

An Interior Point Heuristic for the Hamiltonian Cycle Problem via Markov Decision Processes*

Jerzy Filar [†]

Jacek Gondzio [‡]

Vladimir Ejov [§]

April 3, 2003

Abstract

We consider the Hamiltonian cycle problem embedded in a singularly perturbed Markov decision process (MDP). More specifically, we consider the HCP as an optimization problem over the space of long-run state-action frequencies induced by the MDP's stationary policies. We show that Hamiltonian cycles (if any) correspond to the global minima of a suitably constructed indefinite quadratic programming problem over the frequency space. We show that the above indefinite quadratic can be approximated by quadratic functions that are “nearly convex” and as such suitable for the application of logarithmic barrier methods. We develop an interior-point type algorithm that involves an arc elimination heuristic that appears to perform rather well in moderate size graphs. The approach has the potential for further improvements.

Key words: Hamiltonian cycles, Markov Decision Processes, Interior Point Methods, Non-convex Optimization

1 Introduction

This paper is a continuation of a line of research [11],[4], [12], [7], [10] which aims to exploit the tools of controlled Markov decision chains (MDP's)¹ to study the properties of a famous problem of combinatorial optimization: the Hamiltonian Cycle Problem (HCP). More specifically, the present paper provides evidence that computationally effective algorithms for determining Hamiltonicity can be developed based on this approach. As such it can also be viewed as a continuation of the numerical experiments begun in Andramanov et al [4].

In this paper, we consider the following version of the Hamiltonian cycle problem: *given a directed graph, find a simple cycle that contains all vertices of the graph (Hamiltonian cycle (HC)) or prove that HC does not exist*. With respect to this property - *Hamiltonicity* - graphs possessing HC are called Hamiltonian. Next we shall, briefly, differentiate between our approaches and some of the best known “classical” approaches to the HCP.

*This research was supported in part by the grants from the Australian Research Council no. A00000767 and DP0343028

[†]School of Mathematics, The University of South Australia, Mawson Lakes, SA 5095, Australia; e-mail: jerzy.filar@unisa.edu.au

[‡]School of Mathematics, University of Edinburgh, e-mail: J.Gondzio@ed.ac.uk

[§]School of Mathematics, The University of South Australia, Mawson Lakes, SA 5095, Australia; e-mail: vladimir.ejov@unisa.edu.au

¹The acronym MDP stems from the alternative name of Markov decision processes.

Many of the successful classical approaches of discrete optimisation focus on solving a linear programming “relaxation” followed by heuristics that prevent the formation of sub-cycles. In our approach, we embed a given graph in a singularly perturbed MDP in such a way that we can identify Hamiltonian cycles and sub-cycles with exhaustive and non-exhaustive ergodic classes of induced Markov chains.

More precisely, our dynamic, stochastic approach to the HCP, considers a moving object tracing out a directed path on the graph G with its movement “controlled” by a function f (a *policy*) mapping the set of nodes $\mathcal{V} = \mathcal{V}(G) = \{0, 1, \dots, N\}$ of G into the set of arcs $\mathcal{A} = \mathcal{A}(G)$ of G . We think of this set of nodes as the *state space* of a controlled Markov chain $\Sigma = \Sigma(G)$ where for each state/node i , the *action space* $\mathcal{A}(i) := \{a | (i, a) \in \mathcal{A}\}$ is in one-to-one correspondence with the set of arcs emanating from that node, or, equivalently, with the set of endpoints of those arcs.

Illustration: Consider the complete graph G_5 on five nodes (with no self-loops) and think of the nodes as the states of an MDP, denoted by Σ , and of the arcs emanating from a given node as of actions available at that state. In a natural way the Hamiltonian cycle $c_1 : 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0$ corresponds to the “deterministic policy” $f_1 : \{0, 1, 2, 3, 4\} \rightarrow \{1, 2, 3, 4, 0\}$, where $f_1(2) = 3$ corresponds to the controller choosing arc (2,3) in state 2 with probability 1. The Markov chain induced by f_1 is given by the “zero-one” transition matrix $P(f_1)$ which, clearly, is irreducible. On the other hand, the union of two sub-cycles: $0 \rightarrow 1 \rightarrow 2 \rightarrow 0$ and $3 \rightarrow 4 \rightarrow 3$ corresponds to the policy $f_2 : \{0, 1, 2, 3, 4\} \rightarrow \{1, 2, 0, 4, 3\}$ which identifies the Markov chain transition matrix $P(f_2)$ (see below) containing two distinct ergodic classes. This leads to a natural embedding of the Hamiltonian cycle problem in a Markov control problem Σ . The latter MDP has a multi-chain ergodic structure which considerably complicates the analysis. However, this multi-chain structure can be “disguised” - but not completely lost - with the help of a “singular perturbation”. For instance, we could easily replace $P(f_2)$ with $P_\varepsilon(f_2)$:

$$P(f_2) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad P_\varepsilon(f_2) = \begin{pmatrix} 0 & 1 - 3\varepsilon^2 & \varepsilon^2 & \varepsilon^2 & \varepsilon^2 \\ \varepsilon & 0 & 1 - \varepsilon & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 1 - \varepsilon \\ \varepsilon & 0 & 0 & 1 - \varepsilon & 0 \end{pmatrix}.$$

The above perturbation is singular because it altered the ergodic structure of $P(f_2)$ by changing it to an irreducible (indeed, completely ergodic) Markov Chain $P_\varepsilon(f_2)$.

It is now convenient to work in the so-called *long-run frequency space*, X_ε , of state action frequencies. The effect of our perturbation is such that the long-run frequency of visits to state/node 0 – for a policy that traces out a Hamiltonian Cycle – can be explicitly calculated as $x_0 = 1/d_{1+N}(\varepsilon)$ (see Proposition 1) and introduced as an additional constraint on the frequency space. In [7] it was shown that Hamiltonian cycles are precisely the global minimizers of a suitably constructed quadratic program:

$$\begin{aligned} \min \mathbf{x}^T Q \mathbf{x} \\ \mathbf{x} \in X_\varepsilon \ \& \ x_0 = 1/d_{1+N}(\varepsilon). \end{aligned}$$

Furthermore, at the minimum $\mathbf{x}^T Q \mathbf{x} = 0$ which is equivalent to the policy corresponding to such a minimizer belonging to the set of deterministic policies. Note that Q is indefinite, but possesses a lot of special structure.

The fundamental idea of this paper is to consider a “more convex” objective function than $\mathbf{x}^T Q \mathbf{x}$. Such a function h_α can be constructed for small $\alpha > 0$ as $S(\mathbf{x}) - \alpha s(\mathbf{x})$, where $S(\mathbf{x}) = \sum_i (\sum_a x_{ia})^2$, $s(\mathbf{x}) = \sum_i \sum_a x_{ia}^2$, and x_{ia} is the frequency of choosing arc a when leaving node i . The non-convexity occurs from $\alpha s(\mathbf{x})$ that can now be controlled with the small factor α . We prove that the global minimum of h_α occurs if not at HC itself then in its small neighbourhood sufficient to recover the HC . In view of this it is reasonable to hope that an interior point algorithm that works well for convex quadratic programs may also work “well-enough” for the above surrogate problem, when α is sufficiently small.

We develop a heuristic, within an interior point method, that searches for a local minimum of h_α instead of the global one and decides that a particular arc is used at the prospective HC if it is “dominant” among all arcs from the same node at the local minimum point. The dominant arcs are then declared to be part of a prospective HC and other arcs from the corresponding nodes are eliminated. This simplifies the structure of the original graph.

It is often the case that even a single dominant arc causes a cascade of further reductions. However, it may happen that there are no dominant arcs in the original setting. Then we need to use some branching algorithm that chooses exactly one arc from a particular node and then applies the dominant arc search.

We report a series of numerical experiments with this heuristic. We found that it seems to perform very well on small to moderate size graphs. For instance, on randomly generated graphs with some 100 nodes and 300 arcs, the heuristic found Hamiltonian cycles in approximately 107 seconds. For structured graphs such as the “Knight’s tour” problem we succeeded in solving an instance of the 32×32 chessboard, albeit in 9 hours and 57 minutes. We also report the performance of our interior point heuristic on cubic “generalised Petersen graphs” [1] that have generated interest among graph theorists. We found that instances of these graphs with approximately 500 arcs were solved in about 1 minute. Finally, in our experimentation, we found only one instance of a graph where our heuristic failed to find a Hamiltonian cycle. This was a 94 node graph with 282 arcs. Of course, the heuristic could be replaced by a complete algorithm that was guaranteed to produce a definitive answer but at the cost of returning to the previously eliminated “non-dominant” arcs. The subject of an efficient implementation of the latter is part of the continuing research.

2 A formulation of HCP by means of a perturbed MDP

The Hamiltonian Cycle Problem (HCP) is the problem to find a simple cycle of $1 + N$ arcs, that is a *Hamiltonian Cycle* or a *tour*, in a directed graph G with $1 + N$ nodes numbered from 0 to N and with arcs m arcs (i, a) , or to determine that none exist. Recall that a simple cycle is one that passes exactly once through each node comprising the cycle. We will identify G with its *adjacency matrix*

$$G[i, a] = \begin{cases} 1 & : G \text{ contains arc } (i, a) \\ 0 & : \text{otherwise} \end{cases}$$

We employ the following perspective of the HC problem, introduced in [14]: Consider a moving object tracing out a directed path on the graph G with its movement “controlled” by a function f mapping the set of nodes $\mathcal{V} = \mathcal{V}(G) = \{0, 1, \dots, N\}$ of G into the set of arcs $\mathcal{A} = \mathcal{A}(G)$ of G . We think of this set of nodes as the state space of a Markov decision process $\Gamma = \Gamma(G)$ where for each state/node s , the action space

$$\mathcal{A}(i) = \{a | (i, a) \in \mathcal{A}\}$$

is in one-to-one correspondence with the set of arcs emanating from that node, or, equivalently, with the set of endpoints of those arcs. In order to ensure that our MDP is *irreducible* we now introduce the perturbed transition probabilities for Γ to create an ε -perturbed process Γ_ε (for $0 < \varepsilon < 1$) defined by

$$p_\varepsilon[j; i, a] = \begin{cases} 1 - (N-1)\varepsilon^2 & : \text{ if } i = 1 \text{ and } a = j \\ \varepsilon^2 & : \text{ if } i = 1 \text{ and } a \neq j \\ 1 & : \text{ if } i > 1 \text{ and } a = j = 1 \\ \varepsilon & : \text{ if } i > 1 \text{ and } a \neq j = 1 \\ 1 - \varepsilon & : \text{ if } i > 1 \text{ and } a = j > 1 \\ 0 & : \text{ in all remaining cases} \end{cases}$$

Here $p_\varepsilon(j; i, a)$ represents the probability to move from the node i to the node j by choosing the action (i, a) , so

$$\sum_{j=0}^N p_\varepsilon[j; i, a] = 1 \quad \text{for all } (i, a) \in \mathcal{A}. \quad (1)$$

Note that with the above perturbation, for each pair of nodes i, a (not equal to 1) corresponding to a “deterministic arc” (i, a) , our perturbation replaces that arc by a pair of “stochastic arcs” $(i, 1)$ and (i, a) with weights ε and $(1 - \varepsilon)$ respectively. This stochastic perturbation has the following interpretation: a decision to move along arc (i, a) results in movement along (i, a) only with probability $(1 - \varepsilon)$ and with probability ε the process returns to the home node 1; a decision to move along arc $(1, a)$ results in movement along $(1, a)$ with probability $1 - N\varepsilon^2$ and with probabilities ε^2 along the remaining N arcs $(1, a')$ $a' \neq a$. The quadratic in ε part of the perturbation ensures that there will arise only *irreducible* Markov chains defined by *stationary strategies* through the above perturbation.

A *policy* is defined by $(1 + N) \times (1 + N)$ *stochastic* matrix \mathbf{f} (rows of \mathbf{f} add up to 1) with entries

$$f[i, a] = \begin{cases} \text{probability of action } a \text{ in state } i \text{ whenever } i \text{ is visited} & : a \in \mathcal{A}(i) \\ 0 & : a \notin \mathcal{A}(i) \end{cases}$$

Strategies compose the *strategy space* denoted by \mathbf{F}_S . A strategy is called deterministic if $f[i, a] \in \{0, 1\}$ for all i and a . That is, for each i the controller chooses some particular action $a \in \mathcal{A}(i)$ with probability 1 whenever i is visited. In this case we will also write $f(i) = a$. The space of deterministic strategies will be denoted by \mathbf{F}_D . The fact that controller’s decision does not depend on the time of a visit to the state i is reflected in the name *stationary policy* often used for a strategy. Any stationary policy \mathbf{f} gives rise to a $(1 + N) \times (1 + N)$ *probability transition* matrix $P(\mathbf{f})$ with entries

$$P_\varepsilon(\mathbf{f})[i, j] := p_\varepsilon(j; i, \mathbf{f}) := \sum_{a=0}^N p_\varepsilon[j; i, a] f[i, a].$$

Condition 1 implies that $P_\varepsilon(\mathbf{f})$ is also a stochastic matrix. In matrix notations the Markov chain $P_\varepsilon(\mathbf{f})$ can be written as a regular perturbation of \mathbf{f} given by

$$P_\varepsilon(\mathbf{f}) = \mathbf{f} + \varepsilon \Delta_{1,l} \mathbf{f} \Delta_{1,r} + \varepsilon^2 \Delta_{2,l} \mathbf{f} \Delta_{2,r}$$

where Δ_1 and Δ_2 are sparse matrices with the entries

$$\Delta_{1,l}[i, a] = \begin{cases} 1 & : a = i > 0 \\ 0 & : \text{otherwise} \end{cases},$$

$$\Delta_{1,r}[i, a] = \begin{cases} 1 & : a = 0 \\ -1 & : a = i > 0 \\ 0 & : \text{otherwise} \end{cases},$$

$$\Delta_{2,l}[i, a] = \begin{cases} 1 & : i = 1 \\ 0 & : \text{otherwise} \end{cases},$$

$$\Delta_{2,r}[i, a] = \begin{cases} 0 & : a = 0 \\ 1 - N & : a = i > 0 \\ 1 & : \text{otherwise} \end{cases},$$

and so P_ε may be considered as a non-degenerate operator on stochastic matrices.

The feature of an irreducible Markov chain P is the simple description of its *Cesaro-limit* matrix Q that has identical rows $\mathbf{q} = (q_0, \dots, q_N) > 0$ that represent the unique solution of the linear system of equations:

$$\begin{aligned} \mathbf{qP} &= \mathbf{q} \\ \mathbf{q1} &= \mathbf{1}, \end{aligned}$$

where $\mathbf{1}$ is an N -dimensional column vector with unity in every entry. The vector \mathbf{q} is called the *stationary distribution* of the irreducible Markov chain P .

Consider the irreducible Markov chain $P_\varepsilon(\mathbf{f})$ determined by a stationary policy \mathbf{f} . Let $\mathbf{q}(\mathbf{f})$ be its stationary distribution vector. For each $a \in \mathcal{A}(s)$, $i \in \mathcal{V}$ define the long-run *frequency of the state-action pair* $(i, a) \in \mathcal{A}(i)$ as

$$x_{ia}(\mathbf{f}) := q_s(\mathbf{f})f[i, a].$$

The long run *frequency of the state* i is defined as the aggregate

$$x_i(\mathbf{f}) := \sum_{a \in \mathcal{A}(i)} x_{ia}(\mathbf{f}) = q_s(\mathbf{f}),$$

where the last equality follows from the fact that $\sum_{a \in \mathcal{A}(i)} f[i, a] = 1$. Components $\{x_i\}$ form the *long-run state frequency* $1 + N$ row vector

$$\bar{\mathbf{x}}(\mathbf{f}) = (x_0(\mathbf{f}), x_1(\mathbf{f}), \dots, x_N(\mathbf{f})).$$

Furthermore, define the *long-run (state-action) frequency vector* $\mathbf{x}(\mathbf{f})$ induced by \mathbf{f} as the block-column vector whose i -th block is

$$\mathbf{x}_i(\mathbf{f}) = \left(x_{ia_1^i}(\mathbf{f}), x_{ia_2^i}(\mathbf{f}), \dots, x_{ia_{m_i}^i}(\mathbf{f}) \right)^T,$$

where $a_j^i \in \mathcal{A}(i)$ and m_i is the number of arcs in $\mathcal{A}(i)$. The construction of \mathbf{x} defines a map M of the strategy space \mathbf{F}_S into \mathbb{R}^m by

$$M(\mathbf{f}) := \mathbf{x}(\mathbf{f}).$$

The quadratic in ε part of the perturbation ensures that $x_i(\mathbf{f}) > 0$ for each i . Therefore, M is invertible and its inverse M^{-1} is defined by

$$M^{-1}(\mathbf{x})[i, a] = f_{\mathbf{x}}[i, a] := \frac{x_{ia}}{x_i}.$$

It is shown in [14] that the property that $\bar{\mathbf{x}}$ is the stationary distribution vector for $P_{\varepsilon}(\mathbf{f})$ is equivalent to the inclusion $\bar{\mathbf{x}} \in \mathbf{X}_{\varepsilon}$, where \mathbf{X}_{ε} is a polyhedron in \mathbb{R}^m , defined by the linear constraints

$$\begin{aligned} \text{(i)} \quad & \sum_{i=0}^N \sum_{a \in \mathcal{A}(i)} (\delta(i, j) - p_{\varepsilon}(j; i, a)) x_{ia} = 0; \quad j \in \mathcal{V} \\ \text{(ii)} \quad & \sum_{i=0}^N \sum_{a \in \mathcal{A}(i)} x_{ia} = 1 \\ \text{(iii)} \quad & x_{ia} \geq 0; \quad a \in \mathcal{A}(i), \quad i \in \mathcal{V},. \end{aligned}$$

with Kronecker δ coefficients $\delta(i, j)$.

In matrix notations \mathbf{X}_{ε} can be defined as

$$\mathbf{X}_{\varepsilon} = \{ \mathbf{x} | W_{\varepsilon} \mathbf{x} = \mathbf{0}, \mathbf{1}^T \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0} \},$$

where \mathbf{x} , $\mathbf{1}$ are both m -vectors ($\mathbf{1}$ has unity in every entry), and W_{ε} is an $(1 + N) \times m$ matrix with rows numbered by vertices \mathcal{V} and columns by arcs \mathcal{A} whose $(j, (i, a))$ th entry is

$$w_{\varepsilon}[j, (i, a)] := \delta(i, j) - p_{\varepsilon}(j; i, a).$$

Construction of \mathbf{X}_{ε} provides a connection between W_{ε} and the incidence $(1 + N) \times m$ matrix $D(G)$ of the graph defined as

$$D(G)[j, (i, a)] = \begin{cases} 1 & : i = j \\ -1 & : i = a \\ 0 & : \text{otherwise} \end{cases}$$

Lemma 1 *Matrix W_{ε} is a $(\varepsilon$ -quadratically) perturbed incidence matrix D .*

Proof:

By *augmentation* of several matrices Q_i with the same number of rows we mean “joining” them horizontally, hence, preserving the number of rows. This operation will be denoted as $\text{augment}_i Q_i$. Following the definition of W_{ε} we observe that

$$W_{\varepsilon} = \Delta - \text{augment}_i W_i,$$

where Δ is the “state-arc” analogue of the Kronecker δ :

$$\Delta[j, (i, a)] = \begin{cases} 1 & : j = i \\ 0 & : \text{otherwise} \end{cases}$$

and W_i is the $(1 + N) \times m_i$ -matrix with entries

$$W_i[j, a] = p_\varepsilon[j; i, a]$$

The definition of $p_\varepsilon[j; i, a]$ implies the claim of the lemma. \square

We note, that it is clear from the definition of $p_\varepsilon[j; i, a]$ that each column of W_i adds up to 1, so, each column of W_ε adds up to 0. Thus, $\text{rank}(W_\varepsilon) \leq N$. In fact, since $\text{rank}(D(G)) = N$ (by [6]), it follows from the lemma (1) that $\text{rank}(W_\varepsilon) = N$ for small ε .

We now recall [14] the partition of the space \mathbf{F}_D of deterministic strategies that is based on the subgraphs they “trace out” in G . In particular, note that with each $\mathbf{f} \in \mathbf{F}_D$ we associate a subgraph $G_{\mathbf{f}}$ of G defined by

$$\text{arc } (i, a) \in G_{\mathbf{f}} \iff f(i) = a.$$

We shall also denote a simple cycle of length $1 + k$ and beginning at 0 by a set of arcs

$$c_{1+k}^0 = \{(i_0 = 0, i_1), (i_1, i_2), \dots, (i_k, i_{1+k} = 0)\}; \quad k = 2, 3, \dots, N.$$

Thus, c_{1+N}^0 is a HC . If $G_{\mathbf{f}}$ contains a cycle c_{1+k}^0 we write $G_{\mathbf{f}} \supset c_{1+k}^1$. Let

$$C_{1+k}^0 := \{\mathbf{f} \in \mathbf{F}_D \mid G_{\mathbf{f}} \supset c_{1+k}^1\},$$

namely, the set of deterministic strategies that trace out a simple cycle of length $1 + k$, beginning at 0, for all $k = 2, 3, \dots, N$. Thus, $\bigcup_{k=1}^N C_{1+k}^0$ contains all strategies that start at 0 and the node where the strategy for the first time returns to is also node 0. Denote the complement to $\bigcup_{k=1}^N C_{1+k}^0$ in \mathbf{F}_D by B . Then B will contain strategies that start at the home node 0 and the node where the strategy for the first time returns to is different from node 0. The following proposition [9] can be proved along the same lines as the analogous result in [11]. It characterizes the partition

$$\mathbf{F}_D = \left[\bigcup_{m=2}^N C_m \right] \bigcup B$$

by means of the long-run frequency $x_0(\mathbf{f})$ of visits to the home node 0:

Proposition 1 *Let $\varepsilon \in (0, \frac{1}{\sqrt{N-1}})$, $\mathbf{f} \in \mathbf{F}_D$, and $\mathbf{x}(\mathbf{f})$ be its long-run frequency vector (that is, $\mathbf{x}(\mathbf{f}) = M(\mathbf{f})$). The long-run frequency of visits to the home state 0 is given by*

$$x_0(\mathbf{f}) = \sum_{a \in \mathcal{A}(1)} x_{0a}(\mathbf{f}) = \begin{cases} \frac{1}{d_{1+N}(\varepsilon)} = \frac{1}{1+N} + O(\varepsilon) & : \text{ if } \mathbf{f} \in C_{1+N} \\ \frac{1}{d_{1+k}(\varepsilon)} + O(\varepsilon) = \frac{1}{1+k} + O(\varepsilon) & : \text{ if } \mathbf{f} \in C_{1+k}, \quad k = 1, 2, \dots, N \\ \frac{\varepsilon}{1+\varepsilon} + O(\varepsilon); & : \text{ if } \mathbf{f} \in B, \end{cases}$$

where

$$d_{1+k}(\varepsilon) = \frac{1 - (1 - \varepsilon)^k}{\varepsilon} + \varepsilon + (1 + (k - 1)\varepsilon)(1 - \varepsilon)^k = 1 + k + O(\varepsilon)$$

The above proposition leads to the following characterization of HC in G :

Corollary 1 [14] *Hamiltonian Cycles of the graph G are in 1 : 1 correspondence with those points of \mathbf{X}_ε which satisfy*

$$(i) \quad x_0 = \sum_{a \in \mathcal{A}(0)} x_{0a} = \frac{1}{d_{1+N}(\varepsilon)}$$

$$(ii) \quad \text{For every } i \in \mathcal{V}, x_i = \sum_{a \in \mathcal{A}(i)} x_{ia} > 0 \text{ and } \frac{x_{ia}}{x_i} \in \{0, 1\} \text{ for each } a \in \mathcal{V}(i), i \in \mathcal{V}.$$

We now interpret the HC problems as an optimization problem. Let

$$S(\mathbf{x}) := \sum_i x_i^2 = \sum_i \left(\sum_{a \in \mathcal{A}(i)} x_{i,a} \right)^2 = \mathbf{x}^T E(G) \mathbf{x},$$

where $E(G)$ is the block-diagonal matrix consisting of $1 + N$ blocks with its i -th block being a $m_i \times m_i$ matrix full of units. Let

$$s(\mathbf{x}) := \sum_i \sum_{a \in \mathcal{A}(i)} x_{i,a}^2 = \mathbf{x}^T \mathbf{I}_m \mathbf{x},$$

where \mathbf{I}_m is the identity matrix of the size m . Consider the quadratic form

$$\begin{aligned} Q(\mathbf{x}) &:= S(\mathbf{x}) - s(\mathbf{x}) \\ &= \sum_i Q_i(\mathbf{x}_i) \\ &= \sum_i \left(\left(\sum_{a \in \mathcal{A}(i)} x_{i,a} \right)^2 - \sum_{a \in \mathcal{A}(i)} x_{i,a}^2 \right) \\ &= \mathbf{x}^T Q(G) \mathbf{x}. \end{aligned}$$

Here $Q(G)$ is a block-diagonal matrix with its i -th block being a $m_i \times m_i$ matrix with all diagonal elements equal 0 and all off-diagonal elements equal 1. We observe that

$$Q_i(\mathbf{x}_i) = \left(\sum_{a \in \mathcal{A}(i)} x_{i,a} \right)^2 - \sum_{a \in \mathcal{A}(i)} x_{i,a}^2 = \sum_{a,b \in \mathcal{A}(i), a \neq b} x_{i,a} x_{i,b},$$

is nonconvex and has nonnegative value for any nonnegative \mathbf{x}_i . $Q_i(\mathbf{x}_i)$ is zero if and only if at most one of variables $x_{i,a}$ is nonzero. Hence the (unconstrained) minimization of Q_i chooses one variable $x_{i,a}$ out of all variables in \mathbf{x}_i . In other words, it chooses one neighbour of node i in graph G .

The following indefinite quadratic programming problem is closely related to the HC problem:

$$\min \mathbf{x}^T Q(G) \mathbf{x}$$

subject to:

$$\begin{aligned} (i) \quad & \mathbf{x} \in \mathbf{X}_\varepsilon \\ (ii) \quad & x_0 = \sum_{a \in \mathcal{A}(0)} x_{0a} = \frac{1}{d_{1+N}(\varepsilon)}. \end{aligned} \tag{2}$$

Theorem 1 [14]

- (i) Let \mathbf{f} be a HC in G . Then $\mathbf{x}(\mathbf{f})$ is a global minimum of (2) and $\mathbf{x}^T Q(G) \mathbf{x} = 0$.
- (ii) Conversely, let \mathbf{x}^* be a global minimum of (2) such that $(\mathbf{x}^*)^T Q \mathbf{x}^* = 0$. Then $\mathbf{f}_{\mathbf{x}^*}^* = M^{-1}(\mathbf{x}^*)$ is a deterministic strategy which traces out a HC in G .

We note that both constraints in (2) can be expressed as a single linear equation

$$A_\varepsilon(G) \mathbf{x} = \mathbf{b}, \quad (3)$$

where

- (i) $A_\varepsilon(G)$ is $(N+3) \times m$ matrix that is obtained from W_ε by adding two rows at the bottom: one full of units and the other consisting of units at positions $(0, a)$ corresponding to the arcs emanating from the home node 0 and of zeroes elsewhere,
- (ii) \mathbf{b} is $(3+N)$ -vector consisting of zeroes for the first $1+N$ components, $b_{2+N} = 1$, and $b_{3+N} = \frac{1}{d_{1+N}(\varepsilon)}$.

It is important for numerical purposes that system (3) has maximal rank. In fact, as $\text{rank}(W_\varepsilon) = N$, the maximal rank of $A_\varepsilon(G)$ is, therefore, at most $N+2$. The lemma below ensures the maximality of the $\text{rank}(A_\varepsilon(G))$ for *communicating* graphs, (*i.e.* graphs that for any pair of nodes $\{i, j\} \subset \mathcal{V}(G)$ there exists a path between i and j in G). Without loss of generality we may assume that G is communicating, otherwise, clearly G does not contain a HC. As was shown in [16], it takes polynomial time to verify the communication property for a given graph G .

Lemma 2 *If graph G is communicating then $\text{rank}(A_\varepsilon(G)) = 2 + N$.*

The proof of this lemma is purely technical, so it is relegated to the Appendix. For our heuristic the single objective function $Q(\mathbf{x})$ is insufficient. Our algorithm requires a 1-parameter family of objective functions that (if the graph G contains an HC) achieve the (global) minimum at a HC or in a small neighbourhood of a HC so that the HC can be recognized by the location of such global minima in \mathbf{X}_ε . It appears that the convex combination

$$f_\alpha(\mathbf{x}) := S(\mathbf{x}) - \alpha s(\mathbf{x}) = \alpha Q(\mathbf{x}) + (1 - \alpha)S(\mathbf{x}), \quad 0 < \alpha \leq 1$$

provides such a family. By $\text{argmin}(f_\alpha(\mathbf{x}))$ we denote the location of a global minimum of $f_\alpha(\mathbf{x})$ in \mathbf{X}_ε .

Lemma 3 *If G contains a HC then for every $\delta > 0$ there exists $\varepsilon_0(\delta, \alpha) > 0$ such that for every ε from $0 < \varepsilon \leq \varepsilon_0(\delta, \alpha)$ a global minimum $\{\text{argmin}(f_\alpha(\mathbf{x}))\}$ lies in the δ -neighbourhood of a HC in G .*

Proof:

Let $\mathbf{x}^0 := \mathbf{x}^0(\alpha) = \text{argmin}(f_\alpha(\mathbf{x}))$ and let \mathbf{x}_{HC} be a frequency vector corresponding to HC. As $Q(\mathbf{x}_{HC}) = 0$ and, by Proposition 1, $S(\mathbf{x}_{HC}) = \frac{1}{1+N} + O(\varepsilon)$, then,

$$f_\alpha(\mathbf{x}^0(\alpha)) \leq f_\alpha(\mathbf{x}_{HC}) = \frac{1 - \alpha}{1 + N} + O(\varepsilon).$$

Since the functional $S(\mathbf{x})$ takes its minimal value for all equal values $x_i = \frac{1}{1+N}$, $i = 0, \dots, 1+N$, it follows that $S(\mathbf{x}) \geq \frac{1}{1+N}$ for all $\mathbf{x} \in \mathbf{X}_\varepsilon$. Therefore, $S(\mathbf{x}^0) = \frac{1}{1+N} + O(\varepsilon)$. Also, $Q(\mathbf{x}^0)$ should be of the order $O(\varepsilon)$, which implies that \mathbf{x}^0 is located in a small neighbourhood of an extreme point \mathbf{x}^* in \mathbf{X}_ε . Since map M establishes 1 – 1 correspondence between the extreme points of \mathbf{X}_ε and the deterministic policies F_D on Γ_ε it follows that $\mathbf{f}_{\mathbf{x}^*} := M^{-1}(\mathbf{x}^*)$ is a deterministic policy. By Proposition 1 the only available option for $\mathbf{f}_{\mathbf{x}^*}$ to satisfy $x_0(\mathbf{f}_{\mathbf{x}^*}) = \frac{1}{1+N} + O(\varepsilon)$ is that $\mathbf{f}_{\mathbf{x}^*} = HC$. \square

3 Solution of QP with an interior point method

The theory [20] and the implementation [3] of interior point methods for optimization are well understood. These methods offer a number of advantages, specially when applied to very large problems. Since we intend to apply our approach to large sparse graphs we decided to employ an interior point algorithm to solve the nonconvex quadratic problem (2). We use HOPDM solver [2, 15] for this purpose. Below we discuss the main issues of this application.

An interior point algorithm for quadratic programming implemented in HOPDM is the primal-dual method with multiple centrality correctors. The algorithm is applied to the primal-dual formulation of the quadratic program

Primal	Dual
$\begin{aligned} \min \quad & c^T x + \frac{1}{2} x^T Q x \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0; \end{aligned}$	$\begin{aligned} \max \quad & b^T y - \frac{1}{2} x^T Q x \\ \text{s.t.} \quad & A^T y + s - Qx = c, \\ & y \text{ free, } x, s \geq 0, \end{aligned}$

where $A \in \mathcal{R}^{m \times n}$, $Q \in \mathcal{R}^{n \times n}$, $x, s, c \in \mathcal{R}^n$ and $y, b \in \mathcal{R}^m$. The main computational effort of this algorithm consists in the computation of the primal-dual Newton direction. This requires solving the following linear system

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix}, \quad (4)$$

where

$$\begin{aligned} \xi_p &= b - Ax, \\ \xi_d &= c - A^T y - s + Qx, \\ \xi_\mu &= \mu e - XSe, \end{aligned}$$

and X and S denote $n \times n$ diagonal matrices in which vectors $x, s \in \mathcal{R}^n$ are spread across the diagonals, respectively. After an elimination of

$$\Delta s = X^{-1} \xi_\mu - X^{-1} S \Delta x$$

the Newton system is reduced to:

$$\begin{bmatrix} -Q - X^{-1}S & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1} \xi_\mu \\ \xi_p \end{bmatrix}. \quad (5)$$

The matrix involved in (5) is symmetric but indefinite (even for a convex problem when Q is positive definite). For the sake of efficiency, in HOPDM implementation [2], the matrix in the reduced Newton system is regularized with diagonal terms R_p and R_d

$$H = \begin{bmatrix} -Q - X^{-1}S & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -R_p & 0 \\ 0 & R_d \end{bmatrix} \quad (6)$$

to obtain a quasidefinite one [18]. This allows the use of Cholesky-like factorization in which an LDL^T decomposition is found with diagonal matrix D containing both positive and negative elements. The use of primal-dual regularization (6) guarantees the existence of Cholesky-like factorization with diagonal D and avoids the need of using the 2×2 pivots required otherwise to decompose an indefinite matrix [5, 8].

A direct application of an interior point optimizer to solve (2) faces two difficulties. First, a considerable density of diagonal blocks in $Q(G)$ causes the symmetric LDL^T decomposition of matrix H in (6) to become very dense, which adversely affects the efficiency of the interior point method. Secondly, the matrix $Q(G)$ is indefinite and the quadratic problem (2) has many local minima. Below we discuss our approach to addressing these difficulties.

Diagonal blocks in $Q(G)$ can be represented as rank-one corrections of diagonal matrices. Indeed, for

$$Q_i(\mathbf{x}_i) := \left(\sum_{a \in \mathcal{A}(i)} x_{i,a} \right)^2 - \sum_{a \in \mathcal{A}(i)} x_{i,a}^2 = \mathbf{x}_i^T Q_i \mathbf{x}_i$$

we write

$$Q_i = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} - I = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{bmatrix}.$$

By introducing an auxiliary variable

$$x_i = \sum_{a \in \mathcal{A}(i)} x_{i,a} \quad (7)$$

we transform the quadratic form $Q_i(\mathbf{x}_i)$ into a diagonal one

$$\tilde{Q}_i(\mathbf{x}_i, x_i) = x_i^2 - \sum_{a \in \mathcal{A}(i)} x_{i,a}^2.$$

Of course, we have to add one new constraint (7) and one new variable x_i for each node of the graph in the reformulated quadratic program. Therefore instead of $N + 3$ constraints and $|\mathcal{A}|$ variables in (2) the new QP formulation has $2N + 4$ constraints and $|\mathcal{A}| + N + 1$ variables. However, the system (5) obtained for such transformed problem has diagonal matrix \tilde{Q} and the interior point implementation takes full advantage of the sparsity of this separable formulation.

Nonconvexity of the quadratic programming problem means that the matrix $-Q - X^{-1}S$ in the upper left corner of system (5) is not necessarily negative definite. Consequently, there is no longer a guarantee that the small primal and dual regularizations R_p and R_d would suffice to make H in (6) quasidefinite. Since the main objective of primal-dual regularizations

[2] is to guarantee the existence of a triangular decomposition of H , one way to proceed is to accept positive pivots corresponding to the upper left corner whenever they appear and are sufficiently stable and use the primal regularization R_p only to correct pivot candidates which are dangerously close to zero. Another approach consists of correcting all positive pivot candidates in the upper left corner by subtracting large enough regularization terms. This approach is equivalent to a convexification of the objective. We have implemented the latter approach. To reduce the perturbation introduced by potentially very strong regularization we have used in our approach a different objective, namely:

$$f_\alpha(\mathbf{x}) := S(\mathbf{x}) - \alpha s(\mathbf{x})$$

which, after a transformation to a separable form, gives the following i -th term

$$\bar{Q}_i(\mathbf{x}_i, x_i) = x_i^2 - \alpha \sum_{a \in \mathcal{A}(i)} x_{i,a}^2.$$

This function is still nonconvex. However, its negative curvature decreases when α is reduced. Consequently, a weaker regularization suffices to convexify the problem. Lemma 3 provides a foundation for the use of $f_\alpha(\mathbf{x})$.

4 Heuristics based on the QP solution

For an arbitrary (nontrivial) problem there is little chance to solve the nonconvex QP and obtain the global minimum, i.e., attain zero objective value in (2). If this happens we immediately obtain a Hamiltonian Cycle. What is more likely to happen is that one of the numerous local minima of (2) is found. Such a solution has at least one set of frequencies for which there are two or more nonzero elements in $x_{i,a}$, $a \in \mathcal{A}(i)$. In the usual situation when a nonconvex QP corresponding to a large graph is solved the local optimal solution has many nodes i with such a property. Consequently, we cannot translate the solution into a HC. However, this local solution provides us with a lot of useful information and allows the use of heuristics to find a HC. The heuristics rely on the interpretation of the solution $x_{i,a}$ as frequencies of traversing an arc (i, a) in the graph.

Arc elimination

For the optimal solution (local minimum) of the quadratic problem we compute

$$f[i, a] = x_{i,a}/x_i, \quad a \in \mathcal{A}(i).$$

These variables can be interpreted as relative frequencies of leaving node i by appropriate arcs originating from i . If $f[i, a]$ is negligible, that is, if $f[i, a] < \delta$ for some prescribed tolerance δ , then the arc (i, a) is eliminated from the graph as unlikely to be a part of a HC. After arc elimination the new quadratic program for the reduced graph is solved and the analysis is repeated. After a couple of such reductions and repeated QP solutions we eventually observe that no more arcs satisfy the elimination criteria. If the last solution corresponds to the case that for each node i only one variable $f[i, a]$ is equal to 1, that is, out of all possible outgoing arcs only one is used, then a cycle is found. However if two or more of variables $f[i, a]$ are bounded away from zero, then we start branching on these variables.

Branching

Branching is a technique widely used in integer programming [19]. In our approach we analyze

the solution of the current quadratic program \mathcal{P}_k (corresponding to the reduced graph G_k) and if this solution has any node i with two or more variables which satisfy $f[i, a] \geq \delta$, then we replace the problem with a set of $|\mathcal{A}(i)|$ new problems. Each of them corresponds to a different reduced graph in which (i, a) is the only arc leaving node i and all remaining arcs which originated from i have been removed.

This way branching forces the use of one particular arc leaving node i . By replacement of the original problem with a tree of problems branching inevitably increases the computational complexity. This technique applied in the context of integer programming is often combined with other techniques such as pricing and cut generation which help to prevent uncontrollable growth of the search tree. In our approach branching strategy is combined with arc elimination technique which results in a fast reduction of the size of the graph. However, search trees may occasionally grow to very large depths. We discuss this in more detail in the next section.

5 Implementation and numerical results

The approach presented in this paper has been implemented using HOPDM interior point solver (<http://www.maths.ed.ac.uk/~gondzio/software/hopdm.html>). We have run the program on the 200 MHz Pentium III PC with Linux operating system.

Nonconvex quadratic problems of form (2) are formulated using the auxiliary variable x_i (7) to guarantee the separability. Small parameter $\alpha = 0.01$ or $\alpha = 0.001$ in $f_\alpha(x)$ is used to limit the negative curvature of the quadratic function. The arc elimination procedure is run after a local solution to the QP is found. The elimination threshold $\delta = 0.01$ has been used in our computations. Whenever an elimination occurs the new QP is solved for the reduced problem. Branching starts when no more reductions are possible. At this stage the optimization algorithm switches to the use of search tree. For a given graph (reduced by the earlier arc elimination) a family of subgraphs is created and the procedure identical as described earlier is applied to each of the subgraphs. A combination of branching and arc elimination accelerates the reduction of the graph size. However, every branching increases the size of the tree and potentially contributes to a significant growth of the computation time necessary to analyze the tree.

We have used a *hybrid* strategy to search the tree. For the first two or three levels of the tree we use the breadth-first-search order. Then the list of most attractive nodes in the tree (the ones corresponding to the smallest possible graphs) is created. Each of these nodes is processed in the same way as the initial graph, that is, a sequence of QP problems is solved each resulting with arc eliminations. When no more elimination is possible we apply branching to the problem. All children of the node are processed in a similar way. However only the most promising one (corresponding to the smallest graph) is further analyzed. In other words, we use the depth-first-search order to promote a fast decrease of the graph size. Such a strategy is promising for graphs which do have a HC.

We have applied our approach to three classes of problems:

- (i) randomly generated graphs,
- (ii) knight's tour problems,
- (iii) cubic graphs.

The first class of problems needs little introduction. We have developed these problems for the purpose of testing the approach. In Table 1 the sizes of graphs (number of nodes and

Table 1: Solution times for randomly generated graphs.

Problem	Nodes	Arcs	time (CPU secs)
rand1	25	59	1.48
rand2	30	72	0.44
rand3	40	100	3.92
rand4	50	150	7.92
rand5	100	293	107.15
rand6	110	323	12.94
rand7	120	353	67.23
rand8	130	392	19.11
rand9	140	402	147.53
rand10	150	420	1267.07

Table 2: Solution for the 6×6 chessboard problem.

4	15	34	27	6	17
35	26	5	16	33	28
12	3	14	29	18	7
25	36	11	32	21	30
10	13	2	23	8	19
1	24	9	20	31	22

arcs) and the overall CPU times in seconds needed to find a HC are reported. The analysis of results collected in Table 1 indicates the potential of the approach but also indicates the fast growth of the solution time when the size of the graph increases.

The knight’s tour problem consists in finding a tour of the knight to visit each square of the $k \times k$ chessboard exactly once. This problem has received a lot of attention from the research community and a variety of algorithms have been developed for its solutions, see for example [17]. The problem has a solution only for even $k \geq 6$. Table 2 gives the solution for the 6×6 chessboard: the numbers in the fields provide the order in which the knight visits them. In Table 3 the sizes of graphs and the corresponding solution times for the knight’s tour problems are reported. From the analysis of these results we conclude again that the approach proposed in this paper provides solution to HC problems of moderate size in acceptable computation time. We also observe that the solution time grows rapidly for a larger graph corresponding to the 32×32 chessboard.

Finally, in Table 4 we report the sizes of graphs and the CPU times needed to find Hamiltonian cycles in *Generalised Petersen* (GP) cubic graphs $GP(n, k)$ (see [1] for the discussion of Hamiltonicity of the GP graphs).

Table 3: Solution times for the knight's tour problem.

Problem	Nodes	Arcs	time (CPU secs)
chess6	36	160	1.25
chess8	64	336	3.35
chess10	100	576	29.77
chess12	144	880	33.58
chess14	196	1248	194.45
chess20	400	2736	819.10
chess32	1024	7440	35697.00

Table 4: Solution times for cubic graphs.

Problem	Nodes	Arcs	time
GP(29,5)	58	174	12.26
GP(41,5)	82	246	29.42
GP(59,5)	118	354	29.42
GP(89,8)	178	534	60.97

6 Appendix

Proof of Lemma 2:

If there is a single arc emanating from every node of G , the communication property implies that G is itself a HC . For $G = HC$ the claim is straightforward. Assume that there is a node i with at least two arcs (i, a_0) and (i, a_1) in $\mathcal{A}(i)$. We reorder the nodes of G so that node a_0 becomes the home node “0”, node i becomes node “2” and node a_1 becomes node “1”. This will now guarantee the maximality of $\text{rank}(A_\varepsilon(G))$. As $A_\varepsilon(G)$ is a ε -perturbation of $A_0 := A_{\varepsilon=0}(G)$, it will be sufficient to prove that $\text{rank} A_0 = 2 + N$. Let $\gamma_{0,2}$ be a shortest (*w.r.t* the number of involved arcs) path from “0” to “2”. Suppose that $\gamma_{0,2}$ traces out the path $0 \rightarrow i_1 \rightarrow i_2 \rightarrow \dots \rightarrow i_s \rightarrow 2$. Suppose there exists a vanishing non-trivial linear combination of the rows A_j , $j = 0, 1, \dots, 2 + N$ of the matrix A_0 :

$$\nu A_{2+N} + \mu A_{1+N} + \sum_{i=0}^N \lambda_i A_i = 0. \quad (8)$$

Since $\sum_{i=0}^N \lambda_i A_i = 0$, we assume that $\lambda_0 = 0$. For the arc $(0, i_1)$ equation (8) implies:

$$\lambda_0 - \lambda_{i_1} + \mu = 0,$$

hence,

$$\lambda_{i_1} = \mu \quad (9)$$

For consecutive arcs $(i_1, i_2), \dots, (i_{s-1}, i_s), (i_s, 2)$ equation (8) further reads:

$$\lambda_{i_1} - \lambda_{i_2} + \nu = 0, \dots, \lambda_{i_{s-1}} - \lambda_{i_s} + \nu = 0, \lambda_{i_s} - \lambda_2 + \nu = 0.$$

So, by (9) and by induction on s one obtains:

$$\lambda_{i_2} = \lambda_{i_1} + \nu = \mu + \nu, \dots, \lambda_{i_s} = \mu + (s-1)\nu, \quad \lambda_2 = \mu + s\nu. \quad (10)$$

Arcs $(2, 0)$ and $(2, 1)$, respectively, contribute:

$$\lambda_2 - \lambda_0 + \nu = 0$$

and

$$\lambda_2 - \lambda_1 + \nu = 0.$$

Combined with (10) this implies:

$$\mu = -(s+1)\nu \quad \text{and} \quad \lambda_1 = 0. \quad (11)$$

Choose $\gamma_{1,0}$ to be the shortest path $1 \rightarrow j_1 \rightarrow \dots \rightarrow j_r \rightarrow 0$ from “1” to “0”. For the arc $(1, j_1)$ equation (8) implies:

$$\lambda_1 - \lambda_{j_1} + \nu = 0, \quad \text{hence,} \quad \lambda_{j_1} + \nu.$$

Arc (j_1, j_2) in (8) contributes

$$\lambda_{j_1} - \lambda_{j_2} + \nu = 0, \quad \text{so,} \quad \lambda_{j_2} = \lambda_{j_1} + \nu = 2\nu.$$

Consideration of consecutive arcs in $\gamma_{1,0}$ up to (j_{r-1}, j_r) and induction on r lead to

$$\lambda_{j_t} = t\nu, \quad t = 1, 2, \dots, r.$$

Finally, arc $(j_r, 0)$ implies:

$$\lambda_{j_r} - \lambda_0 + \nu = 0, \quad \text{so} \quad \nu = 0 \quad \text{since} \quad \lambda_0 = 0.$$

It now follows from (11) that $\mu = 0$. It remains to prove that the only vanishing linear combination $\sum_{i=0}^N \lambda_i A_i$ with $\lambda_0 = 0$ has to be trivial. This follows from the property of the incidence matrix $D(G)$ that its rank equals the number of nodes take the number of connected components of the graph ([6], p.23-24). For the simple situation of a single connected component (as in the case of a communicating graph) we supply a short proof: denote by \mathcal{V}_0^j those nodes that can be reached from node “0” by a path of length j . Then the entire $\mathcal{V}(G)$ admits a representation

$$\mathcal{V}(G) = \bigcup_{j=0}^{j=p \leq N} \mathcal{V}_0^j.$$

For each $i_1 \in \mathcal{V}_0^1$ arc $(0, i_1)$ in (8) contributes:

$$\lambda_0 - \lambda_{i_1} = 0, \quad \text{so} \quad \lambda_{i_1} = 0.$$

For each $i_2 \in \mathcal{V}_0^2$ there exists $i_1 \in \mathcal{V}_0^1$ adjacent to “ i_2 ”. So,

$$\lambda_{i_1} - \lambda_{i_2} = 0, \quad \text{hence,} \quad \lambda_{i_2} = 0.$$

Eventually, for each $i_p \in \mathcal{V}_0^p$ there is arc (i_{p-1}, i_p) in G with $i_{p-1} \in \mathcal{V}_0^{p-1}$. Thus,

$$\lambda_{i_p} = \lambda_{i_{p-1}} = 0.$$

□

References

- [1] B. ALSPACH, *The classification of Hamiltonian Generalized Petersen graphs*, Journal of Combinatorial Theory, ser. B 34 (1983), pp. 293–312.
- [2] A. ALTMAN AND J. GONDZIO, *Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization*, Optimization Methods and Software, 11–12 (1999), pp. 275–302.
- [3] E. D. ANDERSEN, J. GONDZIO, C. MÉSZÁROS, AND X. XU, *Implementation of interior point methods for large scale linear programming*, in Interior Point Methods in Mathematical Programming, T. Terlaky, ed., Kluwer Academic Publishers, 1996, pp. 189–252.
- [4] M. ANDRAMONOV, J. A. FILAR, A. RUBINOV, & P. PARDALOS, *Hamiltonian cycle problem via Markov chains and min-type approaches*, in P. M. Pardalos (Ed.), *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems* Dordrecht, The Netherlands: Kluwer Academic Publishers, 2000, pp. 31–47.
- [5] J. R. BUNCH AND B. N. PARLETT, *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM Journal on Numerical Analysis, 8 (1971), pp. 639–655.
- [6] N. L. BIGGS, *Algebraic Graph Theory*, Cambridge University Press, London, New York, 1974.
- [7] M. CHEN AND J. A. FILAR (1992) on Hamiltonian cycles, quadratic programming and ranking of extreme points, in C. A. Floudas & P. M. Pardalos (Eds.), *Recent Advances in Global Optimization*, Princeton, New Jersey: Princeton University Press, 1992, pp. 32–49.
- [8] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct methods for sparse matrices*, Oxford University Press, New York, 1987.
- [9] V. EJOV, J. FILAR AND J. THREDGOLD, *Geometric interpretation of Hamiltonian Cycles problem via singularly perturbed Markov decision processes*, CIAM Technical report, submitted for publication
- [10] E. FEINBERG, *Constrained discounted Markov decision processes with Hamiltonian cycles*, Mathematics of Operations Research, **25** (2000), pp. 130–140.
- [11] J. FILAR AND D. KRASS, *Hamiltonian cycles and Markov chains*, Mathematics of Operations Research, **19** (1994), pp. 223–237.
- [12] J. A. FILAR, J. A. AND J. B. LASSERRE, *A non-standard branch and bound method for the Hamiltonian cycle problem*, ANZIAM J. **42**(E)(2000), 556–577.
- [13] J. FILAR AND T. SCHULTZ, *Communicating MDPs: equivalence and LP properties*, Operations Research Letters, 7 (1988), pp. 303–307.
- [14] J. FILAR AND K. VRIEZE, *Competitive Markov Decision Processes*, Springer, Berlin, 1996.

- [15] J. GONDZIO, *Multiple centrality corrections in a primal-dual method for linear programming*, Computational Optimization and Applications, 6 (1996), pp. 137–156.
- [16] L. KALLENBERG, *Classification problems in MDPs*, in Markov Processes and Controlled Markov Chains, X. WhoIsIt, ed., Kluwer Academic Publishers, 2002, pp. 151–166
- [17] I. PARBERRY, *Scalability of a neural network for the knight’s tour problem*, Neurocomputing, 12 (1996), pp. 19–34.
- [18] R. J. VANDERBEI, *Symmetric quasidefinite matrices*, SIAM Journal on Optimization, 5 (1995), pp. 100–113.
- [19] L. A. WOLSEY, *Integer Programming*, John Wiley & Sons, New York, 1998.
- [20] S. J. WRIGHT, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.