A non-standard branch and bound method for the Hamiltonian cycle problem

J.A. Filar* Jean B. Lasserre†

(Received 7 August 2000)

Abstract

In this note, we consider an embedding of a Hamiltonian cycle problem in a Markov decision process (MDP). We propose a branch & bound type method based on the frequency polytope resulting from this embedding. Among the special features of the proposed scheme are the properties that: (i) a three-way branching is the biggest that

*University of South Australia, Adelaide, AUSTRALIA.
mailto:jerzy.filar@unisa.edu.au
†LAAS-CNRS, Toulouse, FRANCE. mailto:lasserre@laas.fr

can occur, (ii) no integer-valued variable is required, and (iii), the size of the LPs solved at the nodes of the branch & bound tree can only decrease as one moves down the tree. We hope that this note will encourage researchers in combinatorial optimization to experiment with the Markov decision process embedding as a basis for new algorithmic procedures.

Contents

1 Introduction **C588**

2 MDP embedding **C589**

3 A Branch & Bound algorithm **C592**

3.1 Notation and preliminaries C592

3.2 The main idea C596

3.3 A B & B procedure for a Hamiltonian cycle ending in node k C600

4 Example **C602**

References **C606**

1 Introduction

The well-known Hamiltonian Cycle Problem (HCP in short) can be described as follows:

HCP: *In a directed graph, find a path that enters every node exactly once before returning to the starting node, or determine that no such path exists.*

The Hamiltonian cycle problem is considered to be a very difficult problem to solve and it is well-known to be NP-hard. However, there are good heuristic algorithms for solving many instances of the Hamiltonian cycle problem, most of which are based on combinatorial approaches. In this paper, we use an unorthodox approach to the Hamiltonian cycle problem, first initiated by Krass and Filar [4], and then developed in Chen and Filar [7][1]. The approach has been continued by Feinberg [2]. The Hamiltonian cycle problem problem is first embedded in a controlled Markov chain and the induced ergodic structure is then exploited. In this approach, the standard linear program often used to solve the average-cost problem for Markov decision processes plays a central role.

Using the same formulation, Andramonov et al. [1] proposed a mixed-integer programming solution method. This yielded encouraging numerical

[1] Despite the fact that [7] appeared three years earlier, it was in fact a continuation of the work started in Krass and Filar [4].

results (e.g., problems with hundred nodes and three hundred arcs could be solved in approximately two minutes with the help of CPLEX subroutines). However, the mixed-integer programming of [1], did not exploit many (except for the very basic) features of the stochastic embedding. The branch & bound procedure proposed here makes use of the special properties of the solutions of the standard linear programs for average-cost Markov decision processes.

2 MDP embedding

We consider the following problem: Given a directed graph G with N nodes, find a simple cycle of N arcs that is a Hamiltonian cycle (HC in short), or determine that none exists. For a good account of the classical approaches to this problem, the interested reader is referred to the books of Papadimitriou and Steiglitz [8] and Lenstra and Rinnooy Kan [6]

Consider a graph G with a set of N nodes $E := \{1, 2, \ldots, N\}$, and let $A(i)$ be the set of arcs emanating from Node $i \in E$. Let $A := \cup_{i \in E} A(i)$. Assume that $n_i := |A(i)| \geq 1$ for every $i \in E$. An arc a, emanating from a Node $i \in E$, i.e. $a \in A(i)$, is associated with a pair (i, a). Therefore, the notation $(i, a) \rightsquigarrow j \in E$ simply means that the arc a links nodes i and j, that is, an arc comes out from Node i and comes in at Node j.

Given some fixed scalar $\epsilon \in (0, 1)$, and using the arcs from the original graph G, we construct the transition law of a controlled Markov chain as

follows: Let

$$p_{iaj}(\epsilon) := \begin{cases} 1 & \text{if } i = 1 \text{ and } (i,a) \rightsquigarrow j \\ 0 & \text{if } i = 1 \text{ and } (i,a) \not\rightsquigarrow j \\ 1 & \text{if } i > 1 \text{ and } (i,a) \rightsquigarrow j = 1 \\ \epsilon & \text{if } i > 1,\ j = 1 \text{ and } (i,a) \not\rightsquigarrow j \\ 1-\epsilon & \text{if } i > 1,\ (i,a) \rightsquigarrow j > 1 \\ 0 & \text{if } i > 1,\ j > 1 \text{ and } (i,a) \not\rightsquigarrow j. \end{cases} \tag{1}$$

The number $p_{iaj}(\epsilon)$ is interpreted as an ϵ-perturbed probability of a transition from Node i to Node j, if an arc a is selected (with $(i,a) \rightsquigarrow j$). Note that a selection of an arc at a node of the graph G constitutes a choice of an action in a state of the Markov decision process in which the graph has been embedded. Thus it will be convenient to use the same symbol a to denote both such an arc and the index of the corresponding action; as has been done above. Note also that if $\epsilon = 0$, these probabilities are 0 or 1, according to the most natural interpretation (see [3], [4] or [7] for more details).

We now consider the following linear system:

$$\sum_{i \in E} \sum_{a \in A(i)} (\delta_{ij} - p_{iaj}(\epsilon)) x_{ia} = 0, \quad j \in E, \tag{2}$$

$$\sum_{i \in E} \sum_{a \in A(i)} x_{ia} = 1 \tag{3}$$

$$\sum_{a \in A(1)} x_{1a} = 1/d_N(\epsilon) \tag{4}$$

$$x_{ia} \geq 0,\ i \in E,\ a \in A(i), \tag{5}$$

where δ_{ij} is the usual Kronecker symbol.

When associated with a linear objective criterion $\sum_{i,a} c(i,a)x(i,a)$, the linear system (2–3), (5) becomes the standard linear program used to find the minimum average cost and to identify the corresponding ergodic class, in an average-cost Markov decision process [9, e.g.]).

A stationary "deterministic policy" is a mapping $f : E \to A$ with $f(i) \in A(i)$, $i \in E$. Under such a policy, whenever the "system" is in state $i \in E$, one chooses the action $a := f(i) \in A(i)$ and the system moves to a state $j \in E$, with probability $p_{iaj}(\epsilon)$. The set of stationary deterministic policies is denoted by $C(\mathcal{D})$. Hence, for a deterministic policy $f \in C(\mathcal{D})$, $f(i)$ identifies an arc (i,a) in the original graph G.

In [4], it was shown that when (2–5) is associated with the Hamiltonian cycle problem problem, every feasible solution $\{x_{ia}\}$ to the above linear system identifies a Hamiltonian cycle if and only if for every $i \in E$, there exists a single action $a_i \in A(i)$ such that $x_{ia_i} > 0$.

However, for constrained Markov decision processes, in general, an optimal solution to a linear program with (2–5) as constraints, may identify a rather simple "randomized" policy. That is, there will be at most one Node i, such that $x_{ia} > 0$ for two "actions" a_1 and a_2. Hence, in general, it is not possible to obtain a Hamiltonian cycle by just one run of a linear program with (2–5) as constraints. Nonetheless, the two actions where the randomization occurs might provide a useful clue as to where branching should occur.

Therefore, the methodology proposed below is a heuristic to find a Hamiltonian cycle, using a linear program (with (2–5) as constraints) to branch and bound. It might be interesting to emphasize that unlike standard branch & bound methods, the size of the LPs that are solved at progressive nodes of the branch & bound tree, decreases rather than increases, as one moves down the tree.

3 A Branch & Bound algorithm

In this section, we propose a Branch & Bound algorithm based on the above Markov decision process embedding. We first introduce the notation and present the basic principles before describing the algorithm.

3.1 Notation and preliminaries

The underlying polytope that is of interest to us is the "long-run frequency space" of the ϵ-perturbed Markov decision process associated with our graph

G, via (1).

$$X := \left\{ x \mid \sum_{i\in E}\sum_{a\in A(i)} (\delta_{ij} - p_{iaj}(\epsilon))x_{ia} = 0\,,\ j \in E \right.$$
$$\sum_{i\in E}\sum_{a\in A(i)} x_{ia} = 1\,,$$
$$\left. x_{ia} \geq 0, \quad i \in E,\ a \in A(i) \right\}$$

Note that when $\epsilon = 0$, the main set of constraints defining X reduce (see (1)) to the "conservation of flow" constraints:

$$\sum_{a\in A(j)} x_{ja} = \sum_{i\in E}\sum_{a\in A(i)} p_{iaj}(0)x_{ia} = \sum_{(i,a)\rightsquigarrow j} x_{ia},$$

where the last summation is over all arcs incident on j; for each $j \in E$.

For any fixed $\epsilon \in (0, 1)$ define the following quantities:

- $\hat{x}_i := \sum_{a\in A(i)} x_{ia},\ i \in E.$
- $\gamma(\epsilon) := \epsilon(1-\epsilon)^{-1}.$
- $d_i(\epsilon) := 1 + \sum_{k=2}^{i}(1-\epsilon)^{k-2},\ i = 2, 3, \ldots, N.$
- $H := \{x \mid \hat{x}_1 = d_N(\epsilon)^{-1}\}.$

- $X_c := X \cap H$.

Given a convex polyhedron Ω, let $\text{ext}(\Omega)$ be the set of its extreme points.

Of course, it should be noted that each deterministic policy $f \in C(\mathcal{D})$ identifies a subgraph $G_f \subset G$ via the correspondence

$$f(i) = a \Leftrightarrow \text{ arc}(i, a) \in G_f; \quad i \in E.$$

If G_f is a Hamiltonian cycle, we shall say that f is a Hamiltonian cycle as well.

Prior facts: (see Kallenberg [5] or Puterman [9] for 1–3, Filar and Krass [4] for 4, and Chen and Filar [7] or Filar and Vrieze [3] for 5–6).

1. If $x^* \in \text{ext}(X)$, then $x_{ia} > 0$ for at most one action a, for each $i \in E$.
2. If $f \in C(\mathcal{D})$ then $\{(i, f(i))\}$, $i \in E$, identifies an extreme point $x(f) \in \text{ext}(X)$, with $x(f) = \{x_{if(i)}\}$.
3. If $x^* \in \text{ext}(X_c)$, then there exists at most one $i \in E$ for which there are two actions $a \neq b$ s.t. $x^*_{ia}, x^*_{ib} > 0$.

4. Let $f^* \in C(\mathcal{D})$ be a Hamiltonian cycle and let $x^* = x(f^*)$. If $i \in E$ is the kth node on the cycle (when starting from Node 1), then

$$\hat{x}_i^* = \sum_{a \in A(i)} x_{ia}^* = \begin{cases} d_N(\epsilon)^{-1}, & \text{if} \quad k = 1 \text{ or } 2 \\ (-\epsilon)^{k-2} d_N(\epsilon)^{-1} & \text{if} \quad k = 3, 4, \ldots, N \end{cases} \tag{6}$$

5. The class of deterministic policies can be partitioned into a disjoint union

$$C(\mathcal{D}) = C_1 \cup C_2 \cup \cdots \cup C_N \cup B, \tag{7}$$

such that if

$$\begin{aligned} f \in C_i \quad &\text{then} \quad \hat{x}_1(f) = d_i(\epsilon)^{-1};\ i \neq 1, \\ f \in B \quad &\text{then} \quad \hat{x}_1(f) = \epsilon(1+\epsilon)^{-1}. \end{aligned}$$

Clearly, C_N is the (possibly empty) class of Hamiltonian cycles.

6. For all $\epsilon \in (0, 1)$

$$2^{-1} = d_2(\epsilon)^{-1} > d_3(\epsilon)^{-1} > \cdots > d_N(\epsilon)^{-1} > \epsilon(1+\epsilon)^{-1}. \tag{8}$$

Thus (8) can be used to differentiate between the deterministic policies belonging to different elements of the partition (7).

3.2 The main idea

Property 3 in Section 3.1 suggests a natural "3-way branching" that will be developed next and property 6 suggests a bounding scheme for fathoming nodes in a branch & bound tree.

The search for a Hamiltonian cycle will be based on a search for a Hamiltonian cycle ending in a Node k, which means that k is the last node on the Hamiltonian cycle before return to Node 1. If this search proves unsuccessful, it can be repeated with another candidate for the last node on a Hamiltonian cycle. Clearly, at most $(N-1)$ such searches need to be carried out.

Notation: Let G be a graph and $\Gamma(\epsilon)$ be the corresponding perturbed Markov decision process with the transition law given by (1). Assume that Node k has at least one arc of the form $(k,1)$, i.e., one may reach Node 1 from Node k. Let

$$\begin{aligned} \mathcal{N}_k &:= \{\text{conclusion: } \nexists \text{ a HC ending in node } k \text{ in the current graph}\} \\ \mathcal{Y}_k &:= \{\text{conclusion: } \exists \text{ a HC ending in node } k \text{ in the current graph}\}. \end{aligned}$$

Let $r = 0, 1, 2, \ldots$ denote the current branch of the branch and bound search tree and G^r be the corresponding subgraph of G (with $\Gamma^r(\epsilon)$ being the corresponding perturbed Markov decision process, and X^r the corresponding frequency space). Let (i,a) and (i,b) be a pair of arcs from Node $i \in G^r$. Construct three subgraphs of G^r as follows:

- G^r_{ia} := same as G^r except that all arcs coming out of i have been replaced by the single arc (i, a).
- G^r_{ib} := same as G^r except that all arcs coming out of i have been replaced by the single arc (i, b).
- G^r_{i-ab} := same as G^r except that both arcs (i, a) and (i, b) have been deleted

Branching. Let *"3-way branch on the ith node of G^r"* mean that we shall now consider

$$G^{r+1} := G^r_{ia}; \quad G^{r+2} := G^r_{ib}; \quad G^{r+3} := G^r_{i-ab},$$

and with the convention $G^0 := G$.

Note that every time we go to the branch corresponding to either G^r_{ia} or G^r_{ib}, we are fixing one particular arc in the graph. This arc would remain fixed for all subsequent nodes along that branch of the branch and bound tree. Therefore, before solving an LP at that branch and bound node, one should first check if the fixed arcs up to this point contain a cycle. If so, and if such a cycle is not a Hamiltonian cycle, then this branch and bound node should be fathomed. We call this a *subcycle check*.

Bounding: Consider whether a fixed node, say Node $k \in E$, can be the *last* node on a Hamiltonian cycle that returns to Node 1 [of course, for this

to happen, there must exist an arc of the form $(k, 1)]$. However, if k were the last node on a Hamiltonian cycle, then by (6)

$$\hat{x}_k^* = \sum_{a \in A(k)} x_{ka}(f^*) = (1-\epsilon)^{N-2} d_N(\epsilon)^{-1}. \tag{9}$$

Choose $\epsilon \in (0, 1)$ such that

$$L := (1-\epsilon)^{N-2} d_N(\epsilon)^{-1} < \epsilon(1+\epsilon)^{-1}. \tag{10}$$

Then:

1. L will be a lower bound to determine the fathoming in the search tree of the branch and bound procedure. Note that by (9) if k were the last node on a Hamiltonian cycle corresponding to f^*, then $x_k(f^*) = L$.

2. Because the perturbed MDP with transition law given by (1) is "unichain", the deterministic policies correspond to the extreme points of X. To be more precise, to every stationary policy f with stationary distribution ν, corresponds an extreme point that identifies its stationary distribution (i.e. $\sum_a x(i,a) = \nu(i)$, for the recurrent states i.) Therefore, we can now further partition $C(\mathcal{D})$ (and therefore, ext(X)) with respect to Node k, as follows:

$$C(\mathcal{D}) = \left[\left(\bigcup_i \underline{C}_i^k \right) \cup \underline{B}^k \right] \cup \left[\left(\bigcup_i \overline{C}_i^k \right) \cup \overline{B}^k \right] = U^k \cup V^k,$$

where

$$f \in U^k \quad \Leftrightarrow \quad \hat{x}_k(f) = \sum_{a \in A(k)} x_{ka}(f) \leq L$$
$$f \in V^k \quad \Leftrightarrow \quad \hat{x}_k(f) = \sum_{a \in A(k)} x_{ka}(f) > L.$$

3. Note that by (9), all the Hamiltonian cycles (if any) *ending in Node k* must lie in U^k. In fact, if $f \in U^k$, then either

$$\text{(A)} \quad 0 < \hat{x}_k(f) \leq L, \tag{11}$$

or

$$\text{(B)} \quad \hat{x}_k(f) = 0 \text{ and } k \text{ is transient in the Markov Chain induced by } f. \tag{12}$$

Note that whenever a node of the branch and bound tree will be fathomed by an appropriate LP having an objective function value greater than L, this will, typically, eliminate a large portion of $C(\mathcal{D})$ from further consideration, namely, all the $f \in V^k$ that were still among the extreme points of the current LP.

3.3 A B & B procedure for a Hamiltonian cycle ending in node k

The method begins with the original graph G being the "root" node, G^0, of the branch and bound tree. Associated with it, are the long-run frequency space $X^0 := X$, the cut-space $X_c^0 = X^0 \cap H$ (see Section 3.1) and a linear program

$$\text{LP}^0: \quad z^0 := \{\min \hat{x}_k \mid x \in X_c^0\}.$$

Of course, if $z^0 = L$ and x^0 (the optimal solution to LP^0) is a Hamiltonian cycle ending in k (identified by the positive elements $x_{ia}^0 > 0$ of x^0), then the search is over. In general, however, the following branch-and-bound recursive procedure will be required.

B & B algorithm

Step 1 (Testing and Bounding): At the current iteration, r, we are considering a node of the branch and bound tree corresponding to a subgraph G^r of G and an associated linear program

$$\text{LP}^r: \quad z^r := \{\min \hat{x}_k \mid x \in X_c^r\},$$

where X^r is the long-run frequency space corresponding to G^r and $X_c^r = X^r \cap H$. Apply the following fathoming tests:

1. If G^r has been obtained from a previous branch and bound node by fixing an arc (i, a), then
   (a) if $(i, a) \rightsquigarrow 1$ and $i \neq k$, then $\mathcal{N}_k$ and fathom G^r;
   (b) if $(i, a) \rightsquigarrow k$ and $i = 1$, then $\mathcal{N}_k$ and fathom G^r;
   (c) if the fixed arcs of G^r form a cycle of length less than N, then $\mathcal{N}_k$ and fathom G^r.
2. If LP^r is infeasible, then $\mathcal{N}_k$ and fathom G^r.
3. If $z^r > L$, then $\mathcal{N}_k$ and fathom G^r.
4. If $z^r < L$ compute
$$\bar{z}^r := \{\max \hat{x}_k \mid x \in X_c^r\},$$
and if $\bar{z}^r < L$, then $\mathcal{N}_k$ and fathom G^r.
5. If $0 \leq z^r \leq L$ and x^r (the optimal basic solution of LP^r) is not a Hamiltonian cycle, then there exist $i \in E$ and arcs (i, a) and (i, b) in G^r, such that x_{ia}, $x_{ib} > 0$. Go to Step 2.
6. If $z^r = L$ and x^r is a Hamiltonian cycle, then $\mathcal{Y}_k$.

Step 2 (3-way branch): Suppose that condition (v) is met at the current subgraph G^r and associated LP^r. Do a 3-way branch on Node i, to obtain $G^{r+1}, G^{r+2}, G^{r+3}$ and the corresponding LP's, LP^{r+1}, LP^{r+2} and LP^{r+3}. Return to Step 1 with each G^{r+1}, G^{r+2} and G^{r+3}.

Remarks:

1. Note that since each one of G^{r+1}, G^{r+2} and G^{r+3} has at least one arc less than G^r, the above process must terminate finitely; either when all the nodes have been fathomed, or when a Hamiltonian cycle ending in Node k has been found. Furthermore, the dimensions of LP^{r+s} are less than those of LP^r, for each $s = 1, 2, 3$.

2. At this stage, we do not recommend any particular order of searching through the unfathomed nodes of the branch and bound tree.

4 Example

In this section, we illustrate the preceding branch and bound algorithm on a single example. In particular, we consider a complete graph G on $N = 5$ nodes with no self-loops. The value of the perturbation parameter was set at $\epsilon = 0.5$. This immediately implies that $d_5(\epsilon)^{-1} = 0.347826$, $L = (1 - \epsilon)^3 d_5(\epsilon)^{-1} = 0.043478$ and the inequality (10) is satisfied.

For the purpose of demonstrating the branch and bound algorithm, we shall pretend that we do not know that there are Hamiltonian cycles ending in Node $k = 3$ in G and set our algorithm the task of finding one. Thus, the

root graph $G^0 = G$ and LP^0 is a linear program of the form

$$\text{LP}^0 \left\{ \begin{array}{lll} \min\ c^T x & & \\ Ax & = & b \\ x & \geq & 0, \end{array} \right.$$

where A is a 7×20 matrix, c^T is a 1×20 vector and b is a 7×1 vector. In particular, (2–5) imply that $c^T = (0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0)$, $b^T = (0,0,0,0,0,1,0.347826)$ and $A =$

$$\left[\begin{array}{cccc:cccc:cccc:cccc:cccc} 1 & 1 & 1 & 1 & -1 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -1 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -1 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -1 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ -1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & -\frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & -\frac{1}{2} \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

An optimal solution of LP^0 yields $z^0 = 0$ and x^0 which has positive entries corresponding to arcs $(4,1)$ and $(4,2)$. Thus, Step 2 of the algorithm leads to three branch and bound nodes corresponding to subgraphs

$$G^1 = G^0_{(41)}, \quad G^2 = G^0_{(42)}, \quad G^3 = G^0_{(4-12)}.$$

Now, G^1 can be fathomed immediately in view of the test 1a of Step 1. We then set up the linear program LP^2 corresponding to G^2. Note that this is

easily done by deleting three columns (and corresponding variables) from A and c^T.

An optimal solution of LP^2 yields $z^2 = 0$ and x^2 which has positive entries corresponding to arcs $(2,1)$ and $(2,4)$. This yields to three more branch and bound nodes corresponding to the subgraphs

$$G^4 = G^2_{(21)}, \quad G^5 = G^2_{(24)}, \quad G^6 = G^2_{(2-14)}.$$

Now, G^4 can be fathomed by the test 1a of Step 1 and G^5 can be fathomed by the test 1c of Step 1.

At this stage, we still have two unfathomed nodes (G^3 and G^6) of the branch and bound tree.

We choose to set up and solve LP^6 corresponding to Node G^6 of the tree[2]. This is done by deleting the columns (and variables) corresponding to the arcs $(2,1)$ and $(2,4)$ in the preceding linear program LP^2.

This yields $z^6 = 0$ and x^6 which has positive entries corresponding to arcs $(5,1)$ and $(5,4)$. Step 2 now yields branch and bound nodes

$$G^7 = G^6_{(51)}, \quad G^8 = G^6_{(54)}, \quad G^9 = G^6_{(5-14)}.$$

Node G^7 is fathomed by the test 1a of Step 1, and LP^8 is formed and solved next.

[2]This corresponds to a move down a branch rather than a jump to another branch. We do not claim that this is the best strategy

This yields $z^8 = L$ and x^8 which has positive entries corresponding to arcs $(1,3)$ and $(1,5)$. Step 2 now yields nodes

$$G^{10} = G^8_{(13)}, \quad G^{11} = G^8_{(15)}, \quad G^{12} = G^8_{(1-35)}.$$

Node G^{10} is fathomed by the test 1b of Step 1 and LP^{11} is formed and solved. This yields $z^{11} = L$ and x^{11} which identifies only one arc per node and forms the Hamiltonian cycle

$$1 \to 5 \to 4 \to 2 \to 3 \to 1.$$

Hence, in order to find this Hamiltonian cycle, we needed to generate 12 nodes of the branch and bound tree and solve 5 linear programs. Considering that there are $4^5 = 1024$ subgraphs in G that correspond to deterministic policies $f \in C(\mathcal{D})$ which (as far as the algorithm is concerned) are all possible candidates for a Hamiltonian cycle ending in Node 3, and that among these candidates, there are only $4! = 24$ Hamiltonian cycles of any kind, this might be regarded an efficient run.

Of course, the efficiency of this algorithm needs to be tested on a significant sample of problems, but this is beyond the scope of this note that was intended merely to stimulate interest in such a numerical experimentation.

Acknowledgements: This research began during a visit at LAAS while the first author was an invited professor at Paul Sabatier University in Toulouse, and continued in Adelaide, while the second author visited University of South Australia and was supported by the ARC grant #A49906132.

References

[1] M. Andramonov, J.A. Filar, P. Pardalos, and A. Rubinov. Hamiltonian Cycle Problem via Markov Chains and Min-type Approaches. In: P.M. Pardalos, editor, *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems*, Kluwer Academic Publishers, pp. 31–47, 2000. C588, C589

[2] E. Feinberg. Constrained Discounted Markov Decision Processes and Hamiltonian Cycles, *Math. Oper. Res.*, 25:130–144, 2000. C588

[3] J.A. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, New York, 1996. C590, C594

[4] J. Filar and D. Krass. Hamiltonian cycles and Markov chains. *Math. Oper. Res.*, 19:223–237, 1995. C588, C588, C590, C591, C594

[5] L.M.C. Kallenberg. *Linear Programming and Finite Markov Control Problems*. C.W.I., Amsterdam, 1983. C594

[6] E. Lawler. J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys. *The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization*. Wiley, Chichester, 1985. C589

[7] M. Chen and J. Filar. Hamiltonian cycles, quadratic programming and ranking of extreme points. in C. Fouldas and P. Pardalos, editors,

Global Optimization, Princeton University Press, 1992. C588, C588, C590, C594

[8] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, New Jersey, 1982. C589

[9] M.L. Puterman. *Markov Decision Processes*. Wiley, New York, 1994. C591, C594