed a total of N^2 variables taking values one, which consistent with constraints (2.29) which also implies that all other variables take value zero. In particular, $x^j_{r,1a} = 0$ whenever j and i do not lie in the same cycle of the partition. Without loss of generality assume arc $(1, a)$ lies in C_1 and $j \in C_2$. Then applying (2.27) at that arc we have

$$\sum_{r=0}^{N-1} x^1_{r,1a} = x^1_{0,1a} + \sum_{r=1}^{N-1} x^1_{r,1a} = \sum_{r=0}^{N-1} x^j_{r,1a}. \tag{2.36}$$

Since all terms in the summation on the right side of (2.35) are 0 and since $x^1_{0,1a} = 1$, by construction, we have a contradiction. Thus, positive values of $x^i_{0,ia}$ cannot trace out disjoint cycles $C_1, C_2, \ldots, C_p$. Therefore, only case (i) where the positive variables correspond to the Hamiltonian cycles is possible. ■

The Time Dependent Traveling Salesman Problem (TD-TSP) is a generalisation of the classical TSP, where arc costs depend on their position in the tour with respect to the source vertex. Theorem 2.5.2 implies that TD-TSP which naturally has $\mathcal{O}(N^3)$ number of integral variables can be formulated as a model with $\mathcal{O}(N^2)$ integer variables, at the cost of having $\mathcal{O}(N^4)$ continuous variables.

Chapter 3

Hamiltonian Cycle Curves

In this chapter[1], we consider a convex combination of a Hamiltonian cycle and its reverse. We show that this convex combination traces out an interesting "H-curve" in the space of occupational measures. Since such an H-curve always exists for Hamiltonian graphs, its properties may help in differentiating between graphs possessing Hamiltonian cycles and those that do not. Our analysis relies on the fact that the resolvent-like matrix induced by our convex combination can be expanded in terms of finitely many powers of probability transition matrices corresponding to that Hamiltonian cycle. We derive closed form formulae for the coefficients of these powers which are reduced to expressions involving the classical Chebyshev polynomials of the second kind. For regular graphs, we also define a function that is the inner product of points on the H-curve with a suitably defined center of the space of occupational measures and show that, despite the nonlinearity of the inner-product, this function can be expressed as a linear function of auxiliary variables associated with our embedding. These results can be seen as stepping stones towards developing constraints on the space of occupational measures that may help characterise non-Hamiltonian graphs.

Thus the key mathematical object in this study is the set of discounted occupational measures arising in the theory of finite Markov decision processes consisting of

---

[1] The main results of this chapter have appeared in a journal publication [21].

vectors $\mathbf{x^k} = \{x_{ia}^k\}_{i=1,a\in\mathcal{A}(i)}^N$ belonging to the set

$$\mathcal{X}(\beta,k) = \{\mathbf{x}^k | \sum_{i=1}^N \sum_{a\in\mathcal{A}(i)} (\delta_{ij} - \beta p(j|i,a))\, x_{ia}^k = \delta_{kj},\ \forall\, j \in S:\ x_{ia}^k \geq 0,\ \forall\, i \in \mathcal{S}, a \in \mathcal{A}(i)\}\,.$$

With the home node fixed at $k = 1$, and with initial state distribution given as γ, analogous occupational measures were already introduced in Section 2.2. However, to make this chapter self-contained, we re-introduce all the necessary notation below.

Note that δ_{ij} is the Kronecker delta, k in $\mathcal{X}(\beta,k)$ denotes the initial state in the finite *state space* $\mathcal{S}$, $\mathcal{A}(i)$ denotes the finite set of actions in state i, and $\beta \in [0,1)$ is a fixed parameter, sometimes called the *discount factor*. In the MDP the transition probability $p(j|i,a)$ denotes the one-step probability of the system moving from state i to state j when action a available in state i is taken.

Recall that, in the embeddings of the graph G in the MDP denoted by $\Gamma(G)$ described in [20] and [17], the state space $\mathcal{S} = \{1,2,\ldots,N\}$ is the set of nodes of G, $\mathcal{A}(i)$ is the set of arcs emanating from node i, and the transition probabilities simplify to:

$$p(j|i,a) = \begin{cases} 1 & if\,\text{arc}\,(i,a) \in G\,\text{and}\,j = a \\ 0 & if\,\text{arc}\,(i,a) \in G\,\text{and}\,j \neq a \\ 0 & if\,\text{arc}\,(i,a) \notin G, \end{cases}$$

where the arc $(i,a) \in G$ will sometimes be denoted simply by the action a corresponding to the head of that arc.

The innovation proposed here is to introduce certain special curves in $\mathcal{X}(\beta,k)$ that correspond to policies that are convex combinations of pairs of Hamiltonian cycles. In particular, in an undirected graph each Hamiltonian cycle is accompanied by its reverse cycle and hence a convex combination of these results in a stationary policy in the MDP and a point in $\mathcal{X}(\beta,k)$. The set of all such points, correspond to all possible convex combinations, thus results in a curve in $\mathcal{X}(\beta,k)$. We call such a curve an *H-curve*. Clearly, if the original undirected graph is Hamiltonian, each

Hamiltonian cycle in the graph induces an H-curve. It will be seen that H-curves possess attractive symmetry properties and a unique stationary point of a certain inner product function that we call an *angular function.* This angular function is related to the cosine of the angle between points on the H-curve and the center of $\mathcal{X}(\beta, k)$, where the latter is defined by the occupational measure of the policy that chooses all actions available at a state with equal probabilities.

3.1 Methodology

One of the benefits of embedding a graph G in a discounted Markov decision process $\Gamma(G)$ is that it allows us to search for a Hamiltonian cycle in the convex space $\mathcal{X}(\beta, k)$ of (normalised) *discounted occupational measures* which is a polytope with a non-empty interior, thereby converting the original discrete, deterministic, static problem to a continuous, stochastic and dynamic one. Recall that $\mathcal{X}(\beta, k)$ was defined with the help of variables $x_{ia}^k(\beta)$ that can be regarded as realisations of stationary policies. In particular, a stationary policy f in $\Gamma(G)$ is defined by probabilities $f(i,a)$ denoting the probability that the controller chooses the action/arc (i,a), whenever the state/node i is visited. Of course, $\sum_{a \in A(i)} f(i,a) = 1$, for each i. A deterministic policy is simply a stationary policy where all $f(i,a)'s$ are binary. It is easy to verify that each stationary policy f uniquely identifies a Markov chain with the $N \times N$ probability transition matrix $P(f)$ whose ia^{th} entry is simply $f(i,a)$ when the arc $(i,a) \in G$ and 0 when $(i,a) \notin G$. Note that every deterministic policy identifies a unique subgraph of G identified by the non-zero entries of the zero-one transition matrix $P(f)$.

A key object in our analysis will be the resolvent-like matrix $R(\beta) := (I - \beta P(f))^{-1}$ associated with every stationary policy f. It also induces a vector $\mathbf{x}^k(f, \beta) \in \mathcal{X}(\beta, k)$ whose entries are given by

$$x_{ia}^k(f, \beta) := [R(f)]_{ki} \, f(i,a). \tag{3.1}$$

Importantly, we note that the policies f in $\Gamma(G)$ that correspond to Hamiltonian cycles in G are precisely those where $P(f)$ is a permutation matrix containing only a single ergodic class; of course, the corresponding Markov chains have period N. In such a case we shall say that the policy f is *Hamiltonian.* Recall from Section 2.2 that whenever f is Hamiltonian,

$$R(f) = \frac{1}{1-\beta^N} \sum_{r=0}^{N-1} \beta^r P^r(f). \tag{3.2}$$

The above equation suggests that when searching for Hamiltonian cycles in the space of discounted occupational measures $\mathcal{X}(\beta, k)$, it may also be possible to simultaneously search in a higher dimensional - but parameter-free - space defined by the variables

$$x^k_{r,ia}(u) := [P^r(f)]_{ki}\, f(i,a), \tag{3.3}$$

where the entries $[P^r(f)]_{ki}$ have the natural probabilistic interpretation as $r-$step probabilities of transitions from state k to state i under the policy f.

Extreme points of the polytopes related to $\mathcal{X}(\beta, k)$ have been studied extensively in [17], [14] and [16]. However, a further opportunity exists to exploit curves in $\mathcal{X}(\beta, k)$ joining extreme points of interest. In particular, if the policy h is Hamiltonian, then (in an undirected graph) there always exists another Hamiltonian policy h^{-1} which we shall call the *reverse of* h. In addition, we know that $h(i,a) = 1$ if and only if $h^{-1}(a,i) = 1$. Next, we shall consider a parametrised family σ_α of policies defined by

$$\sigma_\alpha(i,a) := \alpha h(i,a) + (1-\alpha)h^{-1}(i,a). \tag{3.4}$$

for all nodes i and arcs (i,a) and $\alpha \in [0,1]$.

In the next section we shall analyse properties of the resolvent-like matrix $R(\sigma_\alpha)$ induced by the policy σ_α as the parameter α varies from 0 to 1.

3.2 A path from a Hamiltonian cycle to its reverse

Suppose h and h^{-1} correspond to a Hamiltonian cycle and its reverse, respectively. To simplify the notation let $P := P(h)$ be the transition matrix induced by the policy

h. It is easy to check that, for the reverse Hamiltonian cycle h^{-1}, the corresponding probability transition matrix satisfies

$$P^{-1} := P(h^{-1}) = P^T(h) = P^{-1}(h),$$

where the last form is used in situations where the argument h^{-1} is suppressed.

In this section we investigate the behavior of the policy $\sigma_\alpha = \alpha\, h + (1-\alpha)\, h^{-1}$. The transition matrix $P(\sigma_\alpha)$ induced by the policy σ_α will be denoted by P_α. It should be clear that it satisfies the relation

$$P_\alpha = \alpha\, P + (1-\alpha)\, P^{-1}.$$

We note that it is possible to write the resolvent matrix of this policy as a linear combination of powers of P. We will use the Neumann expansion to obtain

$$R(\sigma_\alpha) := (I - \beta P_\alpha)^{-1} = \sum_{r=0}^{\infty} (\beta P_\alpha)^r = \sum_{r=0}^{\infty} \beta^r \left(\alpha\, P + (1-\alpha)\, P^{-1}\right)^r. \tag{3.5}$$

Remark 3.2.1. *To simplify the already complicated notation, in some symbols, dependence on the parameter β and/or the policy f is suppressed. For instance, this is the case with R_α defined above and also P and P_α.*

Now, an H-curve in the space of discounted occupational measures $\mathcal{X}(\beta) := \prod_{k=1}^{N} \mathcal{X}(\beta, k)$ is the parametrized family of vectors $x^k(\sigma_\alpha, \beta)$ for $k = 1, \ldots, N$ whose entries are defined by

$$x^k_{ia}(\alpha) := x^k_{ia}(\sigma_\alpha, \beta) = [R(\sigma_\alpha)]_{ki}\ \ \sigma_\alpha(i, a); \quad \alpha \in [0, 1], \tag{3.6}$$

for all $k, i \in S$ and $a \in A(i)$. Note that for $\alpha \in \{0, 1\}$ the above will satisfy the constraint $\sum_{a \in A(k)} x^k_{ka}(\alpha) = (1 - \beta^N)^{-1}$ introduced by Feinberg in [17]. However, for $\alpha \in (0, 1)$ this constraint will, in general, not be satisfied.

As $P^N = P^{-N} = I$, every exponent of P in (3.5), can be replaced by some integer between 0 to $N-1$. This implies that the summation can be rewritten as a finite summation involving only first $N-1$ powers of P. In particular, for some constants $c_r(\alpha, \beta)$ depending only on α, and β,

$$R(\sigma_\alpha) = \sum_{r=0}^{\infty} \beta^r \left(\alpha\, P + (1-\alpha)\, P^{-1}\right)^r = \sum_{r=0}^{N-1} c_r(\alpha, \beta)\, P^r. \tag{3.7}$$

Now the challenge is to find closed form formulae for the coefficients $c_r(\alpha, \beta)$ in (3.7). To achieve this goal we exploit some properties of determinants of tridiagonal matrices and their connections with Chebyshev polynomials.

3.2.1 Coefficients of the expansion of matrix $R(\sigma_\alpha)$

It will be seen below that in order to derive the expressions for the coefficients of the expansion in (3.7) we will need to examine the cofactor form of the inverse $R(\sigma_\alpha) = (I - \beta P_\alpha)^{-1}$ whose structure is inherited from the structure of P_α. It will be seen, later, that it is sufficient to derive these coefficients for the matrix $\bar{R}(\sigma_\alpha) = \left(I - \beta \bar{P}_\alpha\right)^{-1}$, where $\bar{P}_\alpha$ is induced by the standard Hamiltonian cycle $1 \to 2 \to \ldots \to N \to 1$. The general case can subsequently be obtained from an appropriate permutation of the standard cycle.

Now, the entries of $\bar{R}(\sigma_\alpha)$ can be computed by exploiting the special five-diagonal structure of the following matrix:

$$\bar{M}_\alpha(\beta) := I - \beta \bar{P}_\alpha = \begin{bmatrix} 1 & -\alpha\,\beta & 0 & \ldots & 0 & -(1-\alpha)\,\beta \\ -(1-\alpha)\,\beta & 1 & -\alpha\,\beta & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & -(1-\alpha)\,\beta & 1 & -\alpha\,\beta \\ -\alpha\,\beta & 0 & \ldots & 0 & -(1-\alpha)\,\beta & 1 \end{bmatrix}.$$

To simplify the notation, let $x = -\alpha\,\beta$, and $y = -(1-\alpha)\,\beta$ and rewrite $\bar{M}_\alpha(\beta)$

in the following form:

$$\bar{M}_\alpha(\beta) = \begin{bmatrix} 1 & x & 0 & \dots & 0 & y \\ y & 1 & x & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & y & 1 & x \\ x & 0 & \dots & 0 & y & 1 \end{bmatrix}. \tag{3.8}$$

While the above matrix is five-diagonal, it will be shown that in deriving formulae for its cofactors we shall need to consider determinants of specially structured tridiagonal and four-diagonal matrices obtained in the process of expanding the determinant of $\bar{M}_\alpha(\beta)$ along a selected row.

(i) Chebyshev polynomials and determinants of uniform tridiagonal matrices

It is well known (e.g., see [39]), that n-th Chebyshev polynomial $U_n(z)$ of the second kind can be represented as the determinant of the following tridiagonal matrix

$$U_n(z) = \begin{vmatrix} 2z & 1 & 0 & \dots & 0 \\ 1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \dots & 0 & 1 & 2z \end{vmatrix}. \tag{3.9}$$

We shall say that a tridiagonal matrix is *uniform* if all its diagonal elements are equal to unity, all of its superdiagonal elements are equal to some value (say x) and all of its subdiagonal elements are equal to some value (say y).

Lemma 3.2.2. *The determinant of the following uniform tridiagonal $n \times n$ matrices*

$$V_n(x,\,y) = \begin{vmatrix} 1 & x & 0 & \dots & 0 \\ y & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & x \\ 0 & \dots & 0 & y & 1 \end{vmatrix}, \quad \hat{V}_n(x,\,y) = \begin{vmatrix} 1 & \sqrt{xy} & 0 & \dots & 0 \\ \sqrt{xy} & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \sqrt{xy} \\ 0 & \dots & 0 & \sqrt{xy} & 1 \end{vmatrix} \tag{3.10}$$

coincide and can be expressed in terms of Chebyshev polynomials as

$$V_n(x,y) = \hat{V}_n(x,y) = (xy)^{n/2}\, U_n\left(\frac{1}{2\sqrt{xy}}\right) \qquad \textit{for } \; n \geq 2. \tag{3.11}$$

Proof. For $n = 2$ equation (3.11) can be verified by direct computation. Furthermore, from recursive properties of determinants of tridiagonal matrices we must have that

$$V_n(x,y) = V_{n-1}(x,y) - xyV_{n-2}(x,y), \tag{3.12}$$

where $V_{-1}(x,y) = 0, \quad V_0(x,y) = 1$. $\hat{V}_n(x,y)$ also satisfies the same recursion since $xy = \left(\sqrt{xy}\right)\left(\sqrt{xy}\right)$. The equality $V_n(x,y) = \hat{V}_n(x,y)$ now follows by induction on n.

Now multiplying and dividing each row of the matrix corresponding to the determinant $\hat{V}_n(x,y)$ by $\sqrt{xy}$ yields

$$\hat{V}_n(x,y) = (xy)^{n/2} \begin{vmatrix} (xy)^{\frac{-1}{2}} & 1 & 0 & \dots & 0 \\ 1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \dots & 0 & 1 & (xy)^{\frac{-1}{2}} \end{vmatrix} = (xy)^{n/2}\, U_n\left(\frac{1}{2\sqrt{xy}}\right). \tag{3.13}$$

■

(ii) Determinants of specially structured four-diagonal matrices

The preceding results enable us to derive a formula for determinants of certain, specially structured, four diagonal matrices. That formula will play a key role in the main result of this section.

Lemma 3.2.3. *Let $\mathcal{G}$ be an $n \times n$ four-diagonal matrix with the following structure:*

$$\mathcal{G} = \begin{bmatrix} 1 & x & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ y & \ddots & \ddots & \ddots & \ddots & & & & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \vdots \\ \vdots & \ddots & y & 1 & x & 0 & \ddots & & & \vdots \\ \vdots & & \ddots & 0 & y & 1 & x & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & & \ddots & \ddots & \ddots & \ddots & x \\ \vdots & & & & & & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 0 & y \end{bmatrix}.$$

Let k be number of times the argument y appears in the diagonal of $\mathcal{G}$, then

$$|\mathcal{G}| = y^k \, V_{n-k}\,(x,y)\,.$$

Proof. A convenient way to calculate the determinant of $\mathcal{G}$ is to use the expansion on the last row. It can be easily be checked that the minor of the single non-zero element of that row will again have a single non-zero element in its last row. This will repeat itself k times, until we arrive at a uniform tridiagonal matrix of dimension $(n-k) \times (n-k)$. Hence, iterating in this fashion we can easily obtain the relationship

$$|\mathcal{G}| = y^k \begin{vmatrix} 1 & x & 0 & \cdots & 0 \\ y & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & x \\ 0 & \cdots & 0 & y & 1 \end{vmatrix} = y^k \, V_{n-k}\,(x,y)\,. \tag{3.14}$$

■

(iii) Closed form formulae for coefficients $c_r(\alpha, \beta)$

Now we are ready to prove the following theorem that supplies correct coefficients $c_r(\alpha, \beta)$ in (3.7).

Theorem 3.2.4. *For $r = 1, \ldots, N$ the coefficients $c_r(\alpha, \beta)$ of P^r in the finite expansion $R(\sigma_\alpha) = \sum_{r=0}^{N-1} c_r(\alpha, \beta)\, P^r$ are given by:*

$$c_{r-1}(\alpha,\, \beta) = (-1)^{r+1} \; \frac{x^{r-1}\, V_{N-r}\,(x,y) + (-1)^N\, y^{N-r+1}\, V_{r-2}\,(x,y)}{V_{N-1}\,(x,y) - 2\,x\,y\, V_{N-2}\,(x,y) + \,(-1)^{N+1}\,(x^N + y^N)}, \tag{3.15}$$

where

$$V_n(x,y) = (xy)^{n/2}\, U_n\left(\frac{1}{2\sqrt{xy}}\right), \qquad n = 2, \ldots, N$$

and

$$x = -\alpha\, \beta \;\; \text{and} \;\; y = -(1-\alpha)\, \beta.$$

Here we set $V_{-1}(x,y) = 0,\; V_0(x,y) = V_1(x,y) = 1$.

Proof. Recall that $\bar{P}$ is the standard Hamiltonian cycle and $\bar{R}(\sigma_\alpha)$ is the resolvent of the convex combination matrix $\bar{P}_\alpha = \alpha\, \bar{P} + (1-\alpha)\, \bar{P}^{-1}$. Let $\bar{c}_r(\alpha, \beta)$'s denote the coefficient $c_r(\alpha,\, \beta)$ for this standard Hamiltonian cycle. At first these coefficients will be derived, and later it will be proved that they are invariant for all Hamiltonian cycles.

We use equation $\bar{R}(\sigma_\alpha) = (I - \beta \bar{P}_\alpha)^{-1} = \sum_{r=0}^{N-1} \bar{c}_r\,(\alpha, \beta)\; \bar{P}^r$ to find $\bar{c}_r(\alpha, \beta)$'s. To simplify the notation let $\bar{c}_r := \bar{c}_r(\alpha,\, \beta)$ for $r = 0, 1, \ldots N-1$. Observe that for each r, the matrix $\bar{P}^r$ has exactly one entry equal to unity and all remaining entries equal to 0 in every row. Furthermore, the unity entries in such a row appear at non-overlapping values of the index r. It is also the case that $\bar{R}(\sigma_\alpha)$ must be of the form:

$$\bar{R}(\sigma_\alpha) = \begin{bmatrix} \bar{c}_0 & \bar{c}_1 & \bar{c}_2 & \cdots & \bar{c}_{N-1} \\ \bar{c}_{N-1} & \bar{c}_0 & \bar{c}_1 & \cdots & \bar{c}_{N-2} \\ \bar{c}_{N-2} & \bar{c}_{N-1} & \bar{c}_0 & \cdots & \bar{c}_{N-3} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \bar{c}_1 & \cdots & \cdots & \bar{c}_{N-1} & \bar{c}_0 \end{bmatrix}. \tag{3.16}$$

As can be seen from (3.16) all coefficients of interest appear in every single row of $\bar{R}(\sigma_\alpha)$. Therefore, to find these coefficients it is sufficient to calculate the entries of the first row of this matrix.

Hence we need to find the first row of the inverse of $\bar{M}_\alpha(\beta) = I - \beta\bar{P}_\alpha$. We recall the cofactor form of the inverse of any non-singular matrix A, namely, $A^{-1} = \frac{1}{|A|}\Gamma_A^T$, where Γ_A is the matrix of cofactors of A. Thus to find the first row of the inverse of $\bar{M}_\alpha(\beta)$, one only needs to compute the determinant, and the cofactors of its first column. In particular, we have

$$\bar{c}_{r-1} = \bar{c}_{r-1}(\alpha,\beta) = [\left(\bar{M}_\alpha(\beta)\right)^{-1}]_{1r} = \frac{\gamma_{r1}(\bar{M}_\alpha(\beta))}{|\bar{M}_\alpha(\beta)|}, \quad r = 1,\ldots,N. \tag{3.17}$$

where $\gamma_{r1}\left(\bar{M}_\alpha(\beta)\right) = (-1)^{r+1}|\bar{M}_\alpha(\beta)|_{r1}$ and $|\bar{M}_\alpha(\beta)|_{r1}$ denotes the determinant of the $(N-1)\times(N-1)$ submatrix of $\bar{M}_\alpha(\beta)$ obtained after the r^{th} row and the 1^{st} column have been deleted.

We begin by deriving the determinant of $\bar{M}_\alpha(\beta)$. We use the expansion along the first row as follows (see (3.8)):

$$
\left|\bar{M}_\alpha(\beta)\right| = \begin{vmatrix} 1 & x & 0 & \dots & \dots & \dots & 0 \\ y & \ddots & \ddots & \ddots & & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & \ddots & x \\ 0 & \dots & \dots & \dots & 0 & y & 1 \end{vmatrix} - x \begin{vmatrix} y & x & 0 & \dots & \dots & \dots & 0 \\ 0 & 1 & \ddots & \ddots & & & \vdots \\ 0 & y & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & y & 1 & x \\ x & 0 & \dots & \dots & 0 & y & 1 \end{vmatrix}
$$

$$
+(-1)^{N+1} y \begin{vmatrix} y & 1 & x & 0 & \dots & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & \ddots & x \\ 0 & \dots & \dots & \dots & 0 & y & 1 \\ x & 0 & \dots & \dots & \dots & 0 & y \end{vmatrix} . \tag{3.18}
$$

The first term in (3.18) is a uniform tridiagonal matrix and the last two terms are four-diagonal. Hence, we expand the remaining terms further to reach either a triangular or a tridiagonal matrix. In the next step we have:

$$
\left|\bar{M}_\alpha(\beta)\right| = \begin{vmatrix} 1 & x & 0 & \dots & \dots & \dots & 0 \\ y & \ddots & \ddots & \ddots & & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & \ddots & x \\ 0 & \dots & \dots & \dots & 0 & y & 1 \end{vmatrix}
$$

$$
-x\left(y\begin{vmatrix}1 & x & 0 & \dots & \dots & 0\\ y & \ddots & \ddots & \ddots & & \vdots\\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & \ddots & 0\\ \vdots & & \ddots & \ddots & \ddots & x\\ 0 & \dots & \dots & 0 & y & 1\end{vmatrix}+(-1)^N x\begin{vmatrix}x & 0 & \dots & \dots & \dots & 0\\ 1 & \ddots & \ddots & & & \vdots\\ y & \ddots & \ddots & \ddots & & \vdots\\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & \ddots & 0\\ 0 & \dots & 0 & y & 1 & x\end{vmatrix}\right)
$$

$$
+(-1)^{N+1}y\left(y\begin{vmatrix}y & 1 & x & 0 & \dots & 0\\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & \ddots & 0\\ \vdots & & \ddots & \ddots & \ddots & x\\ \vdots & & & \ddots & \ddots & 1\\ 0 & \dots & \dots & \dots & 0 & y\end{vmatrix}+(-1)^N x\begin{vmatrix}1 & x & 0 & \dots & \dots & 0\\ y & \ddots & \ddots & \ddots & & \vdots\\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & \ddots & 0\\ \vdots & & \ddots & \ddots & \ddots & x\\ 0 & \dots & \dots & 0 & y & 1\end{vmatrix}\right).
$$

Now, using the results from Lemmata 3.2.2-3.2.3, we obtain

$$
\begin{aligned}
\left|\bar{M}_\alpha(\beta)\right| &= V_{N-1}(x,\,y) - x\left(y\,V_{N-2}(x,\,y) + (-1)^N\,x\,x^{N-2}\right)\\
&\quad +(-1)^{N+1}\,y\left(y\,y^{N-2} + (-1)^N x\,V_{N-2}(x,\,y)\right) \qquad (3.19)\\
&= V_{N-1}(x,\,y) - 2\,x\,y\,V_{N-2}\,(x,\,y) + (-1)^{N+1}\,\left(x^N + y^N\right).
\end{aligned}
$$

We next return to the numerator of the right side of equation (3.17). We shall require the cofactors of elements in the first column of $\bar{M}_\alpha(\beta)$. We note that the determinant, for $r \geq 4$, of the $(N-1)\times(N-1)$ submatrix of $\bar{M}_\alpha(\beta)$ obtained after the r^{th} row and the 1^{st} column have been deleted has the following generic, five-diagonal, structure:

$$
\begin{vmatrix}
x & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 0 & y \\
1 & x & 0 & \ddots & & & & & & \ddots & 0 \\
y & 1 & x & \ddots & \ddots & & & & & & 0 \\
0 & y & 1 & \ddots & \ddots & \ddots & & & & & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \vdots \\
\vdots & & \ddots & y & 1 & x & 0 & \ddots & & & \vdots \\
\vdots & & & \ddots & 0 & y & 1 & x & \ddots & & \vdots \\
\vdots & & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & & & & & \ddots & \ddots & \ddots & \ddots & x & 0 \\
\vdots & & & & & & \ddots & \ddots & \ddots & 1 & x \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 0 & y & 1
\end{vmatrix}.
$$

Expanding it along the first row, we obtain a sum of two terms

$$
x \left|
\begin{array}{ccccc|ccccc}
x & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
1 & \ddots & \ddots & \ddots & & & & & & \vdots \\
y & \ddots & \ddots & \ddots & \ddots & & & & & \vdots \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \vdots \\
\vdots & \ddots & y & 1 & x & 0 & \ddots & & & \vdots \\
\hline
\vdots & & \ddots & 0 & y & 1 & x & \ddots & & \vdots \\
\vdots & & & \ddots & \ddots & y & \ddots & \ddots & \ddots & \vdots \\
\vdots & & & & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
\vdots & & & & & \ddots & \ddots & \ddots & \ddots & x \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 0 & y & 1
\end{array}
\right| +
$$

$$
+(-1)^{N}\, y \left|\begin{array}{ccccc|ccccc}
1 & x & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
y & \ddots & \ddots & \ddots & \ddots & & & & & \vdots \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \vdots \\
\vdots & \ddots & \ddots & \ddots & x & \ddots & \ddots & & & \vdots \\
\vdots & & \ddots & y & 1 & x & 0 & \ddots & & \vdots \\
\hline
\vdots & & & \ddots & 0 & y & 1 & x & \ddots & \vdots \\
\vdots & & & & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
\vdots & & & & & \ddots & \ddots & \ddots & \ddots & x \\
\vdots & & & & & & \ddots & \ddots & \ddots & 1 \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 0 & y
\end{array}\right| .
$$

It is important to note that the determinants of both of the matrices in the above expansion can be partitioned into two block-diagonal submatrices of dimensions $(r-2)\times(r-2)$ and $(N-r)\times(N-r)$, respectively. However, in the first of these matrices the $(r-2)\times(r-2)$ diagonal block is a lower triangular matrix and the $(N-r)\times(N-r)$ block has the uniform tridiagonal structure. Conversely, in the second matrix of the expansion $(r-2)\times(r-2)$ diagonal block has the uniform tridiagonal structure, whereas the $(N-r)\times(N-r)$ diagonal block is upper triangular. This allows us to exploit expressions derived in Lemmata 3.2.2-3.2.3 to obtain:

$$
(-1)^{r+1}\,|\gamma_{r1}| = (-1)^{r+1}\ \left(x\, x^{r-2}\, V_{N-r}\,(x,y) + (-1)^{N}\, y\, y^{N-r}\, V_{r-2}\,(x,y)\right) =
$$

$$
(-1)^{r+1}\ \left(x^{r-1}\, V_{N-r}\,(x,y) + (-1)^{N}\, y^{N-r+1}\, V_{r-2}\,(x,y)\right),\ r = 4,\ldots,N. \tag{3.20}
$$

Interestingly, the expression (3.20) also holds for the special cases of $r = 1,2,3$, when (as before) $V_{-1}(x,y) = 0$ and $V_0(x,y) = V_1(x,y) = 1$. In particular, for $r = 1$, it is easy to see directly from (3.8) that γ_{11} is $(N-1)\times(N-1)$ uniform tridiagonal matrix and hence $(-1)^2\,|\gamma_{11}| = V_{N-1}(x,y)$.

The case $r = 2$ follows from the fact that γ_{21} is $(N-1)\times(N-1)$ four diagonal matrix with the first row having an x in its first entry, y in its last entry and zeros in all other entries. Then expanding $|\gamma_{21}|$ by its first row we obtain two terms, the first involving a uniform tridiagonal matrix and the second an upper triangular matrix. It can be checked that $(-1)^3\,|\gamma_{21}| = -(xV_{N-2}(x,y) + (-1)^N y^{N-1})$.

The case $r = 3$, follows by an analogous argument to the above and yields $(-1)^4|\gamma_{31}| = x^2 V_{N-3}(x,y) + (-1)^N y^{N-2}$.

Finally, we have all the composite parts to use in (3.17) to find coefficients $\bar{c}_r(\alpha,\,\beta)$'s by dividing (3.20) by (3.19).

$$\bar{c}_{r-1}(\alpha,\,\beta) = (-1)^{r+1}\;\frac{x^{r-1}\,V_{N-r}\,(x,y) + (-1)^N\,y^{N-r+1}\,V_{r-2}\,(x,y)}{V_{N-1}\,(x,y) - 2\,x\,y\,V_{N-2}\,(x,y) +\;(-1)^{N+1}\,(x^N + y^N)},\quad r = 1,\ldots,N.$$

■

Example 3.2.5. *To illustrate the way that the above coefficients are derived, we obtain them for $N = 5$, by using the analogous expansions that are used in the above proof.*

$$\bar{c}_{r-1} = [\big(\bar{M}_\alpha(\beta)\big)^{-1}]_{1r} = \frac{\gamma_{r1}(\bar{M}_\alpha(\beta))}{|\bar{M}_\alpha(\beta)|},\qquad r = 1,\ldots,5. \tag{3.21}$$

Note that the dominator $|\bar{M}_\alpha(\beta)|$ can be computed by expansion along first row as follows:

$$\left|\bar{M}_\alpha(\beta)\right| = \begin{vmatrix} 1 & x & 0 & 0 & y \\ y & 1 & x & 0 & 0 \\ 0 & y & 1 & x & 0 \\ 0 & 0 & y & 1 & x \\ x & 0 & 0 & y & 1 \end{vmatrix} = \begin{vmatrix} 1 & x & 0 & 0 \\ y & 1 & x & 0 \\ 0 & y & 1 & x \\ 0 & 0 & y & 1 \end{vmatrix} - x \begin{vmatrix} y & x & 0 & 0 \\ 0 & 1 & x & 0 \\ 0 & y & 1 & x \\ x & 0 & y & 1 \end{vmatrix} + y \begin{vmatrix} y & 1 & x & 0 \\ 0 & y & 1 & x \\ 0 & 0 & y & 1 \\ x & 0 & 0 & y \end{vmatrix}$$

$$= \begin{vmatrix} 1 & x & 0 & 0 \\ y & 1 & x & 0 \\ 0 & y & 1 & x \\ 0 & 0 & y & 1 \end{vmatrix}$$

$$-x\left(y\begin{vmatrix} 1 & x & 0 \\ y & 1 & x \\ 0 & y & 1 \end{vmatrix} - x\begin{vmatrix} x & 0 & 0 \\ 1 & x & 0 \\ y & 1 & x \end{vmatrix}\right) + y\left(y\begin{vmatrix} y & 1 & x \\ 0 & y & 1 \\ 0 & 0 & y \end{vmatrix} - x\begin{vmatrix} 1 & x & 0 \\ y & 1 & x \\ 0 & y & 1 \end{vmatrix}\right) =$$

$$V_4\,(x,\,y) - 2\,x\,y\,V_3\;(x,\,y) + \big(x^5 + y^5\big)\,.$$

We also can find numerators $\gamma_{r1}(\bar{M}_\alpha(\beta))$ $r = 1, \ldots, 5$ by expanding along the first row as follows:

$$\gamma_{11}(\bar{M}_\alpha(\beta)) = \begin{vmatrix} 1 & x & 0 & 0 \\ y & 1 & x & 0 \\ 0 & y & 1 & x \\ 0 & 0 & y & 1 \end{vmatrix} = V_4\left(x,\, y\right) = x^0 V_4\left(x,\, y\right) + y^5 V_{-1}\left(x,\, y\right).$$

$$\gamma_{21}(\bar{M}_\alpha(\beta)) = \begin{vmatrix} x & 0 & 0 & y \\ y & 1 & x & 0 \\ 0 & y & 1 & x \\ 0 & 0 & y & 1 \end{vmatrix} = x \begin{vmatrix} 1 & x & 0 \\ y & 1 & x \\ 0 & y & 1 \end{vmatrix} - y \begin{vmatrix} y & 1 & x \\ 0 & y & 1 \\ 0 & 0 & y \end{vmatrix} = x\ V_3\left(x,\, y\right) - y.y^3$$
$$= xV_3\left(x,\, y\right) - y^4 V_0\left(x,\, y\right).$$

$$\gamma_{31}(\bar{M}_\alpha(\beta)) = \begin{vmatrix} x & 0 & 0 & y \\ 1 & x & 0 & 0 \\ 0 & y & 1 & x \\ 0 & 0 & y & 1 \end{vmatrix} = x \begin{vmatrix} x & 0 & 0 \\ y & 1 & x \\ 0 & y & 1 \end{vmatrix} - y \begin{vmatrix} 1 & x & 0 \\ 0 & y & 1 \\ 0 & 0 & y \end{vmatrix} = x.x \begin{vmatrix} 1 & x \\ y & 1 \end{vmatrix} - y.y^2$$
$$= x.xV_2\left(x,\, y\right) - y.y^2 = x^2 V_2\left(x,\, y\right) - y^3 V_1\left(x,\, y\right).$$

$$\gamma_{41}(\bar{M}_\alpha(\beta)) = \begin{vmatrix} x & 0 & 0 & y \\ 1 & x & 0 & 0 \\ y & 1 & x & 0 \\ 0 & 0 & y & 1 \end{vmatrix} = x \begin{vmatrix} x & 0 & 0 \\ 1 & x & 0 \\ 0 & y & 1 \end{vmatrix} - y \begin{vmatrix} 1 & x & 0 \\ y & 1 & x \\ 0 & 0 & y \end{vmatrix} = x.x^2 - y.y \begin{vmatrix} 1 & x \\ y & 1 \end{vmatrix}$$
$$= x.x^2 - y.yV_2\left(x,\, y\right) = x^3 V_1\left(x,\, y\right) - y^2 V_2\left(x,\, y\right).$$

$$\gamma_{51}(\bar{M}_\alpha(\beta)) = \begin{vmatrix} x & 0 & 0 & y \\ 1 & x & 0 & 0 \\ y & 1 & x & 0 \\ 0 & y & 1 & x \end{vmatrix} = x \begin{vmatrix} x & 0 & 0 \\ 1 & x & 0 \\ y & 1 & x \end{vmatrix} - y \begin{vmatrix} 1 & x & 0 \\ y & 1 & x \\ 0 & y & 1 \end{vmatrix} = x.x^3 - y\ V_3\left(x,\, y\right)$$

$$= x^4 V_0\,(x,\,y) - y V_3\,(x,\,y)\,.$$

Then coefficients are obtained by

$$c_0(\alpha,\,\beta) = \frac{\gamma_{11}(\bar{M}_\alpha(\beta))}{\left|\bar{M}_\alpha(\beta)\right|} = \frac{V_4\,(x,\,y)}{V_4\,(x,\,y) - 2\,x\,y\,V_3\,\,(x,\,y) + (x^5+y^5)}.$$

$$c_1(\alpha,\,\beta) = \frac{\gamma_{21}(\bar{M}_\alpha(\beta))}{\left|\bar{M}_\alpha(\beta)\right|} = \frac{x\,\,V_3\,(x,\,y) - y^4}{V_4\,(x,\,y) - 2\,x\,y\,V_3\,\,(x,\,y) + (x^5+y^5)}.$$

$$c_2(\alpha,\,\beta) = \frac{\gamma_{31}(\bar{M}_\alpha(\beta))}{\left|\bar{M}_\alpha(\beta)\right|} = \frac{x^2 V_2\,(x,\,y) - y^3}{V_4\,(x,\,y) - 2\,x\,y\,V_3\,\,(x,\,y) + (x^5+y^5)}.$$

$$c_3(\alpha,\,\beta) = \frac{\gamma_{41}(\bar{M}_\alpha(\beta))}{\left|\bar{M}_\alpha(\beta)\right|} = \frac{x^3 - y^2 V_2\,(x,\,y)}{V_4\,(x,\,y) - 2\,x\,y\,V_3\,\,(x,\,y) + (x^5+y^5)}.$$

$$c_4(\alpha,\,\beta) = \frac{\gamma_{51}(\bar{M}_\alpha(\beta))}{\left|\bar{M}_\alpha(\beta)\right|} = \frac{x^4 - y\,\,V_3\,(x,\,y)}{V_4\,(x,\,y) - 2\,x\,y\,V_3\,\,(x,\,y) + (x^5+y^5)}.$$

3.2.2 Uniqueness of coefficients for all Hamiltonian cycles

Let $\bar{P}$ denote the standard Hamiltonian transition matrix, then we know that an arbitrary Hamiltonian transition matrix P can be derived via $P = Q\,\bar{P}\,Q^{-1}$, where Q is a permutation matrix. Let $\bar{M}_\alpha(\beta) := [\bar{R}(\sigma_\alpha)]^{-1} \,=\, I - \beta\left(\alpha\,\bar{P} + (1-\alpha)\,\bar{P}^{-1}\right)$ and $M_\alpha(\beta) := [R(\sigma_\alpha)]^{-1} = I - \beta\,(\alpha\,P + (1-\alpha)\,P^{-1})$. Then it follows immediately that

$$M_\alpha(\beta) = I - \beta\left(\alpha P + (1-\alpha)P^{-1}\right)$$

$$= QIQ^{-1} - \alpha\beta Q\bar{P}Q^{-1} - \beta(1-\alpha)Q\bar{P}^{-1}Q^{-1} = Q\bar{M}_\alpha(\beta)Q^{-1}.$$

Furthermore, taking powers of the relation $P = Q\,\bar{P}\,Q^{-1}$, for every $r = 0, 1, 2, \ldots, N-1$, yields

$$P^r = Q\,\bar{P}^r\,Q^{-1}.$$

However, for the standard Hamiltonian cycle, in Proposition (3.2.4) we already proved that

$$\bar{R}(\sigma_\alpha) = \sum_{r=0}^{N-1} \bar{c}_r(\alpha, \beta)\, \bar{P}^r.$$

Now, multiplying the last equation on the left by Q and on the right by Q^{-1} and exploiting the preceding two equations we obtain

$$R(\sigma_\alpha) = \sum_{r=0}^{N-1} \bar{c}_r(\alpha, \beta)\, Q\, \bar{P}^r\, Q^{-1} = \sum_{r=0}^{N-1} \bar{c}_r(\alpha, \beta) P^r = \sum_{r=0}^{N-1} c_r(\alpha, \beta) P^r. \tag{3.22}$$

Hence the coefficients $c_r(\alpha, \beta)$ in (3.7) of the expansion of $R(\sigma_\alpha)$ corresponding to an arbitrary Hamiltonian cycle, coincide with the coefficients $\bar{c}_r(\alpha, \beta)$ in the expansion of $\bar{R}(\sigma_\alpha)$ induced by the standard Hamiltonian cycle.

3.3 H-curve in the space of occupational measures

In this section we analyse properties of curves $x^k(\sigma_\alpha, \beta)$ in $\mathcal{X}(\beta, k)$ which join oriented Hamiltonian cycles to their reverse cycles as represented by their induced discounted occupational measures as defined by (3.1). Recall that these were named H-curves.

We shall also be interested in inner products of points on H-curves with the center of $\mathcal{X}(\beta, k)$ defined as $x^k(u^\dagger, \beta)$, where the center policy $u^\dagger$ is one that selects all arcs emanating from the vertex i with equal probabilities, for each $i = 1, \ldots, N$. Namely, if the cardinality of $A(i)$ (degree of node i) is m_i, then $u^\dagger(i, a) = \frac{1}{m_i}$, for each $a \in A(i)$ and every $i = 1, \ldots, N$.

We note that the point $x^k(u^\dagger, \beta)$ is well-defined by the equation

$$x^k_{ia}(\dagger) := x^k_{ia}(u^\dagger, \beta) := [R(u^\dagger)]_{ki}\, u^\dagger(i, a), \tag{3.23}$$

and lies in $\mathcal{X}(\beta, k)$, irrespective of the structure of the underlying graph G. Of course, points on an H-curve are known to exist specifically for Hamiltonian graphs. Hence, it will be interesting to investigate the inner product $x^k(\sigma_\alpha, \beta) \,.\, x^k(u^\dagger, \beta)$, $\forall \alpha \in [0, 1]$. To do this we shall need an additional symmetry property of $R(\sigma_\alpha)$.

3.3.1 Relation between $R(\sigma_\alpha)$ and $R(\sigma_{1-\alpha})$

As in the previous section, let P be the transition matrix induced by the Hamiltonian policy h and $0 < \alpha < 1$. Also let $R(\sigma_\alpha)$ denote the resolvent-like matrix of the σ_α policy defined by (3.4). Of course, $R(\sigma_{1-\alpha})$ denotes the resolvent-like matrix of the $\sigma_{1-\alpha}$ policy. Then following expressions can be easily derived

$$P_{1-\alpha}^T = P_\alpha, \tag{3.24}$$

$$R^T(\sigma_{1-\alpha}) = R(\sigma_\alpha). \tag{3.25}$$

In particular, (3.24) holds because

$$P_\alpha = \alpha\, P + (1-\alpha)\, P^T,$$

$$P_{1-\alpha}^T = \left[(1-\alpha)\, P + \alpha\, P^T\right]^T = (1-\alpha)\, P^T + \alpha\, P = P_\alpha.$$

Similarly, (3.25) holds because

$$R^T(\sigma_{1-\alpha}) = \left[(I - \beta\, P_{1-\alpha})^{-1}\right]^T = \left[(I - \beta\, P_{1-\alpha})^T\right]^{-1}.$$

The above expression can now be simplified as follows

$$R^T(\sigma_{1-\alpha}) = \left(I - \beta\, P_{1-\alpha}^T\right)^{-1} = (I - \beta\, P_\alpha)^{-1} = R(\sigma_\alpha).$$

In view of the complexity of the expression (3.15) it is, perhaps, surprising that the following relation can be demonstrated rather easily from (3.25):

$$c_r(1-\alpha,\, \beta) = c_{N-r}(\alpha,\, \beta), \quad r = 1, ..., N-1. \tag{3.26}$$

In particular, if we consider the H-curve corresponding to the standard Hamiltonian cycle, it was seen in (3.16) that the first row of $\bar{R}(\sigma_{1-\alpha})$ contains all the coefficients $\bar{c}_r(1-\alpha,\, \beta)$ in ascending order of r. Now, from (3.25) it follows that this row equals to the first column of $\bar{R}(\sigma_\alpha)$ containing the coefficients $\bar{c}_0(\alpha,\, \beta)$ and $\bar{c}_{N-r}(\alpha,\, \beta)$, in descending order. Hence, for $r = 1, \ldots, N-1$ we have that $\bar{c}_r(1-\alpha,\, \beta) = \bar{c}_{N-r}(\alpha,\, \beta)$. Now, (3.26) follows immediately from the uniqueness of the $c_r(\alpha,\, \beta)$ coefficients for all Hamiltonian cycles, see (3.22).

3.3.2 Properties of the angular function of a regular graph

An undirected graph G is called $d-$regular if the degree of each vertex equals d. From the point of view of determining a regular graph's Hamiltonicity it is clear that the case $d = 2$ is trivial but already for $d = 3$ it is known that HCP is NP-complete [24]. In this section it will be shown that H-curves of Hamiltonian regular graphs possess an interesting property that is related to the angles that the policy $u^\dagger$ makes with the policies σ_α in the space of discounted occupational measures $\mathcal{X}(\beta)$.

Towards this end we define an auxiliary *angular function* of one variable (where β is the fixed discount factor)

$$\theta(\alpha, \beta) := \sum_{k=1}^{N} < x^k(\sigma_\alpha, \beta)\ ,\ x^k(u^\dagger, \beta) >; \quad \alpha \in [0, 1]. \tag{3.27}$$

Theorem 3.3.1. *Let G be a $d-$regular Hamiltonian graph. Then on its H-curves the angular function satisfies:*

1. *For each $\alpha \in [0, 1]$, we have that $\theta(\alpha, \beta) = \theta(1-\alpha, \beta)$*
2. *The derivative of $\theta(\alpha, \beta)$ at the mid-point is zero, that is, $\dot{\theta}(\frac{1}{2}, \beta) = 0$.*

Proof. It is clear from (3.23) that for a $d-$regular graph,

$$x^k_{ia}(\dagger) = \begin{cases} \frac{1}{d}[R(u^\dagger)]_{ki} & if \,\mathrm{arc}\,(i, a) \in G \\ 0 & if \,\mathrm{arc}\,(i, a) \notin G. \end{cases}$$

We note also that $R(u^\dagger)$, the resolvent-like matrix of the center policy $u^\dagger$, is symmetric and hence $[R(u^\dagger)]_{ki} = [R(u^\dagger)]_{ik}$ for all k and i. Then, we define

$$\theta^k(\alpha, \beta) := x^k(\sigma_\alpha, \beta)\ x^k(u^\dagger, \beta) = \sum_{i=1}^{N} \sum_{a \in A(i)} x^k_{ia}(\dagger)\ x^k_{ia}(\alpha)$$

$$= \sum_{i=1}^{N} \sum_{a \in A(i)} \frac{1}{d}\ [R(u^\dagger)]_{ki}\ [R(\sigma_\alpha)]_{ki}\ \sigma_\alpha(i, a) = \frac{1}{d}\ \sum_{i=1}^{N} [R(\sigma_\alpha)]_{ki}\ [R(u^\dagger)]_{ik}$$

$$= \frac{1}{d} \sum_{i=1}^{N} [R(\sigma_\alpha)]_{ki}\ [R(u^\dagger)]_{ik} = \frac{1}{d}\ [R(\sigma_\alpha)\ R(u^\dagger)]_{kk}.$$

Hence it follows that

$$\theta(\alpha, \beta) = \sum_{k=1}^{N} \theta^k(\alpha, \beta) = \frac{1}{d} \sum_{k=1}^{N} [R(\sigma_\alpha) R(u^\dagger)]_{kk} = \frac{1}{d} \mathrm{Tr}\left(R(\sigma_\alpha) R(u^\dagger)\right), \tag{3.28}$$

where $\mathrm{Tr}(A)$ denotes the trace of matrix A.

Now using the well known property of the trace $\mathrm{Tr}(AB) = \mathrm{Tr}(A^T B^T)$, the symmetry of $R(u^\dagger)$ and the equation (3.25) we immediately obtain that for any $\alpha \in [0, 1]$

$$\begin{aligned}\theta(\alpha, \beta) &= \frac{1}{d} \mathrm{Tr}\left(R(\sigma_\alpha) R(u^\dagger)\right) = \frac{1}{d} \mathrm{Tr}\left(R^T(\sigma_\alpha) R^T(u^\dagger)\right) \\ &= \frac{1}{d} \mathrm{Tr}\left(R(\sigma_{1-\alpha}) R(u^\dagger)\right) = \theta(1 - \alpha, \beta).\end{aligned}$$

Thus the first part of the theorem is proved. The second part follows immediately from the first part since $\theta(\alpha, \beta)$ is differentiable with respect α on the interval $(0, 1)$. $\blacksquare$

Remark 3.3.2. *We note that for a fixed starting vertex k, in general, $\theta^k(\alpha, \beta) \neq \theta^k(1 - \alpha, \beta)$. This is why we need to consider all possible starting vertices in the definition of $\theta(\alpha, \beta)$.*

Of course, the angular function $\theta(\alpha, \beta)$ (for any fixed $\beta \in [0, 1)$) is well-defined for any undirected graph, not just for $d-$regular graphs. However, for general graphs there are examples for which Theorem 3.3.1 does not hold, so the regularity requirement is tight. It is also, perhaps, interesting that even in regular graphs $\theta(\alpha, \beta)$

depends on the Hamiltonian cycle inducing the H-curve. For instance, with the value of $\beta \in [0, 1)$ held fixed, the values of $\theta(\frac{1}{2}, \beta)$ can be different when they correspond to the H-curves induced by two distinct Hamiltonian cycles in the same regular graph. It seems that, these differences reflect the relative position of the occupational measure of the center policy $u^{\dagger}$ with respect to these distinct H-curves.

3.3.3 New equality constraints for Hamiltonian regular graphs

Much of the earlier work in this line of research (e.g., see [17] or [14]) relied on the representation of the space of discounted occupational measures $\mathcal{X}(\beta, k)$ in terms of linear constraints in variables $x_{ia}^k(u, \beta)$ defined by (3.1).

Next, we demonstrate that the statements of Theorem 3.3.1 can be also formulated in terms of linear constraints in the variables $x_{r,ia}^k$ introduced in Sections 2.3. This will be achieved with the help of the expansion (3.7) of the resolvent-like matrix $R(\sigma_\alpha)$ in terms of powers of the probability transition matrix $P(h)$ defined by the Hamiltonian cycle policy h. Interestingly, perhaps, the coefficients of these constraints will be functions of the parameters α and β, see (3.15), which are invariant on all H-curves of a Hamiltonian graph.

Now, from (3.4), the expansion (3.7) and the fact that for $P = P(h)$ for every $r = 0, \ldots, N-1$, $P^r = [P^{-1}]^{N-r}$ it follows that the variables $x_{ia}^k(\alpha)$ satisfy

$$x_{ia}^k(\alpha) = [R(\sigma_\alpha)]_{ki}\, \sigma_\alpha(i, a) =$$

$$\sum_{r=0}^{N-1} c_r(\alpha, \beta)[P^r]_{ki} \left(\alpha h(i, a) + (1-\alpha)h^{-1}(i, a)\right) =$$

$$\alpha \sum_{r=0}^{N-1} c_r(\alpha, \beta)[P^r]_{ki}\; h(i, a) + (1-\alpha) \sum_{r=0}^{N-1} c_r(\alpha, \beta) \left[P^{-1}\right]_{ki}^{N-r} h^{-1}(i, a). \qquad (3.29)$$

The above and (3.3) show that the variables in the first summation above can immediately be replaced by $x_{r,ia}^k$ variables. However, a little more analysis is needed to deal with the second summation in (3.29). Fortunately, the following relation connects a Hamiltonian cycle policy h to its reverse h^{-1} for each $r = 1, \dots, N-1$

$$\left[(P^{-1})^{N-r}\right]_{ki} h^{-1}(i,a) = \left[P^{r-1}\right]_{ka} h(a,i) = x^k_{r-1,ai}. \tag{3.30}$$

The above equation can be seen as a consequence of the fact that if the reverse Hamiltonian cycle h^{-1} takes $N-r$ steps to go from vertex k to vertex i and then chooses vertex a next, then the forward cycle h must go from k to a in exactly $r-1$ steps and then choose vertex i. Of course, if the arc (i,a) (respectively, (a,i)) does not lie on h^{-1} (respectively, h), then (3.30) reduces to $0 = 0$. The case of $r = 0$ is slightly different because step -1 is not defined. However, it is easy to check that

$$\left[P^{-1}\right]^N_{ki} h^{-1}(i,a) = \left[P^{N-1}\right]_{ka} h(a,i) = x^k_{N-1,ai}. \tag{3.31}$$

Indeed, for $k \neq i$ the above equation reduces to $0 = 0$ as $P^{-N} = I$ and $\left[P^{N-1}\right]_{ka} = 0$ unless a coincides with b_k, the last vertex on the Hamiltonian cycle h prior to return to vertex k. However, in the latter case $h(b_k, i) = 0$. If, on the other hand, $k = i$, then $h^{-1}(k,a) = 0$ except when $a = b_k$ when it equals to 1. Similarly, $h(a,k) = 0$ unless $a = b_k$ when it is equal 1 and simultaneously $\left[P^{N-1}\right]_{kb_k} = 1$.

Now substituting (3.30) and (3.31) into (3.29) we obtain

$$x^k_{ia}(\alpha) = \alpha \sum_{r=0}^{N-1} c_r(\alpha,\beta) x^k_{r,ia} + (1-\alpha) c_0(\alpha,\beta) x^k_{N-1,ai} + (1-\alpha) \sum_{r=1}^{N-1} c_r(\alpha,\beta) x^k_{r-1,ai}. \tag{3.32}$$

Now, let $\mathbf{x} = \{x^k_{r,ia} | r = 0, \dots, N-1; k, i \in S, a \in A(i)\}$ be a vector of variables induced by a Hamiltonian policy h via equation (3.3), with its entries ordered in any natural way. Then it is easy to see that (3.32) defines a linear function

$$x^k_{ia}(\alpha) = x^k_{ia}(\alpha,\beta) = \ell^k_{ia}(\mathbf{x}, \alpha, \beta), \tag{3.33}$$

in the variables of $\mathbf{x}(\alpha)$ where we are now emphasising the dependence on both parameters $\alpha \in [0,1)$ and $\beta \in (0,1)$.

The final result of this chapter shows that Theorem 3.3.1 supplies a parametrised of family of linear constraints that the vector $\mathbf{x}$ must satisfy.

Lemma 3.3.3. *Let h be any Hamiltonian cycle policy for a $d-$regular graph G. Let the entries of $\mathbf{x}$ be defined by $x^k_{r,ia} = [P^r(h)]_{ki}\, h(i,a)$ for $r = 0, \ldots, N-1; k, i \in S$ and $a \in A(i)$. Then for every $\alpha \in [0,1]$, and $\beta \in [0,1)$, the vector $\mathbf{x}$ satisfies the linear constraints*

$$\sum_{k=1}^{N}\sum_{i=1}^{N}\sum_{a\in A(i)} [\ell^k_{ia}(\mathbf{x},\alpha,\beta) - \ell^k_{ia}(\mathbf{x}, 1-\alpha,\beta)]x^k_{ia}(u^\dagger,\beta) = 0, \tag{3.34}$$

where $u^\dagger$ is the center policy.

Proof. This result follows immediately from the definition (3.27) of the angular function $\theta(\alpha,\beta)$, part one of Theorem 3.3.1 and (3.33). ■

Remark 3.3.4. (i) *Note that for each $\beta \in [0,1)$ numerical values of coefficients $x^k_{ia}(u^\dagger,\beta)$ are known, in the sense that they can be easily calculated from (3.23).*

(ii) *Similarly, the coefficients $c_r(\alpha,\beta)$ embedded in (3.34) via (3.32)-(3.33) are known in the sense that for every $\alpha \in [0,1]$ and $\beta \in [0,1]$ they can be recursively calculated with the help of Theorem 3.2.4.*

(iii) *Thus (3.34) represents a parametrized family of linear constraints that a vector* $\mathbf{x}$ *(induced by a Hamiltonian cycle h) must satisfy. The question of whether these constraints can help to identify non-Hamiltonian regular graphs is the subject of continuing investigations.*

Chapter 4

Discovery of Unidentified Equality Constraints for Integer Programming Problems

Characterising the smallest dimension polytope containing all integer solution of an integer programming problem can be a very challenging task. Frequently, this task is facilitated by identifying linear equality constraints that all integer solutions must satisfy. Typically, some of these constraints are readily available but others need to be discovered by more technical means. This chapter[1] describe a method to assist modelers to obtain such equality constraints. Note that the set of new equality constraints is not unique, and the proposed method generates a set of these new equality constraints for a sufficiently large dimension of the underlying problem. These generated constraints may be of a form that is easily extended for general case of the underlying problem, or they may be in a more complicated form where a generalisable pattern is difficult to identify. For the latter case, a a new mixed-integer program is developed to detect a pattern-recognisable constraints. Furthermore, this mixed-integer program allows modelers to check if there is a new constraint satisfying specific criteria, such as only permitting coefficients to be 1, 0, and -1, or placing a limit on the

[1]The main results of this chapter is under review in a journal for publication [30].

number of non-zero coefficients. In order to illustrate the proposed method, a set of new equality constraints to supplement the parameter-free model defined in Section 2.3, are derived. Subsequently, exploiting these results, some techniques are proposed to tighten integer programming problems. Finally, relaxations of widely used TSP formulations are compared against one another and strengthened with help of the newly discovered equality constraints.

4.1 Introduction

Many problems in Operations Research or Industrial Engineering can be formulated as *Integer Programming* (IP) problems of the form

$$\begin{array}{lll} \text{Minimise} & c^T x & \\ \text{subject to} & A_0 x = b_0 & \\ & B_0 x \geq d_0 & \text{(IP)} \\ & x \in \mathbb{Z}^n & \end{array}$$

where x is a vector representing the decision variables, $c \in \mathbb{R}^n$ is a vector representing objective function coefficients, $A \in \mathbb{R}^{m \times n}$ is matrix representing the technological coefficients, and $b \in \mathbb{R}^m$ is a vector representing the right hand side values.

Since IP problems are known to be NP-hard, it is common to begin their analysis by first considering their LP-relaxation, namely the problems

$$\begin{array}{lll} \text{Minimise} & c^T x & \\ \text{subject to} & A_0 x = b_0 & \text{(LIP)} \\ & B_0 x \geq d_0. & \end{array}$$

A natural question that arises is whether the feasible region FLIP of the above LP-relaxation could be further reduced, without eliminating any integer valued points lying in FIP, the feasible region of (IP). In order to undertake such a reduction task we must utilise the properties of polytopes associated with these problems.

It is well known (*e.g.* see [39]) that there are two main representations for a polytope. In the V-representation, a polytope is identified by the set of its extreme points and in the H-representation, a polytope is specified as set of solutions to linear constraints (equalities and/or inequalities). An interesting question in IP context is how one can obtain an H-representation from its V-representation. More precisely, if FIP is a finite set, then the convex hull of FIP, **conv**(FIP), is a well-defined polytope and thus by Weyl's theorem [37], there exists a finite set of linear constraints that completely describes **conv**(FIP). Such a linear description is called the H-representation of the set FIP (see also [38]). Finding the H-representation for a given set FIP is called the convex hull problem and has been studied by many researches, for example [5], [4], and [23]. There also exist some software packages such as Polymake [25] and Porta [10], for polytope specification purposes. It should be noted that in this chapter we use the term "H-representation" for polyhedral representation of instances of an IP problem, and we use the term "convex hull formulation" for the polyhedral representation in the general case of the underlying problem.

Discovering convex hull formulations is, in general,a difficult task. To the author's knowledge there is no systematic method of doing so, and it is usually achieved by a trial and error process. Generally this means trying to guess a valid equality (or inequality) constraint and then proving that it works for any instance of the underlying problem. One common method for guessing valid constraints is to extract the H-representation of very small instances, and then attempt to find generalisable patterns.

The above trial and error process to detect convex hull formulation usually leads to some difficulties. The main issue with approaches that include first finding the H-representation is that they are very limited in terms of size [3], with even small instances often containing a huge number of inequality constraints. For instance, 10-vertex TSP polytope has over 50 billion facets [28]. This limitation sometimes prevents the identified constraint from being generalisable. The reason behind it is that those small instances are insufficient to develop an intuition, because the behaviors of small instances often are not extendable for the general case. These issues reveal that in order to analyse the polytope of a problem, one needs to study the H-representation of sufficiently large instances to avoid the results being influenced by exceptional behaviors that are peculiar to the H-representations for small instances of underlying problem. Another aspect of the trial and error process is that the H-representation of a polytope is not unique. For example, any linear combination of two equality constraints can replace one of them to obtain a new H-representation. This sometimes means that H-representation approaches yield a set of constraints in which the underlying patterns are disguised beyond easy recognition.

On the other hand, one may only be interested in equality constraints. In particular, this can be the case when a model includes many extended variables. In this case the modeller is able to express many inequality constraints of lower dimensional models as new equality constraints by using these extended variables. Roughly speaking, extended models contain fewer inequalities and more equalities in comparison to original models. Thus, the difficulty of finding good inequality constraints in low dimensional models can, perhaps, be exchanged with the difficulty of finding good equality constraints in high dimensional models. Therefore, extracting all equality constraints for an extended model is not as simple as for a model in lower dimensional space. Motivated by this, the present chapter includes a proposed method to generate a set of all the new equality constraints for a given IP model. More precisely, the proposed method assists modelers in finding as many independent equality constraints as possible to characterise a given set of points (integer-valued solutions).

Apart from its direct advantage, the proposed method can be applied to strengthen an IP model in several ways. For example, by adding new variables and creating new constraints based on the relationships between existing and new variables. This is the main idea of the extended formulation's context (for example see [11]). In Section 4.5, an application of this approach is exhibited to strengthen Desrochers and Laporte's model for TSP [12]. An alternative approach to strengthen an IP model is to create an extended model for the IP problem, and then project the space of this extended model to the original variable space to find a number of good valid inequalities. For example, Gouveia et al. projected an extended model and found some complex facets of the original model [26]. See [7] for an overview of projections of polytopes.

The remainder of the chapter is arranged as follows. Notation and definitions are introduced in Section 4.2. The proposed method is presented in Section 4.3. Details of the proposed method are illustrated through finding new equality constraints for the parameter-free polytope in Section 4.4. In Section 4.5 some applications are given through a numerical comparisons between two vertex-oriented TSP models.

4.2 Preliminaries and notations

This section sets up the notation and offers a quick review of affine sets and their relation to subspaces and nullspaces.

4.2.1 Affine sets

A set $C \subseteq R^n$ is *affine* if the line through any two distinct points in C lies in C, that is, if for any $x_1, x_2 \in C$ and $\theta \in R$, we have $\theta\, x_1 + (1-\theta)\, x_2 \in C$. This idea can be extended to more than two points. We refer to a point of the form $\theta_1\, x_1 + \ldots + \theta_k\, x_k$, where $\theta_1 + \ldots + \theta_k = 1$, as an affine combination of the points $x_1, \ldots, x_k$. It can be shown that an affine set contains every affine combination of its points. The set of all affine combinations of points in some set $S \in R^n$ is called the *affine hull of* S, and denoted $\mathbf{aff}\,(S)$:

$$\mathbf{aff}(S) = \{\theta_1 x1 + \ldots + \theta_k x_k \,|\, x_1, \ldots, x_k \in S, \quad \theta_1, \ldots, \theta_k \in R, \quad \theta_1 + \ldots + \theta_k = 1\}.$$

The points $x_1, ..., x_k \in R^n$ are affinely independent if and only if no point from $x_1, ..., x_k$ can be written as an affine combination of the others. The dimension of an affine set C, denoted by $\mathbf{dim}(C)$, is the maximum number of affinely independent points in C minus one. Similarly, the dimension of the polytope constructed as the convex hull of points $x_1, ..., x_k$ is equal to the dimension of the affine hull of these points.

If C is an affine set and $x_0 \in C$, then the set $V = C - x_0 = \{x - x_0 \,|\, x \in C\}$ is a subspace. Thus, the affine set C can be expressed as a subspace plus an offset, that is, $C = V + x_0 = \{v + x_0 \,|\, v \in V\}$. The subspace V associated with the affine set C does not depend on the choice of x_0, so x_0 can be chosen as any point in C. Furthermore, every affine set can be expressed as the solution set of a system of linear equations, that is $C = \{x \in R^n \,|\, Ax = b\}$, where $A \in R^{m\times n}$ and $b \in R^m$. The subspace associated with the affine set C is the nullspace of A, that is, $C = V + x_0 = \{v + x_0 \,|\, Av = 0\}$, where x_0 is any point in C. If C is nonempty, then $\mathbf{dim}(C) = n - \mathbf{rank}(A)$.

To explain the proposed method, we need the following preliminaries and definitions. Let FIP$= \{s_i \in R^n, i = 0, \ldots, m\}$ denote the set of solutions of (IP) and define: $v_i := s_i - s_0,\ i = 1, \ldots, m,\ V := \mathbf{span}(v_1, \ldots, v_m),\ D := [v_1|v_2|\ldots|v_m]$ the $n \times m$ matrix whose columns are v_i's. Also, let $\mathbf{C}(D)$ denote the column space of the matrix D. Obviously

$$\mathbf{C}(D) = \mathbf{span}(v_1, \ldots, v_m) = V. \tag{4.1}$$

Let $\mathbf{C}(D^T)$ denote the row space of the matrix D, and $A_0 x = b_0$ as in the formulation of (IP) and (LIP). The latter constitute the available equality constraints of the IP model. It follows that $\mathbf{dim}(\text{FLIP}) = n - \mathbf{rank}(A_0)$ if the inequalities $B_0 x \geq d_0$ do not imply additional equality constraints and also, $\mathbf{dim}(\mathbf{conv}(\text{FIP})) = \mathbf{rank}(D)$. We let $\mathbf{null}(G)$ be a matrix whose columns are a basis of the nullspace of the matrix G obtained by the reduced row echelon method and consider $\mathbf{aff}$(FIP) $= \{\theta_0\, s_0 + \ldots + \theta_m\, s_m \,|\, x_1, \ldots, x_k \in S,\ \ \theta_1, \ldots, \theta_k \in R,\ \ \theta_1 + \ldots + \theta_k = 1\}$. We recall that for an arbitrary matrix A, we have

$$(\mathbf{C}(A))^\perp = \text{nullspace}\,(A^T). \tag{4.2}$$

4.3 Equality constraint augmenting method

In this section, we propose a method to append a set of new equality constraints for IP problems already endowed with a partial set of equality constraints. We call this method *Equality Constraint Augmenting* method, or ECA-method, for short. Subsequently, a mixed-integer programming model is formulated to detect patterns in the generated constraints.

First of all, one should check if it is even possible to add any new equality constraints before making any attempts to find them. This can be checked by comparing the dimension of the polytope FLIP constructed by the relaxation of the IP model and the dimension of the polytope **conv**(FIP). If these dimensions coincide, it is not possible to find any new (non-redundant) equality constraints. If the dimension of **conv** (FIP) is d units smaller than the dimension of FLIP, this implies that new equality constraints exist, and it should be possible to identify and add exactly d of them.

4.3.1 ECA-method

Ignoring the available equality constraints $A_0x = b_0$, the ECA-method finds a set of all equality constraints and then removes the ones captured by the equality constraints which are already known. Thus, only the constraints that contribute new information are retained.

To extract all equality constraints, one needs to find a linear equation system, $Ax = b$, which represents the affine hull of FLIP (**aff** (FLIP)), that is, $\{Aw = b \,|\, w \in \mathbf{aff}\,(\text{FLIP})\}$. Since **aff** (FLIP) is an affine set, then as described in Section 4.2, it can be expressed as a subspace and one of its points s_0 as follows:

$$\mathbf{aff}\,(\text{FLIP}) = V + s_0 = \{v + s_0 \,|\, v \in V = \mathbf{span}\,(v_1, \ldots, v_m)\} = \{v + s_0 \,|\, Av = 0\}, \tag{4.3}$$

where s_0 is one of the integer-valued solutions of (IP). In (4.3) we have $Av = 0$ for $v \in V$, which shows that the row space of the matrix A is the orthogonal complement of subspace V, that is, $\mathbf{C}(A^T) = V^{\perp}$. By using (4.1) and (4.2), one can write $\mathbf{C}(A^T) = (\mathbf{C}(D))^{\perp} = \text{nullspace}\,(D^T)$. This means that the row space of matrix A is exactly the nullspace of the matrix D^T, and therefore, the smallest representation of matrix A occurs when rows of matrix A are a basis of the nullspace (D^T). This is the smallest representation, since matrix A does not contain any redundant rows. Suppose $\mathbf{null}(G)$ is a matrix whose columns are a basis of the nullspace of matrix G, then the matrix A can be derived as $A = \left(\mathbf{null}\left(D^T\right)\right)^T$ and the vector b is readily available via $b := As_0$.

Remark 4.3.1. *Of course, the problem of finding a basis of the nullspace(D^T) is practical only in the case of small examples. However, new constraints so discovered may be generalisable for all dimensions.*

All the equality constraints will be available once the above process is completed. The generated constraints must be consecutively checked to see whether they contribute any new information. That is, the first generated constraint is added to the available equality constraints to see if it increases the rank of the coefficient matrix by one. If so, the constraint is non-redundant and the available constraints should be updated by including this new constraint. Repeating this process ensures all new added constraints are non-redundant. Of course, the order in which the new constraints are added influences which ones are retained. Hence, we sort the generated constraints based on the number of their terms, to make it more likely that the constraints to refund are simpler to interpret.

Remark 4.3.2. *One may be interested in directly extracting a set of non-redundant constraints, since in most (IP) models, some equality constraints are available. To consider this problem, let $A_0x = b_0$ be the set of equality constraint that are available, and suppose there exists a set of new equality constraints, namely $A_1x = b_1$. Therefore, the problem is to find A_1 and b_1.*

To create the latter set of independent constraints from the available constraints $A_0x = b_0$, each row of matrix A_1 must be independent from every row of matrix A_0. One way to achieve this independence is to create A_1 so that its row space belongs to the orthogonal complement of the row space of matrix A_0, that is:

$$\boldsymbol{C}(A_0^T) \in \left(\boldsymbol{C}(A_1^T)\right)^{\perp} = \mathit{nullspace}\,(A_1), \tag{4.4}$$

and also based on the result in Section 4.3.1, one can write:

$$\boldsymbol{C}(A_0^T) \in (\boldsymbol{C}(D))^{\perp} = nullspace\,(D^T). \tag{4.5}$$

Exploiting (4.4) and (4.5), one can write $\boldsymbol{C}(A_0^T) \in (\boldsymbol{C}(D))^{\perp} = nullspace\left(\begin{bmatrix} D^T \\ A_0 \end{bmatrix}\right)$. *Therefore, the matrix* A_1 *can be expressed as* $A_1 = \left(\boldsymbol{null}\begin{bmatrix} D^T \\ A_0 \end{bmatrix}\right)^T$. *Finally vector* b_1 *is derived by* $b_1 = A_1 s_0$. *Note that this is a one way condition, namely, if this condition is satisfied, rows of the matrix* A_0 *and* A_1 *are orthogonal, and consequently they will be independent. However, it is also possible to find an* A_1 *which violates this condition even though its rows are also independent from the rows of* A_0.

Although this direct method can easily generate the set of new (non-redundant) equality constraints, in practice, it increases the likelihood of generating complicated constraints, and hence is likely to result in difficulties in revealing the pattern of generated constraints. Therefore, the ECA-method is preferable. We are now in a position to illustrate the details of these two methods via a simple example.

Example 4.3.3. *Suppose* $s_0 = [0\ 1\ 0\ 1\ 1\ 0]^T$, $s_1 = [1\ 0\ 1\ 0\ 1\ 0]^T$, $s_2 = [0\ 0\ 0\ 1\ 0\ 1]^T$ *in* R^6 *are the only feasible points for an IP model, and there are two known equality constraints,* $(-x_2 + x_4 + x_5 = 1,\ 2x_1 + x_2 + x_4 - x_5 = 1)$, *which are satisfied by those three points. We need to find the new equality constraints (if any). Both ECA-method and direct method are implemented to find the new equality constraints below. The known constraints can be written as:*

$$\begin{bmatrix} 0 & -1 & 0 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Therefore, $A_0 = \begin{bmatrix} 0 & -1 & 0 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 & -1 & 0 \end{bmatrix}$ *and* $b_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

ECA-method: *We define* v_i *vectors and matrix* D *as follows:* $v_i = s_i - s_0$, $i = 1, 2$.

$$v_1^T = \begin{bmatrix} 1 & -1 & 1 & -1 & 0 & 0 \end{bmatrix}, v_2^T = \begin{bmatrix} 0 & -1 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

Then

$$D^T = \begin{bmatrix} 1 & -1 & 1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

Note that ***dim****(FLIP)* $= 6 - $ ***rank***$(A_0) = 4$, *and* ***dim****(****conv****(FIP))* $=$ ***rank*** $(D) = 2$. *These computations shows that it is possible to find 2 new equality constraints* $(\boldsymbol{dim}(FLIP) - \boldsymbol{dim}(\boldsymbol{conv}(FIP)))$. *Applying the ECA-method, we find all equality constraints as follows:*

$$A = \left(\boldsymbol{null}\left(D^T\right)\right)^T = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \ b = As_0 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix},$$

therefore the system of equations is:

$$Ax = b; \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

Thus we have found a set of all equality constraints, and now we are required to remove the redundant ones. Considering

$$\begin{bmatrix} A_0 | b_0 \end{bmatrix} = \left[\begin{array}{cccccc|c} 0 & -1 & 0 & 1 & 1 & 0 & 1 \\ 2 & 1 & 0 & 1 & -1 & 0 & 1 \end{array}\right], \ \boldsymbol{rank}\left(\begin{bmatrix} A_0 | b_0 \end{bmatrix}\right) = 2,$$

we add the first generated constraint to $\left[A_0|b_0\right]$ *and check the rank of the so constructed matrix*

$$\boldsymbol{rank}\left(\left[\begin{array}{cccccc|c} 0 & -1 & 0 & 1 & 1 & 0 & 1 \\ 2 & 1 & 0 & 1 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]\right) = 3,$$

it increases the rank by one and this reveals that it is a new non-redundant constraint. Thus, this constraint should be appended to the existing constraints. Next, we add the second generated constraint to this matrix and check its rank as follows:

$$\boldsymbol{rank}\left(\left[\begin{array}{cccccc|c} 0 & -1 & 0 & 1 & 1 & 0 & 1 \\ 2 & 1 & 0 & 1 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right]\right) = 3.$$

The above rank of matrix has not changed, therefore the second generated constraint is redundant and should be removed. Then, we try the third generated constraint as follows:

$$\boldsymbol{rank}\left(\left[\begin{array}{cccccc|c} 0 & -1 & 0 & 1 & 1 & 0 & 1 \\ 2 & 1 & 0 & 1 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & 0 & 0 \end{array}\right]\right) = 3.$$

Similarly, the third constraint also does not change the rank and is redundant. Finally we check the forth generated constraint as follows:

$$\boldsymbol{rank}\left(\left[\begin{array}{cccccc|c} 0 & -1 & 0 & 1 & 1 & 0 & 1 \\ 2 & 1 & 0 & 1 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{array}\right]\right) = 4.$$

The latter is a new non-redundant constraint since it increases the rank. As is shown with the above process, the first and fourth generated constraints ($x_1 = x_3$, $x_1 + x_2 + x_6 = 1$) *are the full set of new equality constraints.*

Direct method: *As explained before, constraints in this method can be obtained by setting:*

$$A_1 = \left[\boldsymbol{null}\left(\begin{bmatrix} D^T \\ A_0 \end{bmatrix}\right)\right]^T = \begin{bmatrix} 2 & -1 & -5 & -2 & 1 & 0 \\ -1 & 1 & 3 & 1 & 0 & 1 \end{bmatrix},\ b_1 = A_1 s_0 = \begin{bmatrix} -2 \\ 2 \end{bmatrix},$$

In other words, the new found constraints are $(2x_1 - x_2 - 5x_3 - 2x_4 + x_5 = -2,\ -x_1 + x_2 + 3x_3 + x_4 + x_6 = 2)$.

As seen in Example 4.3.3 the new non-redundant constraints generated by the ECA-method seem simpler and more convenient to explore with the view of finding generalizable patterns. Obviously, the constraints found by either method imply those found by the other.

4.3.2 A mixed-integer model to assist in pattern recognition

Two cases can arise after generating a set of new equality constraints in a problem of small dimension. In the first case, one is able to recognise a pattern in the newly generated constraints. Those constraints are then generalised, shown to be valid in all instances, and appended to the available set of constraints. In the second case, we cannot find any pattern, but as the set of equality constraints for an IP is not unique, we may be interested in taking clever linear combination of the constraints in the hope of discovering an alternative set of generalisable constraints. To achieve this, a mixed-integer model is designed to identify the constraints that are more likely to be generalisable.

The mixed-integer model enables the modeler to check if there is any new constraint that satisfies specific criteria. These criteria may restrict the coefficients of the new constraints to be 1, 0, and -1, or demand a constraint with a minimal number of non-zero coefficients. Another interesting case is to request zero coefficients for some of the variables or, perhaps, one might be interested in having some symmetries in the new constraint. For example, we may consider a case where the number of positive coefficients is equal to the number of negative coefficients. All of these patterns are verifiable via the proposed mixed-integer model. We explain the main framework of this model through an example, and one can adapt it based on the desired criteria.

Suppose we are interested in a new constraint whose coefficients take values of -1 or 0 or 1, and which also includes the minimum number of non-zero coefficients subject to this requirement. Furthermore, we are interested in having an equal number of positive and negative coefficients. Also, suppose that a representation for A_1, b_1 is known as they were generated by ECA-method. Clearly a new generalisable constraint must be constructed from a linear combination of known constraints ($[A_0|b_0]$) and generated constraints ($[A_1|b_1]$). The following mixed-integer model is designed for the mentioned criteria.

Proposed mixed-integer model: Let Y and Z be two $(0,1)$-vectors in R^n (where n is the number of decision variables in the IP model) and w_0 and w_1 be two continuous vectors. Then the proposed mixed-integer model for the given preferences is:

$$\text{Minimise} \qquad \mathbf{Y}^T\mathbf{e} + \mathbf{Z}^T\mathbf{e} \tag{4.6}$$

$$\text{subject to} \qquad \mathbf{w}_0^T[A_0|b_0] + \mathbf{w}_1^T[A_1|b_1] = \mathbf{Y}^T - \mathbf{Z}^T \tag{4.7}$$

$$\mathbf{w}_1^T\mathbf{e} \geq \epsilon \tag{4.8}$$

$$\mathbf{Y}^T\mathbf{e} = \mathbf{Z}^T\mathbf{e} \tag{4.9}$$

$$\mathbf{Y}, \mathbf{Z} \in \{0,1\} \tag{4.10}$$

where ϵ is a small positive parameter and $\mathbf{e}$ is a vector with all unit entries. Let $\mathcal{L}_i = \mathbf{Y}_i - \mathbf{Z}_i, i = 1, \ldots, n$. Note that each $\mathcal{L}_i \in \{-1, 0, 1\}$ since it is obtained by subtracting of two binary variables. Note also that if $\mathcal{L}_i = 1$, then $\mathbf{Y}_i = 1$, $\mathbf{Z}_i = 0$ and when $\mathcal{L}_i = -1$, then $\mathbf{Y}_i = 0, \mathbf{Z}_i = 1$, and finally when $\mathcal{L}_i = 0$, then $\mathbf{Y}_i = 0, \mathbf{Z}_i = 0$. The case of $\mathcal{L}_i = 0$ derived by $\mathbf{Y}_i = 1, \mathbf{Z}_i = 1$, never occurs, because the objective function prevents it. In constraint (4.7), a linear combination of the known constraints (rows of $[A_0|b_0]$), and generated constraints (rows of $[A_1|b_1]$) are used to construct the vector $\mathcal{L}$. In this linear combination $\mathbf{w}_0$ is a vector that contains multipliers of the known constraints, and $\mathbf{w}_1$ has the same role for the generated constraints. Constraint

(4.8) forces at least one multiplier of the generated constraints to be non-zero, and this guarantees that the model finds a non-redundant constraint. Constraint (4.9) ensures that the number of positive $\mathcal{L}_i$'s equals the number of negative $\mathcal{L}_i$'s. If the model is infeasible, it shows there are no constraints that satisfy these criteria. In such a case, modeler should relax these criteria. For example, Constraint (4.9) could be removed to see if there exist a new constraint with coefficients in -1, 0, or 1.

4.4 New equality constraints for parameter-free model

To illustrate ECA-method explained in Section 4.3, it was implemented to discover new equality constraints to refine the parameter-free polytope introduced in [19], and [6] for the Hamiltonian Cycle Problem. Moreover, this polytope can be considered as an extended model for TSP and *Time Dependent Traveling Salesman Problem* (TD-TSP). In this model variable $x^k_{r,ia}$ is equal to 1 whenever (i,a) is the r-th arc on the tour, assuming vertex k to be the starting vertex. Recall from Section 2.3 that the constraints of this model can be written as

$$\sum_{a\in\mathcal{A}(i)} x^k_{r,ia} - \sum_{a\in\mathcal{A}(i)} x^k_{r-1,ai} = 0, \qquad k,i=1,\dots,N;\ r=1,\dots,N-1, \tag{4.11}$$

$$\sum_{a\in\mathcal{A}(i)} x^k_{r,ia} - \sum_{a\in\mathcal{A}(k)} x^i_{N-r,ka} = 0, \qquad k,i=1,\dots,N;\ r=1,\dots,N-1, \tag{4.12}$$

$$\sum_{r=0}^{N-1} x^k_{r,ia} - \sum_{r=0}^{N-1} x^j_{r,ia} = 0, \qquad k,j=1,\dots,N, k\neq j;\ (i,a)\in\Gamma, \tag{4.13}$$

$$\sum_{k=1}^{N} x^k_{r,ia} - \sum_{k=1}^{N} x^k_{s,ia} = 0, \qquad r,s=0,\dots,N-1, r\neq s;\ (i,a)\in\Gamma, \tag{4.14}$$

$$\sum_{r=0}^{N-1}\sum_{a\in\mathcal{A}(i)} x^k_{r,ia} = 1, \qquad k,i=1,\dots,N, \tag{4.15}$$

$$\sum_{k=1}^{N}\sum_{a\in\mathcal{A}(i)} x^k_{r,ia} = 1, \qquad r=0,\dots,N-1;\ i=1,\dots,N, \tag{4.16}$$

$$x^k_{0,ia} = 0, \qquad k,i=1,\dots,N, i\neq k; (i,a)\in\Gamma, \tag{4.17}$$

$$x^k_{r,ia} \geq 0, \quad k=1,\dots,N;\ r=0,\dots N-1;\ (i,a)\in\Gamma. \tag{4.18}$$

In the above, $\mathcal{A}(i)$ denotes the set of vertices that can be reached from a given vertex i in a single step. Note that Theorem 2.5.2 implies if, in addition to (4.11)–(4.18), we demand that $x^k_{r,ia} \in \{0, 1\}$, then every feasible point defines a tour. Thus (4.11)–(4.18) essentially constitutes an LP-relaxation.

To start the analysis, we first find the dimension of **conv**(FIP) and FLIP for the complete graphs of various size. These data are provided in the Table 1. Note that in this example, FIP is set of all tours and FLIP is the polytope constructed by constraints (4.11)–(4.18).

Table 4.1: First dimension study

N	#Decision Variables	**rank**(A_0)	**dim**(**conv**(FIP))	**dim**(FLIP)	#Unidentified Equality Constraints
2	4	4	0	0	0
3	18	17	1	1	0
4	72	67	5	6	1
5	220	175	22	45	23
6	540	361	95	179	84
7	1134	645	316	489	173
8	2128	1050	859	1078	219
9	3672	1595	1763	2077	314
10	5940	2303	3158	3637	479
11	9130	3193	5245	5937	692
12	13464	4288	8217	9176	959
13	19188	5607	12295	13581	1286
14	26572	7173	16106	19399	3293

As shown in the above table, the dimension of **conv**(FIP) is always smaller than (or equal to) the dimension of FLIP. Interestingly, Table 4.1 reveals that there exist some new equality constraints. Implementing the ECA-method of Section 4.3 for $N = 5$, the following 23 non-redundant new equality constraints were generated. We call this

Set 1 of new constraints and list them below

$x^2_{1,31} - x^1_{3,23} = 0,$

$x^2_{1,41} - x^1_{3,24} = 0,$

$x^3_{1,41} - x^1_{3,34} = 0,$

$x^1_{2,23} + x^2_{1,53} - x^1_{1,42} - x^1_{3,34} = 0,$

$x^1_{3,23} + x^2_{2,13} - x^1_{2,42} - x^1_{2,52} = 0,$

$x^1_{3,24} + x^2_{2,14} - x^1_{2,32} - x^1_{2,52} = 0,$

$x^1_{2,23} + x^2_{2,31} - x^1_{1,42} - x^1_{1,52} = 0,$

$x^1_{2,24} + x^2_{2,41} - x^1_{1,32} - x^1_{1,52} = 0,$

$x^1_{3,32} + x^3_{2,12} - x^1_{2,43} - x^1_{2,53} = 0,$

$x^1_{3,34} + x^3_{2,14} - x^1_{2,23} - x^1_{2,53} = 0,$

$x^1_{2,34} + x^3_{2,41} - x^1_{1,23} - x^1_{1,53} = 0,$

$x^1_{1,32} + x^1_{1,34} + x^1_{3,24} + x^1_{3,42} - x^1_{0,13} - x^1_{1,53} = 0,$

$x^1_{1,32} + x^1_{2,42} + x^1_{2,52} + x^2_{1,13} - x^1_{0,13} - x^1_{3,23} = 0,$

$x^1_{1,42} + x^1_{2,32} + x^1_{2,52} + x^2_{1,14} - x^1_{0,14} - x^1_{3,24} = 0,$

$x^1_{3,32} + x^1_{3,34} + x^2_{1,43} - x^1_{1,52} - x^1_{2,43} - x^1_{2,53} = 0,$

$x^1_{1,23} + x^1_{2,43} + x^1_{2,53} + x^3_{1,12} - x^1_{0,12} - x^1_{3,32} = 0,$

$x^1_{1,43} + x^1_{2,23} + x^1_{2,53} + x^3_{1,14} - x^1_{0,14} - x^1_{3,34} = 0,$

$x^1_{0,14} + x^1_{2,42} + x^2_{2,34} - x^1_{1,34} - x^1_{1,42} - x^1_{1,43} - x^1_{2,53} - x^1_{3,23} = 0,$

$x^1_{0,12} + x^1_{0,13} + x^1_{1,42} + x^1_{1,43} + x^1_{1,52} + x^1_{1,53} + x^1_{3,23} + x^1_{3,32} = 1,$

$x^1_{1,42} + x^1_{1,43} + x^1_{1,53} + x^1_{2,52} + x^1_{2,53} + x^2_{1,34} - x^1_{0,14} - x^1_{2,34} - x^1_{3,24} - x^1_{3,34} = 0,$

$x^1_{1,42} + x^1_{1,43} + x^1_{1,52} + x^1_{2,52} + x^1_{2,53} + x^3_{1,24} - x^1_{0,14} - x^1_{2,24} - x^1_{3,24} - x^1_{3,34} = 0,$

$x^1_{0,14} + x^1_{2,34} + x^1_{3,24} + x^4_{1,12} - x^1_{1,23} - x^1_{1,42} - x^1_{1,43} - x^1_{1,53} - x^1_{2,52} - x^1_{2,53} = 0,$

$x^1_{1,23} + x^1_{1,24} + x^1_{1,42} + x^1_{1,43} + x^1_{1,53} + x^1_{2,52} + x^1_{2,53} + x^4_{2,12} - x^1_{0,12} - x^1_{0,14} - x^1_{2,34} - x^1_{3,24} = 0.$

We try to identify a generalisable pattern in the simplest group of the generated constraints, namely, the first three constraints. Seemingly, the pattern is $x^k_{1,ij} =$

$x^j_{N-2,ki}$, for all distinct $k, i, j = 1, \ldots, N$. These constraints have a clear interpretation illustrated by Figure 4.1. They show that if a tour starts at vertex k and goes to i in the first step, followed by transition from i to j, then this ensures that the tour starting at j will pass through arc (k, i) after $N - 2$ steps. We shall call these the 2-step constraints.

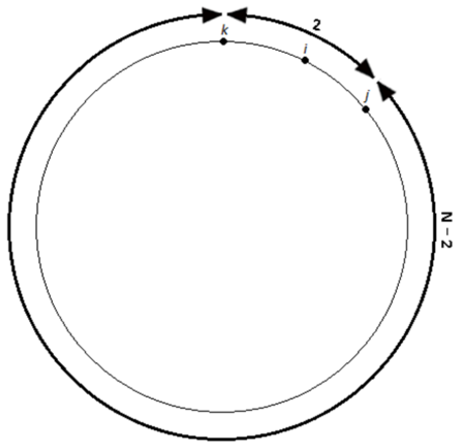

Figure 4.1: Demonstration of a 2-step constraint

There are 60 possible 2-step constraints for $N = 5$ and only three of them appeared in Set 1. One may think that all remaining constraints of this kind (which are not present in the generated constraints) are redundant, but it is not the case as they only would be redundant if all other generated constraints (constraints 4 to 23) were added. Note that, at this stage of the example, the generalisable patterns (if any) of constraints 4 to 23 are still unknown. Therefore we must add all the 60 constraints of this simple kind and then begin the process again. Analysing the dimension of FLIP after adding the new constraints reveals that those 60 constraints provide 14 new non-redundant equality constraints, which verifies that the 2-step constraints were more valuable than just the three new constraints in Set 1. After appending the 2-step constraints to (4.11)–(4.18) constraints and repeating the ECA-method, the following nine unidentified generated constraints remain. We call this Set 2 of unidentified constraints:

$$x^1_{2,23} + x^2_{1,53} - x^1_{142} - x^1_{3,34} = 0,$$

$$x^1_{1,32} + x^1_{1,34} + x^1_{3,24} + x^1_{3,42} - x^1_{0,13} - x^1_{1,53} = 0,$$

$$x^1_{1,32} + x^1_{2,42} + x^1_{2,52} + x^2_{1,13} - x^1_{0,13} - x^1_{3,23} = 0,$$

$$x^1_{1,42} + x^1_{2,32} + x^1_{2,52} + x^2_{1,14} - x^1_{0,14} - x^1_{3,24} = 0,$$

$$x^1_{3,32} + x^1_{3,34} + x^2_{1,43} - x^1_{1,52} - x^1_{2,43} - x^1_{2,53} = 0,$$

$$x^1_{1,43} + x^1_{2,23} + x^1_{2,53} + x^3_{1,14} - x^1_{0,14} - x^1_{3,34} = 0,$$

$$x^1_{0,12} + x^1_{0,13} + x^1_{1,42} + x^1_{1,43} + x^1_{1,52} + x^1_{1,53} + x^1_{3,23} + x^1_{3,32} = 1,$$

$$x^1_{1,42} + x^1_{1,43} + x^1_{1,53} + x^1_{2,52} + x^1_{2,53} + x^2_{1,34} - x^1_{0,14} - x^1_{2,34} - x^1_{3,24} - x^1_{3,34} = 0,$$

$$x^1_{1,42} + x^1_{1,43} + x^1_{1,52} + x^1_{2,52} + x^1_{2,53} + x^3_{1,24} - x^1_{0,14} - x^1_{2,24} - x^1_{3,24} - x^1_{3,34} = 0.$$

We now define FLIP$^{(1)}$ to consist of (4.11)–(4.18) and all of the 2-step constraints. Then the calculations analogous to those used to generate Table 4.1 were repeated to generate Table 4.2 below.

Table 4.2: Dimension study after appending 2-step constraints

N	#Decision Variables	$\mathbf{rank}(A_0^{(1)})$	$\mathbf{dim}(\mathbf{conv}\,(\text{FLIP}^{\,(1)}))$	$\mathbf{dim}(\text{FLIP}^{(1)})$	#Unidentified Equality Constraints
2	4	4	0	0	0
3	18	17	1	1	0
4	72	67	5	5	0
5	220	189	22	31	9
6	540	408	95	132	37
7	1134	749	316	385	69
8	2128	1241	859	887	28
9	3672	1909	1763	1763	0
10	5940	2782	3158	3158	0
11	9130	3885	5245	5245	0
12	13464	5247	8217	8217	0
13	19188	6893	12295	12295	0
14	26572	10466	16106	16106	0

We note that the entries in the column corresponding to $\mathbf{rank}\left(A_0^{(1)}\right)$ have increased for all $N \geq 5$. Similarly, the entries in the column $\mathbf{dim}\left(\mathrm{FLIP}^{(1)}\right)$ have decreased for all $N \geq 4$. For large N these changes are quite substantial. Arguably, this is counter-intuitive because one would expect the under-determination to grow with the increasing size of the problem. This leads to the following conjecture.

Conjecture 4.1. For $N \geq 9$, following holds

$$\mathbf{dim}(\mathbf{conv}(\mathrm{FIP})) = \mathbf{dim}\left(\mathrm{FLIP}^{(1)}\right).$$

If true, Conjecture 4.1 suggests that all non-redundant equality constraints can be obtained from the parameter-free model with the help of a single iteration of the ECA-method of Section 4.1.

Consequently, we also investigated the case of $N \leq 8$ with a view of identifying generalisable equality constraints. For instance, in the case of $N = 6$ we discovered the following new constraint

$$\sum_{k=1}^{\frac{N}{2}} \sum_{\substack{i=1 \\ i \neq k}}^{\frac{N}{2}} \sum_{\substack{a=1 \\ a \neq k,i}}^{\frac{N}{2}} x^k_{\frac{N}{2}-1,ia} = \sum_{k=1+\frac{N}{2}}^{N} \sum_{\substack{i=1+\frac{N}{2} \\ i \neq k}}^{N} \sum_{\substack{a=1+\frac{N}{2} \\ a \neq k,i}}^{N} x^k_{\frac{N}{2},ia}. \tag{4.19}$$

The constraint (4.19) was identified by solving the integer programming model (4.6)–(4.10) (for the case $N = 6$) and examining the pattern of zeros and ones in the solution. This constraints also holds for $N = 8$ but not for any other N. However, interestingly, the following inequality relaxation of (4.19) holds for all even N

$$-1 \leq \sum_{k=1}^{\frac{N}{2}} \sum_{\substack{i=1 \\ i \neq k}}^{\frac{N}{2}} \sum_{\substack{a=1 \\ a \neq k,i}}^{\frac{N}{2}} x^k_{\frac{N}{2}-1,ia} - \sum_{k=1+\frac{N}{2}}^{N} \sum_{\substack{i=1+\frac{N}{2} \\ i \neq k}}^{N} \sum_{\substack{a=1+\frac{N}{2} \\ a \neq k,i}}^{N} x^k_{\frac{N}{2},ia} \leq 1.$$

An interpretation of the above findings could be that in small instances of IP models, a number of relations may exist which can be expressed as non-redundant equality constraints, but those equality constraints are not valid for the larger instances of the underlying problem. This example is a good illustration of the need to consider instances that are sufficiently large.

4.5 Applications

One important application of the technique developed in Section 4.3 is to strengthen existing integer programming formulations of important operations research problems. In particular, it is possible that - with the additional equality constraints identified by our method - the strengthened IP formulations and their corresponding LP-relaxations will provide better quality solutions.

We illustrate this by considering the well-known Desrochers and Laporte (DL) model [12] for solving the asymmetric TSP. The latter model was compared to a model due to Sherali and Driscoll (SD) [35]. It was shown that in terms of objective function values of LP-relaxation the SD model dominates the DL model. This raises the question of whether such domination could be reversed by strengthening the DL model with additional equality constraints identified by the method of Section 4.3. The DL model is as follows:

$$\text{Minimise} \quad \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} c_{ij} x_{ij}$$

Subject to

$$\begin{aligned}
&\sum_{\substack{i=1 \\ i \neq k}}^{N} x_{ki} = 1, && k = 1, \ldots, N, \\
&\sum_{\substack{k=1 \\ i \neq k}}^{N} x_{ki} = 1, && i = 1, \ldots, N, \\
&U_1 = 0, && \\
&U_j \geq (U_i + 1) - (N-1)(1 - x_{ij}) + (N-3)x_{ji} && i \neq j = 2, \ldots, N, \\
&2 - x_{1j} + (N-3)x_{j1} \leq U_j \leq N - (N-3)x_{1j} + x_{j1} - 2, && j = 2, \ldots, N. \\
&x_{ik} \geq 0, && i \neq k = 1, \ldots, N,
\end{aligned} \tag{DL}$$

where U_i denotes the rank order when vertex i is visited with the base city being assigned a rank of zero.

One way to tighten the model would be to add additional variables, with approprite linking constraints. However, it is not necessarily obvious how bets to do this. The ECA-method provides a systematic method for doing so, and as will be shown below, the resulting extended model is stronger than before.

Suppose we add $N^2 - N$ variables to the DL model, with the intention of replacing U_i variables with U_{ki} variables, denoting the rank order of visiting vertex i on a tour starting at vertex k. After the application of our method the extended DL model becomes

$$
\begin{aligned}
&\text{Minimise} \quad \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} c_{ij} x_{ij} \\
&\text{Subject to} \\
&\sum_{\substack{i=1 \\ i \neq k}}^{N} x_{ki} = 1, && k = 1, \ldots, N, \\
&\sum_{\substack{k=1 \\ i \neq k}}^{N} x_{ki} = 1, && i = 1, \ldots, N, \\
&U_{kk} = 0, && k = 1, \ldots, N, \qquad \text{(EDL)} \\
&U_{kj} \geq (U_{ki} + 1) - (N-1)(1 - x_{ij}) + (N-3)x_{ji} && k \neq i \neq j = 1, \ldots, N, \\
&2 - x_{kj} + (N-3)x_{jk} \leq U_{kj} \leq N - (N-3)x_{kj} + x_{jk} - 2, && k \neq j = 1, \ldots, N, \\
&U_{ki} + U_{ik} = N, && i \neq k = 1, \ldots, N, \\
&\sum_{i=1}^{N} U_{ki} = \frac{N(N-1)}{2}, && k = 1, \ldots, N, \\
&x_{ik} \geq 0, && i \neq k = 1, \ldots, N.
\end{aligned}
$$

We then compared the performance of LP-relaxation of EDL versus the original DL model and SD model. We tested their performance on all examples of ATSP listed on TSBLIB [34]. It is important to note that, despite adding $N^2 - N$ variables, the number of decision variables in EDL is still fewer than in SD model. The results are summarised in Table 4.3. Note that, EDL matches or outperforms SD in eleven out of nineteen listed problems, exactly matches SD in five instances and is outperformed by SD in the remaining three instances. Thus it is clear that our technique has considerably strengthened the DL model.

Table 4.3: LP-relaxation comparison study

Problem	DL	EDL	SD	OPT*
br17	22	22	27.679	39
ft-53	6011.875	6044.673	6118.404	6905
ft-70	38333.543	38409.767	38364.552	38673
ftv-33	1217.182	1224.86	1224.504	1286
ftv-35	1413.5	1425.866	1415.512	1473
ftv-38	1477.155	1482.2	1480.055	1530
ftv-44	1573.75	1582.01	1573.75	1613
ftv-47	1725.657	1733.162	1727.208	1776
ftv-55	1510.733	1534.274	1513.27	1608
ftv-64	1761	1765.3	1765.3	1839
ftv-70	1858.533	1864.162	1859.577	1950
ftv-170	2698.472	2703.633	2698.679	2755
kro124p	34976.667	35512.509	35059.582	36230
p43	216	216	864.581	5620
rbg323	1326	1326	1326	1326
rbg358	1163	1163	1163	1163
rbg403	2465	2465	2465	2465
rbg443	2720	2720	2720	2720
ry48p	13809.168	13837.595	13820.433	14422

Chapter 5

Structural Equality Constraints for Cubic Graphs

In this chapter[1], we focus on connected undirected cubic graphs. A cubic graph is one in which every vertex has degree three. We recall that HCP is still NP-complete even when considered only on cubic graphs [24].

Let $\mathcal{Q}_c$ be the polytope which is the convex hull of all HCs that belong to a complete graph on N vertices. A key goal of the methods for TSP which are designed based on polyhedra and integer programing is to construct a good approximation for the polytope $\mathcal{Q}_c$. In particular, the feasible regions of LP-relaxations of all integer programming formulations for TSP can be considered as approximations of this polytope. However, in most real-world TSP applications and also in the case of HCP, we deal with non-complete graphs. Therefore, we would ideally like to approximate the polytope which is the convex hull of all HCs that belong to a given graph G. We shall call this polytope $\mathcal{Q}(G)$.

To the best of our knowledge, there is no standard approximation for $\mathcal{Q}(G)$, and $\mathcal{Q}_c$ often plays that role. However, $\mathcal{Q}_c$ can be a very poor approximation for $\mathcal{Q}(G)$, especially when the graph G is sparse. One should note that, the number of HCs

[1]The main results of this chapter is under review in a journal for publication [31].

of a given cubic graph G can be as small as three [36] while the number of HCs of the complete graph is $(N-1)!$. Hence, the dimension of $\mathcal{Q}(G)$ may be much smaller than the dimension of $\mathcal{Q}_c$. Furthermore, appending each non-redundant equality constraint to a polytope decreases the dimension of that polytope by one. That is, there may exist many new equality constraints, depending on the special structure of G which we call *structural equality constraints*, that could be appended to the already available constraints that were used to approximate $\mathcal{Q}_c$.

Recall that in Section 4.4, the case of the complete graph was considered. We generated all of the equality constraints for different sizes of the complete graph, and then by generalising the patterns in the generated constraints, we extracted a number of new equality constraints. Although it might seen as sensible to use an analogous process to extract all the equality constraints for the set of HCs of a particular cubic graph, the set of equality constraints so produced would only be meaningful for that graph and would not be extendable to the general constraints for all cubic graphs.

Hence, the natural question that arises is: *How can we generate new equality constraints that take advantage of cubicity in some fashion, but are still valid for all cubic graphs?* This issue is focus of the rest of this chapter.

5.1 Extracting structural equality constraints

Before going into details, some definitions and notation are needed. Let the complement set of $\mathcal{A}(i)$, the set of neighbours of i, be denoted by $\mathcal{A}'(i)$, which contains the set of vertices that are not accessible from i in a single step and excluding i as well. Also, let $H(G)$ denote the set of HCs of a given graph G and K_N denote the complete graph on N vertices. The following lemma plays a crucial role in exploiting the structure of the graph to find a better approximation for $\mathcal{Q}(G)$.

Theorem 5.1.1. *Let G and $\mathcal{G}$ be two given graphs on the same set of vertices and G be a subgraph of $\mathcal{G}$, that is, $(G \subseteq \mathcal{G})$, then the following holds:*

$$\mathcal{Q}(G) \subseteq \boldsymbol{conv}\,(H(\mathcal{G}))\,.$$

Proof. As G is a subgraph of $\mathcal{G}$, all of the edges in G also belong to $\mathcal{G}$. It is clear that any HC of G exists in $\mathcal{G}$ as well, that is, $H(G) \subseteq H(\mathcal{G})$. Therefore, we have $\mathcal{Q}(G) = \mathbf{conv}(H(G)) \subseteq \mathbf{conv}(H(\mathcal{G}))$. ∎

We now, immediately conclude two following corollaries from the above theorem:

Corollary 5.1.2. *If $G \subseteq \mathcal{G}$, any valid constraint used to represent $\boldsymbol{conv}(H(\mathcal{G}))$ is also a constraint that can be used to represent $\mathcal{Q}(G)$.*

Corollary 5.1.3. *If G is a subgraph of graphs $\mathcal{G}_1, \ldots, \mathcal{G}_m$, we have*

$$\mathcal{Q}(G) \subseteq \bigcap_{\mathcal{L}=1}^{m} \boldsymbol{conv}(H(\mathcal{G}_{\mathcal{L}})).$$

One can see from Corollary 5.1.2 that any constraint used to represent $\mathcal{Q}_c$ will be satisfied by any point in $\mathcal{Q}(G)$, since any given graph G on N vertices is a subgraph of K_N. However, depending on the structure of the given graph G, $\mathcal{Q}_c$ may be a very poor approximation of $\mathcal{Q}(G)$. Moreover, characterising the polytope $\mathcal{Q}(G)$ may be as difficult as finding all HCs of the graph G.

Although we do not know all HCs of the graph G, we may eliminate consideration of many HCs which do not belong to graph G. For example, all of the HCs that contain an edge not belonging to graph G clearly could be eliminated. However, our aim is to find equality constraints which are applicable for any given cubic graph. Therefore, we need to remove some HCs (not belonging to graph G) so that the convex hull of remaining HCs has special properties. That is, we need to keep some HCs which do not belong to grah G in order to preserve generalisability of the generated constraints. This is achieved by constructing a subgraph of the complete graph that, in a special sense, contains all cubic graphs.

In particular, let $\mathcal{C}_N$ be the set of all connected cubic graphs on N vertices. We wish to construct a graph U_N on N vertices that has the following two desirable properties:

(a) for every graph $G \in \mathcal{C}_N$ there exists a graph isomorphism $\mathcal{M}$ such $G \subset \mathcal{M}(U_N)$. In such a case we say that G is *isomorphically contained* in U_N or alternatively U_N *covers* G *isomorphically*;

(b) the construction of U_N is generalisable in an obvious and unique way for all N.

Indeed, property (a), above defines a *Universal* graph for $\mathcal{C}_N$. That is, for any $G \in \mathcal{C}_N$ we have $G \subset \mathcal{M}(U_N)$, for some graph isomorphism $\mathcal{M}$. Hence, by exploiting the Corollary 5.1.2 we arrive at

$$\mathcal{Q}(G) \subset \mathbf{conv}(H(\mathcal{M}(U_N))).$$

Therefore, we want to be able to construct $\mathbf{conv}(H(\mathcal{M}(U_N)))$ or a good approximation of $\mathbf{conv}(H(\mathcal{M}(U_N)))$ to find a set of new equality constraints for $\mathcal{Q}(G)$ without knowing any information about HCs of any particular G. Property (b) is also important as it is necessary to ensure any new constraints we subsequently discover can be generalised to larger N.

In Figure 5.1, we demonstrate an example of such a universal graph U_N for $N = 6$.

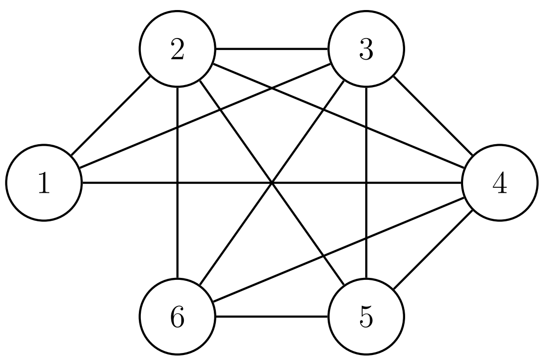

Figure 5.1: Universal graph U_6

First we take the complete graph K_N, then we construct a graph by removing

$N-4$ edges $((1,4),(1,5),\ldots,(1,N))$ incident on the vertex 1. We shall denote the resulting graph by U_N. Figure 5.1 demonstrates this graph in the case when $N=6$.

Remark 5.1.4. *Even though U_N is only marginally sparser than K_N, we will see that its consideration will lead to many new constraints. Note that, U_N contains $6(N-4)!$ HCs which is only $\frac{6}{(N-1)(N-2)(N-3)}$ as many HCs as are present in K_N. Obviously sparser choices of U_N will lead to even greater opportunities.*

Now, suppose our given graph $G \in \mathcal{C}_6$ is the "envelope" graph, which is labelled as shown in Figure 5.2.

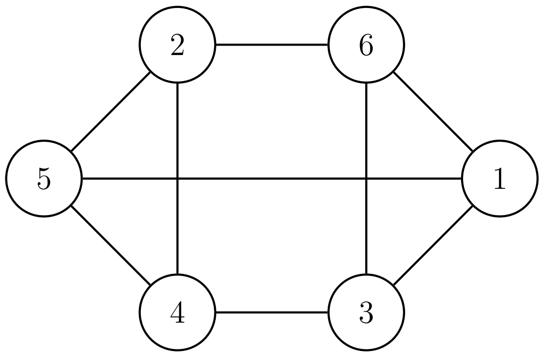

Figure 5.2: Envelope graph

Then, we can construct an isomorphic map $\mathcal{M}$ such that $G \subset \mathcal{M}(U_6)$. For example, we could choose $\mathcal{M} : \{1,2,3,4,5,6\} \to \{4,3,5,2,6,1\}$.

Note that one way of obtaining a subset of constraints which characterise $\mathbf{conv}(H(\mathcal{M}(U_N)))$ is to find a subset of constraints which characterise $\mathbf{conv}(H(U_N))$ and then relabel the variables of those constraints based on the map $\mathcal{M}$.

To illustrate the above discussion, let $x^2_{2,34} = x^1_{1,34}$ be a new constraint to characterise $\mathbf{conv}(H(U_6))$. Then the constraint $x^{\mathcal{M}(2)}_{2,\mathcal{M}(3)\mathcal{M}(4)} = x^{\mathcal{M}(1)}_{1,\mathcal{M}(3)\mathcal{M}(4)}$, $(x^3_{2,52} = x^4_{1,52})$ will be the corresponding new constraint for the envelope graph labelled as above.

Of course, the isomorphism $\mathcal{M}$ specified above is not the only way that U_6 can cover the envelope graph. Indeed, there are many such isomorphisms. To obtain all of these isomorphisms, we first map vertex 1 of U_N to an arbitrary vertex z of the envelope graph, that is, $z = \mathcal{M}(1)$. After mapping vertex 1 of U_6, there are 3!2! (3!($N-4$)!) possible isomorphisms. These correspond to 3! ways of mapping the three neighbours of vertex 1 of U_6 to the neighbours of vertex z in the envelope graph, and 2! (($N-4$)!) ways of mapping the remaining vertices of U_6 to those of the envelope graph. For instance, after mapping 1 to 4, $\hat{\mathcal{M}} : \{1, 2, 3, 4, 5, 6\} \to \{4, 2, 3, 5, 1, 6\}$ is another one of these 12 isomorphisms. In particular, this means that the number of isomorphisms of U_N that can cover any given cubic graph is $6N(N-4)!$ since the vertex z can be any vertex of the given graph.

Obviously, the constraint so discovered depends on the map chosen, and using a different map leads to finding a different constraint. One may be interested in rewriting the constraints based on different relabellings to gain more constraints. Interestingly, this can only be achieved by considering the maps that can be obtained through the 3! ways that the neighbour vertices of the vertex 1 of U_N can be mapped to the neighbours of z in the given cubic graph. It will be shown that $(N-4)!$ permutations of remaining vertices do not produce any new constraints. In general, we can write $6N$ new constraints for the given cubic graph based on each constraint associated with $\mathbf{conv}(H(U_N))$. More precisely, by exploiting Corollary 5.1.3, we arrive at

$$\mathbf{conv}(H(G)) \subset \bigcap_{\mathcal{L}=1}^{6N} \mathbf{conv}(H(\mathcal{M}_{\mathcal{L}}(U_N))). \tag{5.1}$$

Based on the above discussion, we only need to find a set of constraints to characterise $\mathbf{conv}(H(U_N))$. However, it is still very hard to find a set of constraints which fully characterises the polytope $\mathbf{conv}(H(U_N))$. We recall from Section 4.2 that having many extended variables results in having more equality constraints and less inequality constraints (facets) to characterise a convex hull of a given set of HCs. Recall the parameter-free model introduced in Section 2.3, since there are many extended variables in that model, it is reasonable to just concentrate on finding the equality constraints in those variables, rather than all constraints that characterise

$\mathbf{conv}(H(U_N))$. That is, we try to find a subset of constraints which characterise $\mathbf{aff}(H(U_N))$ instead of $\mathbf{conv}(H(U_N))$.

We recall from Section 2.4 that the existing constraints of the refined parameter-free polytope $\bar{\mathcal{P}}$ provide a number of structured constraints needed to represent $\mathbf{aff}(H(U_N))$. We exploit the ECA-method introduced in Chapter 4.3 to extract unidentified equality constraints to represent $\mathbf{aff}(H(U_N))$.

We then consider $\mathcal{M}$ as a general isomorphism such that $G \subset \mathcal{M}(U_N)$ for any given $G \in \mathcal{C}_N$, and then transform those newly extracted constraints to the constraints for $\mathbf{aff}(\mathcal{M}(H(U_N)))$ and call them the set of *cubic structural equality constraints* (CSEC). In order to achieve this generalisation, we assume vertex z of a given cubic graph G is adjacent to vertices i, j, and k. That is, $\mathcal{A}(z) = \{i, j, k\}$. We then map vertices $1, 2, 3, 4$ of U_N into vetices z, i, j, k respectively. We also map vertices $5, \ldots, N$ of U_N into the vertices belonging to $\mathcal{A}'(z)$ with an arbitrary order. This generalisation allows us to write the constraints for all choices of z and $3!$ permutations of i, j, and k.

This procedure results in a rather large set CSEC. Fortunately, the constraints of that set which we were able to discover can be classified into 31 types of constraints, where each type can be given a natural interpretation. Note that we are not claiming that these 31 types of constraints are mutually exclusive.

Those 31 types of constraints of CSEC are listed for all $z = 1, \ldots, N$, and $i, j, k \in \mathcal{A}(z)$, $i \neq j \neq k$, as follows

$$x^z_{r,ij} = 0 \qquad r = 2, \ldots, N-3; \tag{5.2}$$

$$x^i_{r,jk} = 0 \qquad r = 1, 3, \ldots, N-4; \tag{5.3}$$

$$x^i_{2,jk} = x^z_{1,jk}; \tag{5.4}$$

$$x^j_{N-3,ki} = x^z_{N-2,ki}; \tag{5.5}$$

$$x^z_{2,ia} = x^j_{1,ia} + x^k_{1,ia} \qquad a \in \mathcal{A}'(z); \tag{5.6}$$

$$x^i_{2,ka} + x^j_{2,ka} = x^z_{1,ka} + x^z_{3,ka} \qquad a \in \mathcal{A}'(z); \tag{5.7}$$

$$x^j_{N-3,ai} + x^k_{N-3,ai} = x^z_{N-4,ai} + x^z_{N-2,ai} \qquad a \in \mathcal{A}'(z); \tag{5.8}$$

$$x^z_{N-2,ij} + x^i_{2,zj} = \sum_{a \in \mathcal{A}'(z)} x^z_{N-3,ai}; \tag{5.9}$$

$$x^i_{r,jz} + x^i_{r+1,zj} = \sum_{a \in \mathcal{A}'(z)} x^z_{N-2-r,ai} \qquad r = 2, \ldots, N-5; \tag{5.10}$$

$$x^i_{3,jz} + x^j_{N-4,zi} = \sum_{a \in \mathcal{A}'(z)} x^i_{2,aj}; \tag{5.11}$$

$$\sum_{a \in \mathcal{A}'(z)} x^i_{1,aj} + x^j_{N-4,iz} = \sum_{a \in \mathcal{A}'(z)} x^z_{2,aj} + x^i_{2,jz}; \tag{5.12}$$

$$\sum_{a \in \mathcal{A}'(z)} x^i_{2,aj} + x^j_{N-5,iz} = \sum_{a \in \mathcal{A}'(z)} x^z_{3,aj} + x^i_{3,jz}; \tag{5.13}$$

$$\sum_{a \in \mathcal{A}'(z)} x^z_{N-5,ai} + x^k_{N-4,zi} = \sum_{a \in \mathcal{A}'(z)} x^i_{2,ak} + x^i_{3,jz}; \tag{5.14}$$

$$\sum_{a \in \mathcal{A}'(z)} x^i_{2,ak} + x^i_{3,jz} + x^k_{N-5,iz} = \sum_{a \in \mathcal{A}'(z)} x^z_{3,ak} + \sum_{a \in \mathcal{A}'(z)} x^z_{N-5,ai}; \tag{5.15}$$

$$\sum_{a \in \mathcal{A}'(z)} x^i_{1,ak} + x^i_{2,jz} + x^k_{N-4,iz} = \sum_{a \in \mathcal{A}'(z)} x^z_{2,ak} + \sum_{a \in \mathcal{A}'(z)} x^z_{N-4,ai}; \tag{5.16}$$

$$\sum_{a \in \mathcal{A}'(z)} x^z_{2,aj} + x^i_{2,jz} + x^i_{4,jz} + x^j_{N-5,zi} = \sum_{a \in \mathcal{A}'(z)} x^i_{1,aj} + \sum_{a \in \mathcal{A}'(z)} x^i_{3,aj}; \tag{5.17}$$

$$\sum_{a \in \mathcal{A}'(z)} x^z_{3,aj} + x^i_{3,jz} + x^i_{5,jz} + x^j_{N-6,zi} = \sum_{a \in \mathcal{A}'(z)} x^i_{2,aj} + \sum_{a \in \mathcal{A}'(z)} x^i_{4,aj}; \tag{5.18}$$

$$\sum_{a \in \mathcal{A}'(z)} x^i_{1,aj} + \sum_{a \in \mathcal{A}'(z)} x^i_{3,aj} + x^j_{N-6,iz} = \sum_{a \in \mathcal{A}'(z)} x^z_{2,aj} + \sum_{a \in \mathcal{A}'(z)} x^z_{4,aj} + x^i_{2,jz} + x^i_{4,jz}; \tag{5.19}$$

$$\sum_{a \in \mathcal{A}'(z)} x^i_{2,aj} + \sum_{a \in \mathcal{A}'(z)} x^i_{4,aj} + x^j_{N-7,iz} = \sum_{a \in \mathcal{A}'(z)} x^z_{3,aj} + \sum_{a \in \mathcal{A}'(z)} x^z_{5,aj} + x^i_{3,jz} + x^i_{5,jz}; \tag{5.20}$$

$$\begin{aligned} &\sum_{a \in \mathcal{A}'(z)} x^z_{2,ak} + \sum_{a \in \mathcal{A}'(z)} x^z_{N-6,ai} + \sum_{a \in \mathcal{A}'(z)} x^z_{N-4,ai} + x^k_{N-5,zi} = \\ &\qquad \sum_{a \in \mathcal{A}'(z)} x^i_{1,ak} + \sum_{a \in \mathcal{A}'(z)} x^i_{3,ak} + x^i_{2,jz} + x^i_{4,jz}; \end{aligned} \tag{5.21}$$

$$\begin{aligned} &\sum_{a \in \mathcal{A}'(z)} x^z_{N-7,ai} + \sum_{a \in \mathcal{A}'(z)} x^z_{3,ak} + \sum_{a \in \mathcal{A}'(z)} x^z_{N-5,ai} + x^k_{N-6,zi} = \\ &\qquad \sum_{a \in \mathcal{A}'(z)} x^i_{2,ak} + \sum_{a \in \mathcal{A}'(z)} x^i_{4,ak} + x^i_{3,jz} + x^i_{5,jz}; \end{aligned} \tag{5.22}$$

$$\sum_{a\in\mathcal{A}(z)} x^z_{2,ma} + x^z_{N-3,im} + \sum_{r=1}^{N-3} x^i_{r,jm} + \sum_{r=3}^{N-3} x^i_{r,km} + \sum_{r=3}^{N-3} x^j_{r,km} = \tag{5.23}$$

$$\sum_{r=1}^{N-4} x^z_{r,jm} + \sum_{r=1,2,3,N-3} x^z_{r,km} + 2\sum_{r=4}^{N-4} x^z_{r,km}$$

$$+ x^i_{1,mj} + x^i_{1,mk} + x^j_{1,mk} \quad m \in \mathcal{A}'(z);$$

$$x^i_{3,jz} + \sum_{a\in\mathcal{A}'(z)} x^i_{N-6,ak} + x^i_{N-5,jz} + \sum_{a\in\mathcal{A}'(z)} x^j_{N-6,ak} = \tag{5.24}$$

$$\sum_{a\in\mathcal{A}'(z)} x^z_{3,ai} + \sum_{a\in\mathcal{A}'(z)} x^z_{N-7,ak} + \sum_{a\in\mathcal{A}'(z)} x^z_{N-5,ak} + \sum_{a\in\mathcal{A}'(z)} x^i_{2,aj};$$

$$\sum_{a\in\mathcal{A}'(z)} x^i_{1,ak} + x^i_{2,jz} + \sum_{a\in\mathcal{A}'(z)} x^i_{3,ak} + x^i_{4,jz} + x^k_{N-6,iz} = \tag{5.25}$$

$$\sum_{a\in\mathcal{A}'(z)} x^z_{2,ak} + \sum_{a\in\mathcal{A}'(z)} x^z_{N-6,ai} + \sum_{a\in\mathcal{A}'(z)} x^z_{4,ak} + \sum_{a\in\mathcal{A}'(z)} x^z_{N-4,ai};$$

$$\sum_{a\in\mathcal{A}'(z)} x^i_{2,ak} + x^i_{3,jz} + \sum_{a\in\mathcal{A}'(z)} x^i_{4,ak} + x^i_{5,jz} + x^k_{N-7,iz} = \tag{5.26}$$

$$\sum_{a\in\mathcal{A}'(z)} x^z_{3,ak} + \sum_{a\in\mathcal{A}'(z)} x^z_{N-7,ai} + \sum_{a\in\mathcal{A}'(z)} x^z_{5,ak} + \sum_{a\in\mathcal{A}'(z)} x^z_{N-5,ai};$$

$$\sum_{r=1}^{N-3} x^z_{r,jm} + \sum_{r=1}^{N-3} x^z_{r,km} + \sum_{r=2}^{N-4} x^j_{r,mi} + \sum_{r=2}^{N-4} x^k_{r,mi} = \tag{5.27}$$

$$2\sum_{r=2}^{N-5} x^z_{r,mi} + \sum_{r=N-4}^{N-2} x^z_{r,mi} + \sum_{r=1}^{N-3} x^i_{r,jm} + \sum_{r=1}^{N-3} x^i_{r,km} \qquad m \in \mathcal{A}'(z);$$

$$x^z_{2,mi} + x^z_{2,mk} + x^z_{N-3,im} + \sum_{r=1}^{N-3} x^i_{r,jm} + \sum_{r=1}^{N-3} x^i_{r,km} + \sum_{r=2}^{N-4} x^i_{r,mj} + \sum_{r=2}^{N-4} x^k_{r,mj} = \tag{5.28}$$

$$x^z_{1,jm} + x^z_{1,km} + x^z_{N-2,mj} + \sum_{r=2}^{N-4} x^z_{r,jm} + \sum_{r=2}^{N-4} x^z_{r,km} + 2\sum_{r=3}^{N-5} x^z_{r,mj} +$$

$$\sum_{r=N-4}^{N-3} x^z_{r,mj} + x^z_{2,mj} + x^i_{1,mk} + x^j_{1,mk} \qquad m \in \mathcal{A}'(z);$$

$$\sum_{a\in\mathcal{A}(z)} x^z_{1,am} + \sum_{r=2}^{N-4} \sum_{\substack{g=1 \\ g\neq m,z,j}}^{N} x^z_{r,gm} + x^z_{N-2,mi} + x^z_{N-2,mj} + x^i_{N-3,mk} \tag{5.29}$$

$$+ x^j_{N-3,mk} - x^z_{N-4,mk} = 1 \qquad m \in \mathcal{A}'(z);$$

$$\sum_{a\in\mathcal{A}(z)} x^z_{1,am} + \sum_{r=2}^{N-4} \sum_{\substack{g=1 \\ g\neq m,z,j}}^{N} x^z_{r,gm} + x^z_{N-2,mi} + x^z_{N-2,mj} + \sum_{r=1}^{N-4} x^i_{r,mk} + \tag{5.30}$$

$$\sum_{r=1}^{N-4} x^j_{r,mk} - 2\sum_{r=2}^{N-5} x^z_{r,mk} - x^z_{N-4,mk} - x^z_{N-3,mk} = 1 \qquad m \in \mathcal{A}'(z);$$

$$\sum_{a\in\mathcal{A}(z)} x^z_{2,ma} + x^z_{N-3,im} + \sum_{r=1}^{N-3} x^i_{r,jm} + \sum_{r=3}^{N-3} x^i_{r,km} + \sum_{r=3}^{N-3} x^j_{r,km} = \tag{5.31}$$

$$\sum_{r=1}^{N-4} x^z_{r,jm} + \sum_{r=1,2,3,N-3} x^z_{r,km} + 2\sum_{r=4}^{N-4} x^z_{r,km} + x^i_{1,mj} + x^i_{1,mk} + x^j_{1,mk} \quad m \in \mathcal{A}'(z). \tag{5.32}$$

Theorem 5.1.5. *Rewriting CSEC constraints based on $(N-4)!$ permutations, that vertices $5,\ldots,N$ of U_N can be mapped into the vertices in $\mathcal{A}'(z)$ (non-adjacent vertices of vertex z) of a given cubic graph, does not lead to any new constraints.*

Proof. Note that the CSEC constraints can be classified in three categories.

1) The constraints which do not contain any index from $\mathcal{A}'(z)$ such as Constraints (5.2). Obviously, these kind of constraints do not vary by considering any of those $(N-4)!$ permutations. Thus, changing the map does not lead to any new constraints in this category.

2) The constraints which have only one index in $\mathcal{A}'(z)$ such as Constraints (5.6). Obviously, different permutations may change this index to another index in $\mathcal{A}'(z)$. However, since we already consider all choices of that index, further consideration of those $(N-4)!$ permutations only leads to repeating the constraints of this category.

3) The constraints which include some summations so that those summations enumerate all indices in $\mathcal{A}'(z)$ such as Constraints (5.9). Obviously, writing the terms of a summation in different order has no effect, and so, considering those $(N-4)!$ permutations only repeats the constraints of this category $(N-4)!$ times.

■

As mentioned earlier the above 31 types of constraints have natural interpretations. These vary from very simple ones to quite technical ones. Below we supply explanations of just three types (5.2), (5,4) and (5.28). The others can be explained by derivations that are conceptually similar.

Starting with (5.2) it is helpful to consider the configuration shown in Figure 5.3.

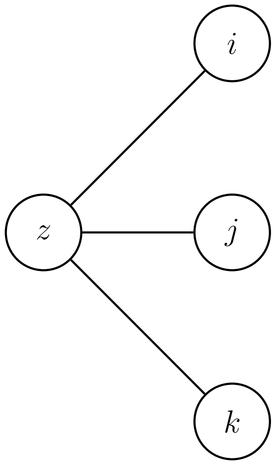

Figure 5.3: A three degree vertex

Recall that any HC passes through z and uses exactly two out of three edges. Thus starting at z one of i, j or k must precede z at $(N-1)^{\text{st}}$ step $(r = N-1)$ and one must follow z at the 1^{st} step $(r = 1)$. Now, by inspection, it is easy to see that for $r = 2, \ldots, N-3$, $x^z_{r,ij} = 0$. For instance, if $x^z_{2,ij} = 1$ were possible that would mean that i is the 2^{nd} vertex on the HC starting at z and j is the 3^{rd} vertex on that cycle. But this implies that vertex k had to be visited on the first step which means that there are no more neighbours of z left to return to it on the last step. This yields a contradiction.

In Constraints (5.4), $x^i_{2,jk} = 1$ if an HC starts at vertex i and arrives at j after 2 steps, followed by transition from j to k. Since vertex z is only adjacent to i, j, k and k is visited after j, z must be preceded by i , and succeeded by j, that is, $x^i_{1,zj} = 1$. Otherwise, this constraint yields $0 = 0$.

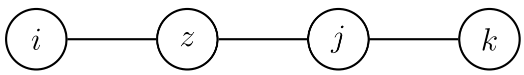

Figure 5.4: Constraints (5.4)

To show that the constraint (5.28) is valid, we let each term of the right hand side of this constraint be non-zero at a time and then analyse its influence on the other terms of the constraint. For example, we assume $x^j_{1,mk} = 1$ (the last term in the right hand side). Since vertex z is only adjacent to i, j, and k, one of the following (i) or (ii) cases can occur.

(i) One configuration that allows $x^j_{1,mk} = 1$ is shown in Figure 5.5, this configuration leads to $x^z_{1,jm} = 1$ and all other terms in right hand side are zero and the only non-negative terms in left hand side are $x^z_{2,mk} = 1$ and $\sum_{r=1}^{N-3} x^i_{r,jm} = 1$. This implies that the Constraint (5.28) is satisfied in the form $2 = x^z_{2,mk} + \sum_{r=1}^{N-3} x^i_{r,jm} = x^j_{1,mk} + x^z_{1,jm} = 2$.

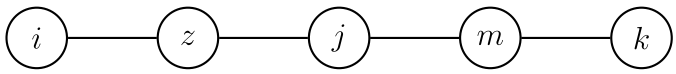

Figure 5.5: Constraints (5.28), case (i)

(ii) In the case displayed in Figure 5.6, when $x^j_{1,mk} = 1$, all other terms in right hand side have zero value and in the left hand side only one of the terms in $\sum_{r=1}^{N-3} x^i_{r,jm}$, namely, $x^i_{N-4,jm} = 1$ will be non-zero. That is, Constraint (5.28) is satisfied in the form $1 = \sum_{r=1}^{N-3} x^i_{r,jm} = x^j_{1,mk} = 1$.

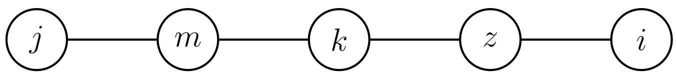

Figure 5.6: Constraints (5.28), case (ii)

Continuing these analyses results in 18 additional cases ((iii) - (xx)) which are demonstrated in the following figures. We show how Constraint (5.28) is satisfied in each case by introducing the non-negative terms of the constraint for each case.

(iii) $1 = x^z_{2,mk} = x^i_{1,mk} = 1$.

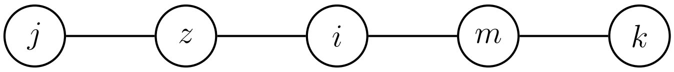

Figure 5.7: Constraints (5.28), case (iii)

(iv) $1 = x^{z}_{N-3,im} = x^{i}_{1,mk} = 1.$

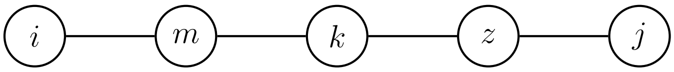

Figure 5.8: Constraints (5.28), case (iv)

(v) $2 = \sum_{r=1}^{N-3} x^{i}_{r,km} + \sum_{r=2}^{N-4} x^{i}_{r,mj} = x^{z}_{2,mj} + x^{z}_{1,km} = 2.$

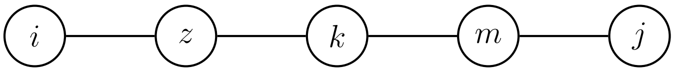

Figure 5.9: Constraints (5.28), case (v)

(vi) $1 = \sum_{r=2}^{N-4} x^{k}_{r,mj} = x^{z}_{2,mj} = 1.$

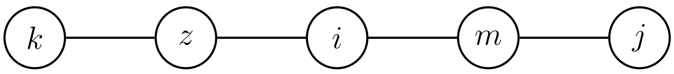

Figure 5.10: Constraints (5.28), case (vi)

(vii) $1 = \sum_{r=2}^{N-4} x^{k}_{r,mj} = \sum_{r=N-4}^{N-3} x^{z}_{r,mj} = 1.$

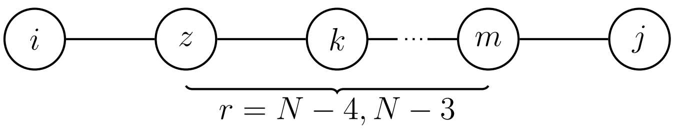

Figure 5.11: Constraints (5.28), case (vii)

(viii) $1 = \sum_{r=2}^{N-4} x^{i}_{r,mj} = \sum_{r=N-4}^{N-3} x^{z}_{r,mj} = 1.$

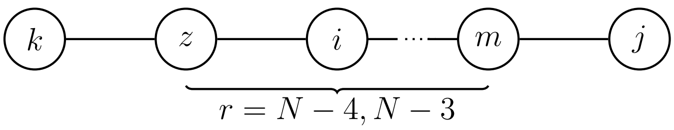

Figure 5.12: Constraints (5.28), case (viii)

(ix) $2 = \sum_{r=2}^{N-4} x_{r,mj}^{i} + \sum_{r=2}^{N-4} x_{r,mj}^{k} = 2 \sum_{r=3}^{N-5} x_{r,mj}^{z} = 2.$

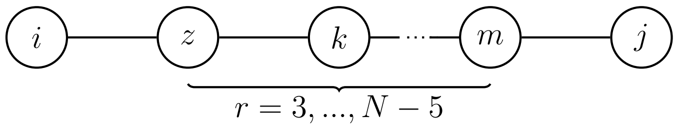

Figure 5.13: Constraints (5.28), case (ix)

(x) $2 = \sum_{r=2}^{N-4} x_{r,mj}^{i} + \sum_{r=2}^{N-4} x_{r,mj}^{k} = 2 \sum_{r=3}^{N-5} x_{r,mj}^{z} = 2.$

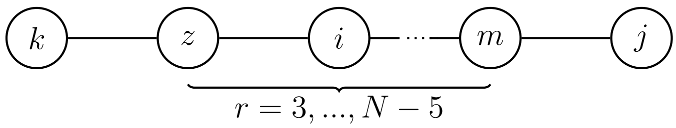

Figure 5.14: Constraints (5.28), case (x)

(xi) $1 = \sum_{r=1}^{N-3} x_{r,km}^{i} = \sum_{r=2}^{N-4} x_{r,km}^{z} = 1.$

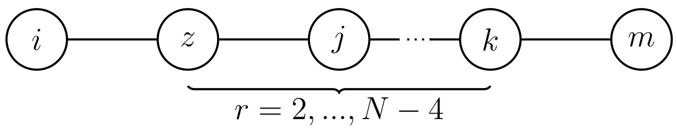

Figure 5.15: Constraints (5.28), case (xi)

(xii) $1 = \sum_{r=1}^{N-3} x_{r,km}^{i} = \sum_{r=2}^{N-4} x_{r,km}^{z} = 1.$

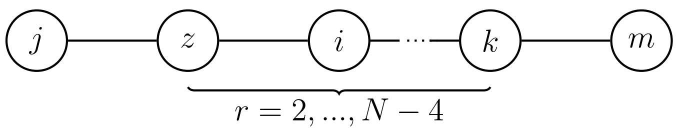

Figure 5.16: Constraints (5.28), case (xii)

(xiii) $1 = \sum_{r=1}^{N-3} x_{r,jm}^{i} = \sum_{r=2}^{N-4} x_{r,jm}^{z} = 1.$

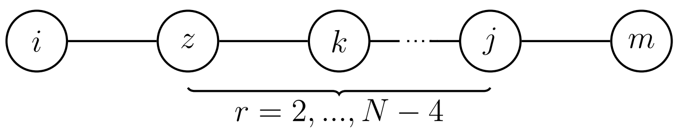

Figure 5.17: Constraints (5.28), case (xiii)

(xiv) $1 = \sum_{r=1}^{N-3} x^i_{r,jm} = \sum_{r=2}^{N-4} x^z_{r,jm} = 1.$

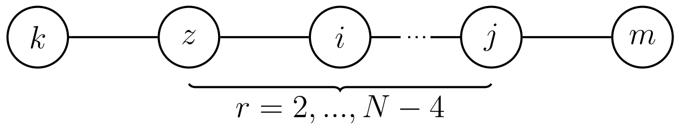

Figure 5.18: Constraints (5.28), case (xiv)

In case (xv), based on the position of vertex k, only one of the following terms in left hand side can be non-zero. More precisely, if k appears one step before m in an HC, then $\sum_{r=1}^{N-3} x^i_{r,km} = 1$, and $\sum_{r=2}^{N-4} x^k_{r,mj} = 0$ and if k visited in any other position rather than one step before m, then $\sum_{r=1}^{N-3} x^i_{r,km} = 0$ and $\sum_{r=2}^{N-4} x^k_{r,mj} = 1$.

(xv) $1 = \sum_{r=1}^{N-3} x^i_{r,km} + \sum_{r=2}^{N-4} x^k_{r,mj} = x^z_{N-2,mj} = 1.$

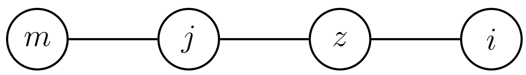

Figure 5.19: Constraints (5.28), case (xv)

An analogous reasoning for the configuration shown in Figure 5.20 leads to

(xvi) $1 = x^z_{N-3,im} + \sum_{r=2}^{N-4} x^i_{r,mj} = x^z_{N-2,mj} = 1.$

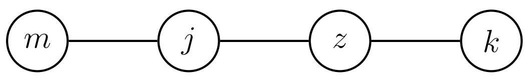

Figure 5.20: Constraints (5.28), case (xvi)

In case (xvii), if vertex j is preceded exactly one step after m, then $\sum_{r=1}^{N-3} x^i_{r,km} = 1$, and $\sum_{r=2}^{N-4} x^i_{r,mj} = 1$ in the left side of the constraints and accordingly the terms $x^z_{2,mj}$, and $x^z_{1,km}$ in the right hand side take value 1. Otherwise if j is not visited in exactly one step after m, then $\sum_{r=1}^{N-3} x^i_{r,km} = 1$, and $\sum_{r=2}^{N-4} x^i_{r,mj} = 0$ in the left side and in the right side, we see $x^z_{2,mj} = 0$, and $x^z_{1,km} = 1$.

(xvii) $1 \text{ or } 2 = \sum_{r=1}^{N-3} x^i_{r,km} + \sum_{r=2}^{N-4} x^i_{r,mj} = x^z_{2,mj} + x^z_{1,km} = 1 \text{ or } 2.$

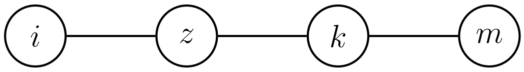

Figure 5.21: Constraints (5.28), case (xvii)

In case (xviii), if vertex i is preceded exactly one step after m, then $x^z_{2,mi} = 1$, and $\sum_{r=1}^{N-3} x^i_{r,km} = 0$. Otherwise if j is not visited in exactly one step after m, then $x^z_{2,mi} = 0$, and $\sum_{r=1}^{N-3} x^i_{r,km} = 1$.

(xviii) $1 = x^z_{2,mi} + \sum_{r=1}^{N-3} x^i_{r,km} = x^z_{1,km} = 1$.

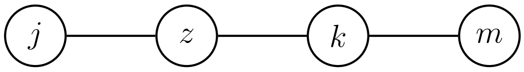

Figure 5.22: Constraints (5.28), case (xviii)

Obviously, in case (xix) we see $x^z_{1,jm} = 1$ and $\sum_{r=1}^{N-3} x^i_{r,jm} = 1$. Also, if vertex k is preceded exactly one step after m, then $x^j_{1,mk} = 1$, and $x^z_{2,mk} = 1$. Otherwise if k is not visited in exactly one step after m, then $x^j_{1,mk} = 0$, and $x^z_{2,mk} = 0$.

(xix) $1 \text{ or } 2 = x^z_{2,mk} + \sum_{r=1}^{N-3} x^i_{r,jm} = x^z_{1,jm} + x^j_{1,mk} = 1 \text{ or } 2$.

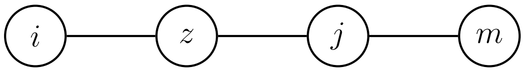

Figure 5.23: Constraints (5.28), case (xix)

Similar reasoning applies to case (xx). If vertex i is preceded exactly one step after m, then $x^z_{2,mi} = 1$, and $\sum_{r=1}^{N-3} x^i_{r,jm} = 0$. Otherwise if i is not visited in exactly one step after m, then $x^z_{2,mk} = 0$, and $\sum_{r=1}^{N-3} x^i_{r,jm} = 1$.

(xx) $1 = x^z_{2,mi} + \sum_{r=1}^{N-3} x^i_{r,jm} = x^z_{1,jm} = 1$.

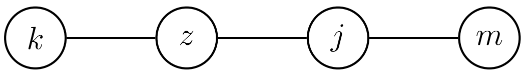

Figure 5.24: Constraints (5.28), case (xx)

5.2 Comparison of performance

We recall that $\mathcal{Q} := \mathcal{Q}(G)$ is a polytope constructed as the convex hull of HCs of a given graph G. Obviously, polytope $\mathcal{Q}$ is empty when G is a non-Hamiltonian graph and we are hopeful that the polytope $\hat{\mathcal{Q}}$ which is constructed as an approximation of $\mathcal{Q}$ will be empty as well. That is, the constraints that represent $\hat{\mathcal{Q}}$ should be infeasible for a non-Hamiltonian graph.

Since $\mathcal{Q}$ is a subset of $\hat{\mathcal{Q}}$, the emptiness of $\hat{\mathcal{Q}}$ implies that the graph G is non-Hamiltonian. However, whenever polytope $\hat{\mathcal{Q}}$ is non-empty, the result is inconclusive. Hence, the following natural question arises: if we assume that a non-empty polytope $\hat{\mathcal{Q}}$ implies Hamiltonicity, how frequently is this diagnosis incorrect? Filar et al. [19] proved that, in the case of bridge graphs[2], the refined parameter-free polytope $\bar{\mathcal{P}}$ is always empty. Therefore, we only compare $\bar{\mathcal{P}}$ with the new constructed model over non-bridge non-Hamiltonian (NBNH for short) cubic graphs. The new constructed model $\bar{\mathcal{P}}_{\text{CSEC}}$ includes CSEC constraints (5.2)-(5.32) and the constraints which represent $\bar{\mathcal{P}}$ (2.25)-(2.31). The results are reported in Table 5.1.

In Table 5.1, from the left, the first column shows the number of nodes in the family of cubic graphs, the second columns displays the total number of NBNH graphs in that particular family, the third and fifth columns display the number of such NHNB graphs which had empty polytopes, $\bar{\mathcal{P}}$ and $\bar{\mathcal{P}}_{\text{CSEC}}$, respectively, and the fourth and six columns gives the ratio of the correct detection by $\bar{\mathcal{P}}$ and $\bar{\mathcal{P}}_{\text{CSEC}}$ respectively.

Table 5.1: Solving HCP for non-Hamiltonian cubic graphs varying from 10-18 nodes

n	# NBNH	# Detected by $\bar{\mathcal{P}}$	Ratio by $\bar{\mathcal{P}}$	# Detected by $\bar{\mathcal{P}}_{\text{CSEC}}$	Ratio by $\bar{\mathcal{P}}_{\text{CSEC}}$
10	1	0	0	1	1.0
12	1	0	0	1	1.0
14	6	1	0.167	6	1.0
16	33	6	0.182	33	1.0
18	231	42	0.182	231	1.0

[2] A graph is called a *bridge graph*, if the set $\mathcal{V}$ can be partitioned into two non-empty sets such that there is only one arc from one partition to the other one.

As seen in the above table, for cubic graphs containing up to 18 vertices, 223 out of 272 undirected connected cubic graphs were so misdiagnosed by $\bar{\mathcal{P}}$. In contrast, the new constructed polytope $\bar{\mathcal{P}}_{\mathrm{CSEC}}$ was successful at diagnosing non-Hamiltonicity of all of tested instances. Finally, it should be noted that promising results indicated in Table 5.1 strongly suggest the need for further work on extracting more structural equality constraints.

Chapter 6

Conclusion and Future Work

In this thesis, we reviewed MDPs and the space of discounted occupational measures when applied to HCP and TSP for a given graph. We also reviewed the parameter-free model as an extension of the space of discounted occupational measures and then refined this model by removing a number of redundant constraints. Accordingly, we showed that all variables of this model are forced to be binary even if only the initial $\mathcal{O}(N^2)$ variables are constrained to be binary. Following that, we proved that if all variables of the parameter-free model are binary the feasible space of this model corresponds precisely to the set of Hamiltonian cycles of the associated graph.

In Chapter 3 we considered the convex combination of a Hamiltonian cycle policy and its reverse. We showed that the resolvent-like matrix induced by this combined policy can be expanded in terms of finitely many powers of the probability transition matrix corresponding to the underlying Hamiltonian cycle. We derived closed-form formulae for the coefficients of these powers which were reduced to expressions involving the classical Chebyshev polynomials of the second kind.

In Chapter 4 we developed a generic method to discover and generate unidentified equality constraints that can be used to refine feasible regions of LP-relaxations of integer programming problems, and demonstrated this model on the parameter-free model which resulted in identification of the 2-step constraints in Section 4.4.

In Chapter 5 we embedded cubic graphs in suitably constructed universal graph and subsequently identified many structural equality constraints by exploiting the proposed method in Chapter 4. A comparison study between the performance of this updated model with that of the original parameter-free model in terms of their ability to correctly identify non-Hamiltonian cubic graphs up to size 18 revealed promising results. While the original parameter-free model was successful at identifying approximately 18 % of non-Hamiltonian non-bridge graphs, it failed on the remaining 82% of the latter. However the new model achieved 100% success rate with identifying all tested non-Hamiltonian instances.

There are several naturally arising topics for future research in this direction. They include the following:

1. A naturally arising topic for future research is to exploit the proposed method in Chapter 4 to extract effective inequality constraints for IP problems. Notably, one can use the ECA-method to generate equality constraints for IP problems. Then, one might find good inequality constraints by projecting the feasible space of those IP formulations into the original, lower dimensional, variable space.

2. As mentioned in Remark 5.1.4, a sparser universal graph leads to a tighter approximation for the convex hull of HCs of a given cubic graph. Note that designing sparse universal graphs for regular graphs is extensively studied in [1] and references within. However, designing a universal graph with the properties that are discussed in Section 5.2 is a new problem. For example, following graph is a universal graph with our desired properties.

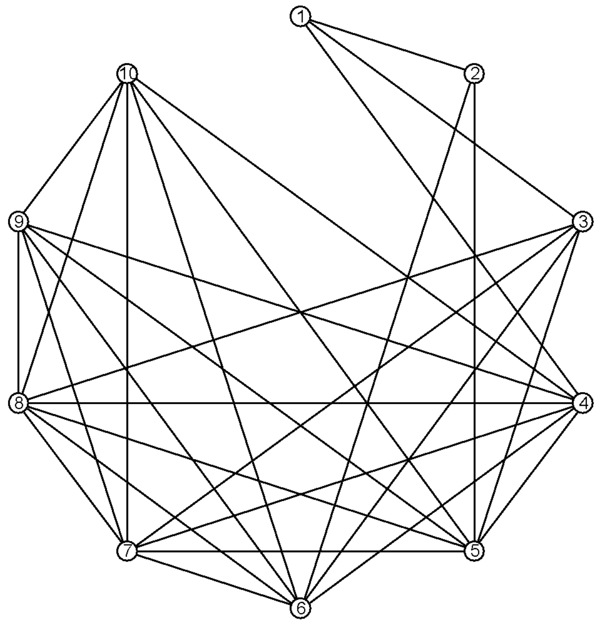

Figure 6.1: A sparse universal graph

In this graph, we removed $4N - 4$ edges from the complete graph, while in the universal graph U_N that was used in Section 5.2, we only removed $N - 4$ edges from the complete graph. This indicates the further advantages that can still be made, and hence designing a sparser universal graph with the desired properties is an interesting problem for future work, which seems likely to lead to better approximations.

3. One practical issue with the parameter-free model is that it possesses many variables, and so despite being polynomially large, it is still very inefficient. Ideally, we would like to design an equivalently tight model with only $\mathcal{O}(N^2)$ variables. However, early experiments along this line have indicated that using only $\mathcal{O}(N^2)$ variables decreases the accuracy of detecting non-hamiltonicity. One potential idea for overcoming this issue is to construct a family of graphs which are, in some sense, universal for all cubic graphs (see Section 6.1). Based on this family of universal graphs, we could establish stronger structural equality constraints for TSP and HCP on cubic graphs using only $\mathcal{O}(N^2)$ variables.

4. As discussed in item 2 above, we can improve the accuracy of detecting non-Hamiltonicity through having a model with more variables (increasing the number of indices) or considering more vertices at a time (see Section 6.1). It seems increasing the indices is more expensive than increasing the number of vertices considered at a time. Therefore, it seems preferable to use the latter alternative to improve the accuracy. However, a proper comparison study should be carried out to determine which combination of alternatives is the most effective. For example, we could try a 3-index model and consider 3 vertices at a time, or alternatively a 2-index model, and consider 4 or 5 vertices at a time, and see which model is the most accurate and efficient.

6.1 CSEC constraints based on 2-index variables

In this section[1], we demonstrate the idea of maximising the benefit of a model with $\mathcal{O}(N^2)$ variables. In particular, we may be interested in having a model with only 2-index variables, namely x_{ij}, for cubic graphs. In the directed case, the variable $x_{ij} = 1$ if the HC goes from vertex i to vertex j and otherwise $x_{ij} = 0$.

Considering K_N, the only equality constraints based on the 2-index variables are assignment constraints which are listed below

$$\sum_{\substack{i=1 \\ i \neq j}}^{N} x_{ij} = 1, \quad j = 1, \ldots, N$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{N} x_{ij} = 1, \quad i = 1, \ldots, N.$$

It is easy to check that the dimension of $\mathbf{conv}(H(G))$ is much smaller than the dimension of $\mathbf{conv}(K_N)$ for any given cubic graph G. Therefore, there exist a number of structural equality constraints in 2-index variables model.

[1]The results of this section is under review in a journal for publication [32].

To extract a set of structural equality constraints for 2-index variable models, a very natural thought is to exploit the analysis that discussed in Section 4.3. However, that process will not lead to any structural equality constraints, because unlike for the parameter-free model (with 4-index variables), the dimension of $\mathbf{conv}(H(G))$ is equal to the dimension of $\mathbf{conv}(K_N)$ in the 2-index variable model. We will extend the idea of using one universal graph, to using several subgraphs in order to extract new equality constraints for the 2-index case.

We consider three arbitrary vertices of complete graph K_N and remove $N-4$ edges conecting to each of these three vertice as in Section 5.2. This construction results in a graph which has three vertices with degree 3. Enumerating all such graphs and removing those which are isomorphic, we found that there exist 40 graphs with that properties, for any $N \geq 12$. We call these 40 graphs *quasi-universal* graphs and denote them by $U^{(1)}, \ldots, U^{(40)}$. The name quasi-universal graph is inspired from the fact that each $U^{(i)}$ graph is not universal for all cubic graphs, but we can find a copy of any given cubic graph G in at least one of the quasi-universal graphs. We also consider subgraphs of $U^{(i)}$ graphs that are constructed by just taking into account three degree 3 vertices and their adjacent edges and vertices. We call them kernels and denote them by $\mathcal{G}_1, \ldots, \mathcal{G}_{40}$.

Comparing the dimension of $\mathbf{conv}\left(H(U^{(i)})\right)$, $i = 1, \ldots, 40$ to the dimension of $\mathbf{conv}\,(K_N)$ shows that there are only thirteen interesting quasi-universal graphs, namely $U^{(1)}, \ldots, U^{(13)}$ and the other quasi-universal graphs will result in no new equality constraints. The kernels corresponding to those thirteen quasi-universal graphs, along with their associated structural equality constraints obtained from the method in Section 4.3, are demonstrated below.

Remark 6.1.1.

1. The appended structural equality constraints have been identified via the method in Section 4.3, but have not yet been formally proved. In principle, such proofs should be straightforward (but extensive) by the enumeration of all possible

cases.

2. The constraints applicable for each kernel are valid for any graph that contains that kernel as an induced subgraph. However, if only cubic graphs are considered, some constraints will be redundant, for example, those arising from kernel $\mathcal{G}_{12}$.

Kernel $\mathcal{G}_1$:

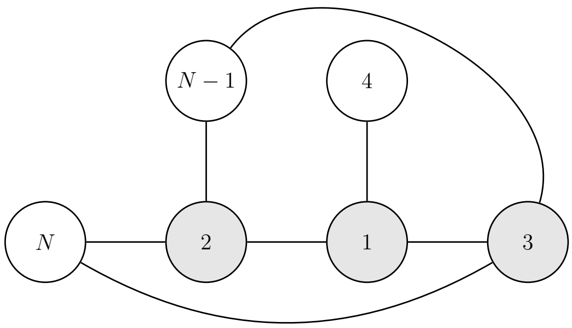

Figure 6.2: Demonstration of $\mathcal{G}_1$

Associated 2-index structural equality constraints:

$$\sum_{i=2}^{N-2} x_{i(N-1)} = 1,$$
$$\sum_{j=4}^{N-2} \sum_{\substack{i=4 \\ i \neq j}}^{N-2} x_{ij} = N-6.$$

Kernel $\mathcal{G}_2$:

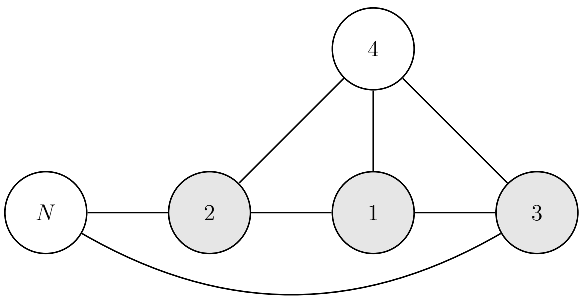

Figure 6.3: Demonstration of $\mathcal{G}_2$

Associated 2-index structural equality constraints:

$$x_{12} + x_{13} = 1,$$

$$x_{12} + x_{21} = 1,$$

$$x_{31} = x_{12},$$

$$x_{12} + x_{34} + x_{42} = 1,$$

$$x_{24} + x_{43} = x_{12},$$

$$\sum_{i=5}^{N-1} x_{4i} = x_{24} + x_{34},$$

$$\sum_{i=2}^{N-1} x_{i4} = 1.$$

Kernel $\mathcal{G}_3$:

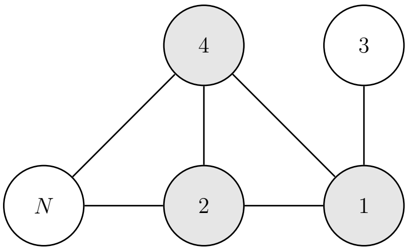

Figure 6.4: Demonstration of $\mathcal{G}_3$

Associated 2-index structural equality constraints:

$$x_{13} + x_{31} = 1,$$

$$x_{21} + x_{24} = x_{12} + x_{13},$$

$$x_{12} + x_{13} + x_{42} - x_{21} = 1,$$

$$\sum_{i=5}^{N-1} x_{3i} = x_{13},$$

$$x_{13} + \sum_{i=5}^{N-1} x_{i3} = 1.$$

Kernel $\mathcal{G}_4$:

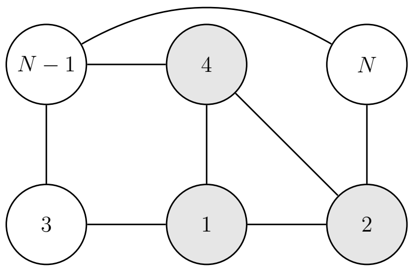

Figure 6.5: Demonstration of $\mathcal{G}_4$

Associated 2-index structural equality constraints:

$$x_{24} + x_{42} = x_{13} + x_{31}.$$

Kernel $\mathcal{G}_5$:

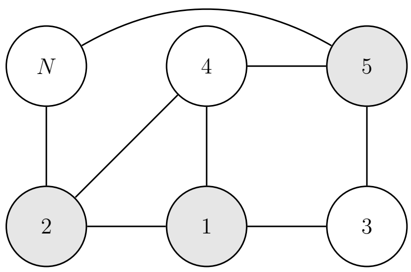

Figure 6.6: Demonstration of $\mathcal{G}_5$

Associated 2-index structural equality constraints:

$$x_{34} = 0,$$

$$x_{43} = 0,$$

$$x_{12} + x_{21} = 1,$$

$$x_{31} + \sum_{i=5}^{N-1} x_{3i} = 1,$$

$$x_{13} + \sum_{i=5}^{N-1} x_{i3} = 1,$$

$$x_{12} + x_{42} + \sum_{i=5}^{N-1} x_{4i} - x_{31} = 1,$$

$$x_{24} + \sum_{i=5}^{N-1} x_{i4} = x_{12} + x_{13}.$$

Kernel $\mathcal{G}_6$:

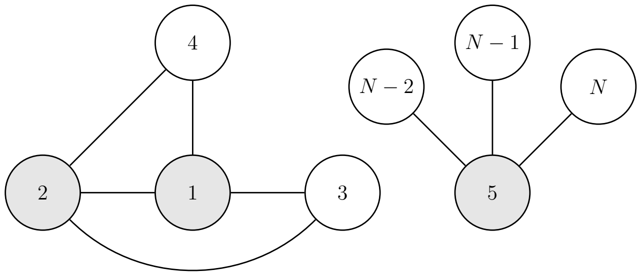

Figure 6.7: Demonstration of $\mathcal{G}_6$

Associated 2-index structural equality constraints:

$$x_{34} = 0,$$

$$x_{43} = 0,$$

$$x_{12} + x_{21} = 1,$$

$$x_{23} + x_{31} = x_{12},$$

$$x_{12} + x_{13} + x_{32} = 1.$$

Kernel $\mathcal{G}_7$:

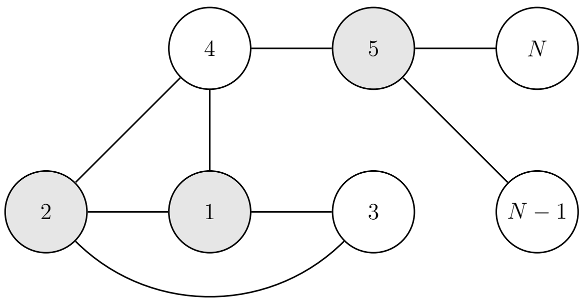

Figure 6.8: Demonstration of $\mathcal{G}_7$

Associated 2-index structural equality constraints:

$$x_{34} = 0,$$

$$x_{43} = 0,$$

$$x_{12} + x_{21} = 1,$$

$$x_{23} + x_{31} = x_{12},$$

$$x_{12} + x_{13} + x_{32} = 1.$$

Kernel $\mathcal{G}_8$:

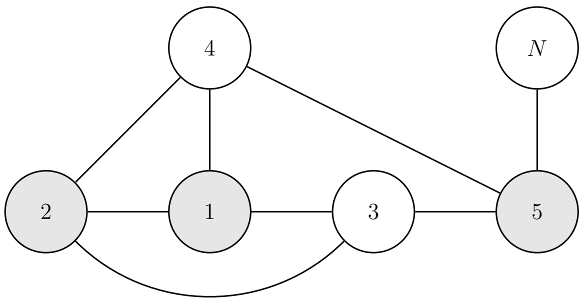

Figure 6.9: Demonstration of $\mathcal{G}_8$

Associated 2-index structural equality constraints:

$$x_{34} = 0,$$

$$x_{43} = 0,$$

$$x_{12} + x_{21} = 1,$$

$$x_{23} + x_{31} = x_{12},$$

$$x_{12} + x_{13} + x_{32} = 1,$$

$$x_{13} + x_{23} + x_{45} + x_{53} = 1,$$

$$x_{35} + x_{54} = x_{13} + x_{23},$$

$$\sum_{i=6}^{N-1} x_{i3} = x_{45},$$

$$\sum_{i=6}^{N-1} x_{i4} = x_{35},$$

$$x_{13} + x_{23} + \sum_{i=5}^{N-1} x_{4i} = 1,$$
$$\sum_{i=5}^{N-1} x_{3i} = x_{13} + x_{23}.$$

Kernel $\mathcal{G}_9$:

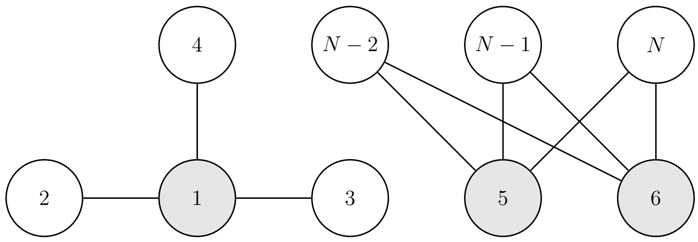

Figure 6.10: Demonstration of $\mathcal{G}_9$

Associated 2-index structural equality constraints:

$$x_{N-2,N-1} = 0,$$
$$\sum_{i=2}^{N-3} x_{i(N-2)} = 1,$$
$$\sum_{i=2}^{N-3} x_{i(N-1)} = 1,$$
$$\sum_{i=2}^{N-3} x_{(N-2)i} = 1,$$
$$\sum_{i=2}^{N-3} x_{(N-1)i} = 1,$$
$$\sum_{i=2}^{4} \left(\sum_{\substack{j=2 \\ j \neq i}}^{4} x_{ij} + \sum_{j=7}^{N-3} x_{ij} \right) + \sum_{i=7}^{N-3} \left(\sum_{j=2}^{4} x_{ij} + \sum_{\substack{j=7 \\ j \neq i}}^{N-3} x_{ij} \right) = N - 8.$$

Kernel $\mathcal{G}_{10}$:

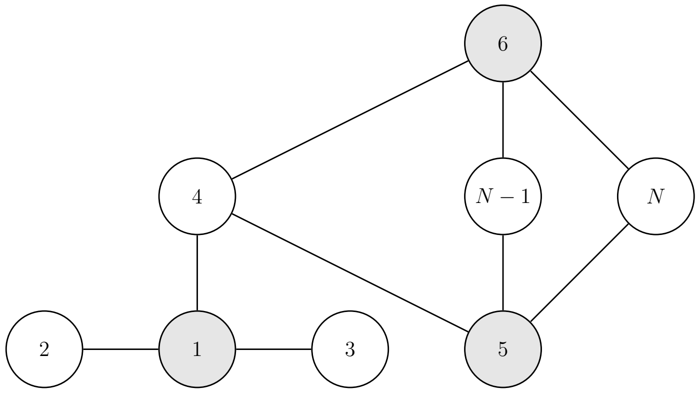

Figure 6.11: Demonstration of $\mathcal{G}_{10}$

Associated 2-index structural equality constraints:

$$
\begin{aligned}
&x_{4(N-1)} = 0,\\
&x_{(N-1)4} = 0,\\
&\sum_{\substack{i=2\\ i\neq 4}}^{N-2} x_{i(N-1)} = 1,\\
&\sum_{\substack{i=2\\ i\neq 4}}^{N-2} x_{4i} = x_{21} + x_{31},\\
&\sum_{\substack{i=2\\ i\neq 4}}^{N-2} x_{i4} = x_{12} + x_{13},\\
&\sum_{i=2}^{3}\left(\sum_{\substack{j=1\\ j\neq i}}^{3} x_{ij} + \sum_{j=7}^{N-2} x_{ij}\right) + \sum_{i=7}^{N-2}\left(\sum_{j=2}^{3} x_{ij} + \sum_{\substack{j=7\\ j\neq i}}^{N-2} x_{ij}\right) = N-6.
\end{aligned}
$$

Kernel $\mathcal{G}_{11}$:

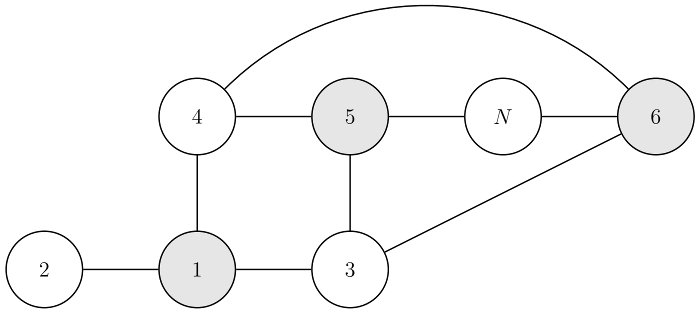

Figure 6.12: Demonstration of $\mathcal{G}_{11}$

Associated 2-index structural equality constraints:

$$
\begin{aligned}
&x_{23} = 0,\\
&x_{24} = 0,\\
&x_{32} = 0,\\
&x_{34} = 0,\\
&x_{42} = 0,\\
&x_{43} = 0,\\
&x_{12} + x_{21} = 1,\\
&\sum_{i=7}^{N-1} x_{2i} = x_{12},\\
&x_{12} + \sum_{i=7}^{N-1} x_{i2} = 1,\\
&x_{31} + \sum_{i=5}^{N-1} x_{3i} = 1,\\
&x_{13} + \sum_{i=5}^{N-1} x_{i3} = 1,\\
&x_{12} + \sum_{i=5}^{N-1} x_{4i} - x_{31} = 1,\\
&\sum_{i=5}^{N-1} x_{i4} = x_{12} + x_{13}.
\end{aligned}
$$

Kernel $\mathcal{G}_{12}$:

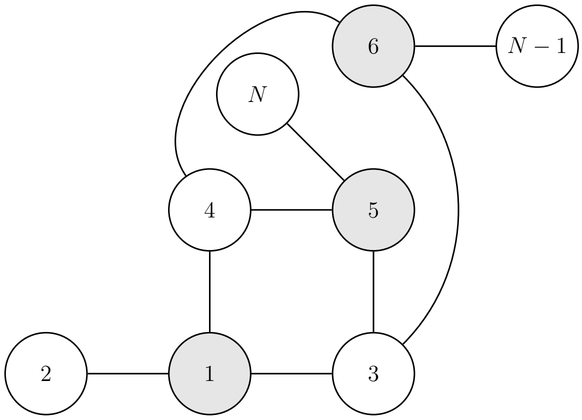

Figure 6.13: Demonstration of $\mathcal{G}_{12}$

Associated 2-index structural equality constraints:

$$x_{34} = 0,$$

$$x_{43} = 0.$$

Kernel $\mathcal{G}_{13}$:

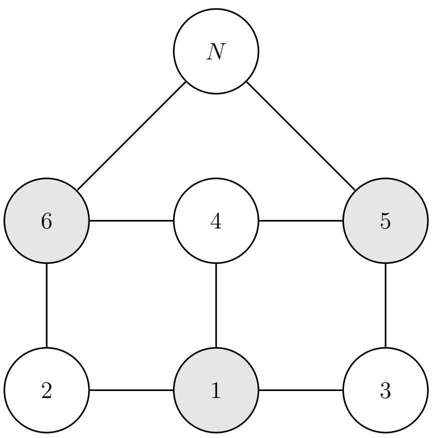

Figure 6.14: Demonstration of $\mathcal{G}_{13}$

Associated 2-index structural equality constraints:

$$
\begin{aligned}
&x_{23} = 0,\\
&x_{24} = 0,\\
&x_{32} = 0,\\
&x_{34} = 0,\\
&x_{42} = 0,\\
&x_{43} = 0,\\
&x_{21} + \sum_{i=6}^{N-1} x_{2i} = 1,\\
&x_{31} + x_{35} + \sum_{i=7}^{N-1} x_{3i} = 1,\\
&x_{12} + \sum_{i=6}^{N-1} x_{i2} = 1,\\
&x_{13} + x_{53} + \sum_{i=7}^{N-1} x_{i3} = 1,\\
&\sum_{i=5}^{N-1} x_{4i} = x_{21} + x_{31},\\
&\sum_{i=5}^{N-1} x_{i4} = x_{12} + x_{13}.
\end{aligned}
$$

To exploit the above quasi-universal graphs, we consider all combinations of three vertices of a given cubic graph G, one combination at a time. Suppose we consider vertices i, j, and k when $i < j < k$ and construct a subgraph of graph G, namely G_{ijk}, which contains vertices i, j, k and their adjacent vertices and edges. We then check if G_{ijk} is isomorphic with any of those thirteen interesting kernels. For example, suppose G_{ijk} is isomorphic with $\mathcal{G}_1$ under map $\mathcal{M}_1$, that is $G_{ijk} = \mathcal{M}_1(\mathcal{G}_1)$. Therefore, we can relabel the variables in the constraints for $\mathcal{G}_1$ based on map $\mathcal{M}_1$ and add them to the set of constraints for that graph.

To illustrate the above discussion, consider the following 12-vertex cubic graph.

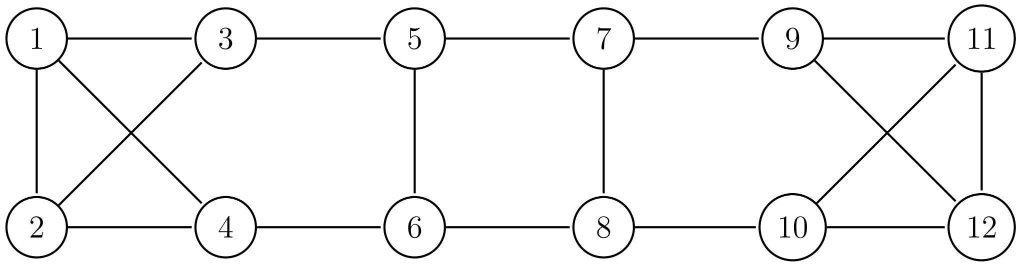

Figure 6.15: A 12-vertex cubic graph

Then G_{123} is the following subgraph:

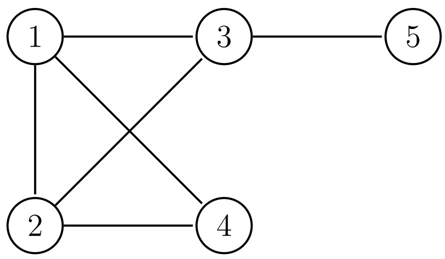

Figure 6.16: Subgraph G_{123}

It is clear, by construction in the case $N = 12$, that each G_{ijk} can be isomorphic to some $\mathcal{G}_i$ for $i = 1, \ldots, 13$. For instance G_{123} is isomorphic with $\mathcal{G}_3$ via the isomorphism $\mathcal{M}_3 : \{1, 2, 3, 4, 12\} \to \{3, 2, 5, 1, 4\}$.

We then use $\mathcal{M}_3$ to construct a map, namely $\mathcal{M}_3^U$, such that $G \subset \mathcal{M}_3^U(U^{(3)})$. To achieve this, we can consider $\mathcal{M}_3^U$ as the same map as $\mathcal{M}_3$ for those vertices which belong to G_{123} and for the remaining vertices, we just choose an arbitrary map. For instance we could choose

$$\mathcal{M}_3^U : \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\} \to \{3, 2, 5, 1, 6, 7, 8, 9, 10, 11, 12, 4\},$$

Obviously, exploiting Corollary 5.1.2, we see $\mathbf{conv}(H(G)) \subset \mathbf{conv}(H(\mathcal{M}_3^U(U^{(3)})))$. Therefore, following constraints hold for this 12-vertex graph:

$$x_{\mathcal{M}_3^U(1)\mathcal{M}_3^U(3)} + x_{\mathcal{M}_3^U(3)\mathcal{M}_3^U(1)} = 1,$$

$$x_{\mathcal{M}_3^U(2)\mathcal{M}_3^U(1)} + x_{\mathcal{M}_3^U(2)\mathcal{M}_3^U(4)} = x_{\mathcal{M}_3^U(1)\mathcal{M}_3^U(2)} + x_{\mathcal{M}_3^U(1)\mathcal{M}_3^U(3)},$$

$$x_{\mathcal{M}_3^U(1)\mathcal{M}_3^U(2)} + x_{\mathcal{M}_3^U(1)\mathcal{M}_3^U(3)} + x_{\mathcal{M}_3^U(4)\mathcal{M}_3^U(2)} - x_{\mathcal{M}_3^U(2)\mathcal{M}_3^U(1)} = 1,$$

$$\sum_{i=6}^{N-1} x_{\mathcal{M}_3^U(3)\mathcal{M}_3^U(i)} = x_{\mathcal{M}_3^U(1)\mathcal{M}_3^U(3)},$$

$$x_{\mathcal{M}_3^U(1)\mathcal{M}_3^U(3)} + \sum_{i=5}^{N-1} x_{\mathcal{M}_3^U(i)\mathcal{M}_3^U(3)} = 1.$$

We repeat this procedure for all possible combinations of three vertices out of twelve. This analysis shows that 20 quasi universal graphs cover the 12-vertex graph isomorphically. This results are reported in Table 6.1.

Table 6.1: Isomorphisms for quasi-universal graphs

Chosen vertices	Kernel k-th	Map $\mathcal{M}_k$
1, 2, 3	3	$\{1,2,3,4,12\} \rightarrow \{3,2,5,1,4\}$
1, 2, 4	3	$\{1,2,3,4,12\} \rightarrow \{4,2,6,1,3\}$
1, 2, 5	7	$\{1,2,3,4,5,11,12\} \rightarrow \{1,2,4,3,5,6,7\}$
1, 2, 6	7	$\{1,2,3,4,5,11,12\} \rightarrow \{1,2,3,4,6,5,8\}$
1, 2, 7	6	$\{1,2,3,4,5,10,11,12\} \rightarrow \{1,2,3,4,7,5,8,9\}$
1, 2, 8	6	$\{1,2,3,4,5,10,11,12\} \rightarrow \{1,2,3,4,8,6,7,10\}$
1, 2, 9	6	$\{1,2,3,4,5,10,11,12\} \rightarrow \{1,2,3,4,9,7,11,12\}$
1, 2, 10	6	$\{1,2,3,4,5,10,11,12\} \rightarrow \{1,2,3,4,10,8,11,12\}$
1, 2, 11	6	$\{1,2,3,4,5,10,11,12\} \rightarrow \{1,2,3,4,11,10,9,12\}$
1, 2, 12	6	$\{1,2,3,4,5,10,11,12\} \rightarrow \{1,2,3,4,12,10,11,9\}$
1, 11, 12	6	$\{1,2,3,4,5,10,11,12\} \rightarrow \{12,11,10,9,1,2,3,4\}$
2, 11, 12	6	$\{1,2,3,4,5,10,11,12\} \rightarrow \{12,11,10,9,2,1,3,4\}$
3, 11, 12	6	$\{1,2,3,4,5,10,11,12\} \rightarrow \{12,11,10,9,3,1,2,5\}$
4, 11, 12	6	$\{1,2,3,4,5,10,11,12\} \rightarrow \{12,11,10,9,4,1,2,6\}$
5, 11, 12	6	$\{1,2,3,4,5,10,11,12\} \rightarrow \{12,11,10,9,5,3,6,7\}$
6, 11, 12	6	$\{1,2,3,4,5,10,11,12\} \rightarrow \{12,11,10,9,6,4,5,8\}$
7, 11, 12	7	$\{1,2,3,4,5,11,12\} \rightarrow \{11,12,10,9,7,8,5\}$
8, 11, 12	7	$\{1,2,3,4,5,11,12\} \rightarrow \{11,12,9,10,8,6,7\}$
9, 11, 12	3	$\{1,2,3,4,12\} \rightarrow \{9,11,7,12,10\}$
10, 11, 12,	3	$\{1,2,3,412\} \rightarrow \{10,12,8,119\}$

In the above table, the first column shows the combination of three vertices chosen, and the second column indicates a kernel which is isomorphic to the subgraph constructed by those three vertices. The third column provides an isomorphism which maps the kernel on that constructed subgraph. We utilise the maps in the third column as discussed in the mentioned example to construct constraints for the given

12-vertex graph. Note that the maps in the third column are not unique and one could gain more constraints through an analogous way to that discussed in Section 5.2.

These theoretical and numerical results along with the results of Chapter 5, naturally, lead to the following conjecture.

Conjecture 6.1.2. *For a given non-Hamiltonian cubic graph G, only a constant number of combinations of vertices need to be considered in order to obtain sufficiently many structural equality constraints to ensure that the constructed polytope is infeasible.*

References

[1] N. Alon and M. Capalbo, “Sparse universal graphs for bounded-degree graphs”, *Random Structures and Algorithms*, 31(2):123–133, 2007.

[2] D.L. Applegate, R.E. Bixby, V. Chvatal, and W.J. Cook, *The Traveling Salesman Problem: A Computational Study*, Princeton University Press, New Jersey, 2007.

[3] D. Avis, B. Bremner, and R. Seidel, “How good are convex hull algorithms?”, *Computational Geometry: Theory and Applications*, 7:265–302, 1997.

[4] D. Avis and K. Fukuda,“A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra”, *Discrete & Computational Geometry*, 8(3):295–313, 1992.

[5] D. Avis and K. Fukuda, “Reverse search for enumeration”, *Discrete Applied Mathematics*, 65:21–46, 1996.

[6] K. Avrachenkov, A. Eshragh, and J.A. Filar, “On transition matrices of Markov chains corresponding to Hamiltonian cycles”, *Annals of Operations Research* doi:10.1007/s10479–014–1642–2, 2014.

[7] B. Balas, “Projection, lifting and extended formulation in integer and combinatorial optimization”, *Annals of Operations Research*, 140:125–161, 2005.

[8] V.S. Borkar, V. Ejov, and J.A. Filar, “Directed Graphs, Hamiltonicity and Doubly Stochastic Matrices”, *Random Structures and Algorithms*, 25(4):376–395, 2004.

[9] V.S. Borkar, V. Ejov, and J.A. Filar, "On the Hamiltonicity Gap and Doubly Stochastic Matrices", *Random Structures & Algorithms*, 34(4):502–519, 2009.

[10] T. Christof and A. Loebel, "PORTA: Polyhedron Representation Transformation Algorithm", http://www.zib.de/Optimization/Software/Porta, 1997.

[11] M. Conforti, G. Cornuejols, and G. Zambelli, *Integer Programming*, Springer-Verlag, Berlin, 2014.

[12] M. Desrochers and G. Laporte, "Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints", *Operations Research Letters*, 10:27–36, 1991.

[13] V. Ejov, J.A. Filar, and J. Gondzio, "An Interior Point Heuristic for the Hamiltonian Cycle Problem via Markov Decision Processes", *Journal of Global Optimization*, 29:315–334, 2004.

[14] V. Ejov, J.A. Filar, M. Haythorpe, and G.T. Nguyen, "Refined MDP-Based Branch-and-Bound Algorithm for the Hamiltonian Cycles Problem", *Mathematics of Operations Research*, 34(3):758–768, 2009.

[15] V. Ejov, J.A. Filar, W. Murray, and G.T. Nguyen,"Determinants and Longest Cycles of Graphs", *SIAM Journal of Discrete Mathematics*, 22(3):1215–1225, 2008.

[16] A. Eshragh and J.A. Filar, ""Hamiltonian Cycles, Random Walks and Discounted Occupational Measures", *Mathematics of Operations Research*, 36(2):258–270, 2011.

[17] E.A. Feinberg, "Constrained Discounted Markov Decision Processes and Hamiltonian Cycles", *Mathematics of Operations Research*, 25(1):130–140, 2000.

[18] J.A. Filar, "Controlled Markov Chains, Graphs & Hamiltonicity", *Foundation and Trends® in Stochastic Systems*, 1(2):77–162, 2006.

[19] J.A. Filar, M. Haythorpe, and S. Rossomakhine, 'A new heuristic for detecting non-Hamiltonicity in cubic graphs", *Computers & Operations Research*, 64:283–292, 2015.

[20] J.A. Filar and D. Krass, "Hamiltonian Cycles and Markov Chains", *Mathematics of Operations Research*, 19(1):223–237, 1994.

[21] J.A. Filar and A. Moeini, "Hamiltonian cycle curves in the space of discounted occupational measures", *Annals of Operations Research* doi:10.1007/s10479–015–2030–2, 2015.

[22] J.A. Filar and K. Vrieze, *Competitive Markov Decision Processes*, 1st Edition, Springer-Verlag, New York, 1996.

[23] K. Fukuda, T.M. Liebling, and F. Margot, "Analysis of backtrack algorithms for listing all vertices and all faces of a convex polyhedron", *Computational Geometry - Theory and Applications*, 8:1–12, 1997.

[24] M.R. Garey, D.S. Johnson and R. Tarjan, "The planar Hamiltonian circuit problem is NP-complete", *SIAM Journal on Computing*, 5:704–714, 1976.

[25] E. Gawrilow and M. Joswig, "Polymake: a framework for analyzing convex polytopes", In: G. Kalai et al (eds.) "Polytopes Combinatorics and Computation", DMV Seminar 29:43–73. Springer Basel AG, 2000.

[26] L. Gouveia, L. Simonetti, and E. Uchoa, "Modeling hop-constrained and diameter-constrained minimum spanning tree problems as Steiner tree problems over layered graphs", *Mathematical Programming*, 128:123–148, 2011.

[27] L.C.M. Kallenberg, *Linear Programming and Finite Markovian Control Problems*, Mathematical Centre Tracts 148, Amsterdam, 1983.

[28] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, Springer-Verlag, Berlin, 2000.

[29] N. Litvak and V. Ejov, "Markov Chains and Optimality of the Hamiltonian Cycle", *Mathematics of Operations Research*, 34(1):71–82, 2009.

[30] A. Moeini, "Identification of new Equality Constraints for Integer Programming Problems", *European Journal of operational research*, (Under review)

[31] A. Moeini and J.A. Filar, "Approximations of the convex hall of Hamiltonian cycles for cubic graphs", *Mathematical programming* , (Under review)

[32] A. Moeini and J.A. Filar, "Structural Equality Constraints for TSP on cubic graphs", *Mathematical Programming Computation*, (Under review)

[33] M.L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st Edition, Wiley-Interscience, New Jersey, 2005.

[34] G. Reinelt, TSPLIB. http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/

[35] H.D. Sherali and P.J. Driscoll, "On tightening the relaxations of Miller-Tucker-Zemlin formulations for asymmetric traveling salesman problems", *Operations Research*, 50(4):656–669, 2002.

[36] W.T. Tutte, "On Hamiltonian circuits", *Journal of the London Mathematics Society*, 21:98–101, 1946.

[37] H. Weyl, "Elementare Theorie der konvexen Polyeder. *Commentarii Mathematici Helvetici*, 7:290-306, 1935. Translation in: H. Kuhn, A. Tucker, *Contributions to the Theory of Games I*, Princeton University Press, Princeton, New Jersey, pp. 3–18, 1950.

[38] L. Wolsey, "Strong formulations for mixed integer programming: a survey", *Mathematical Programming, Series B*, 45:173–191, 1989.

[39] G.M. Ziegler, *Lectures on Polytopes*, Springer-Verlag, New York, 2007.