Hamiltonian Cycle Problem and Cross Entropy: Passing Through the Nodes by Learning

Ali Eshragh, Jerzy A. Filar and Michael Haythorpe

Centre for Industrial and Applied Mathematics (CIAM)
University of South Australia, Mawson Lakes SA 5095 Australia

A Conference on the Occasion of R.Y. Rubinstein's 70th Birthday
Sønderborg, Denmark
14-18 July, 2008

The Hamiltonian Cycle Problem: An Introduction

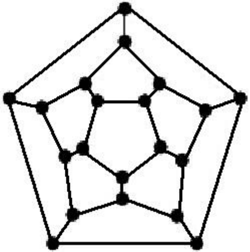
- The essence of the Hamiltonian Cycle Problem (HCP) is contained in the following - deceptively simple - single sentence statement:

Given a graph, find a simple cycle that contains all vertices of the graph (Hamiltonian cycle (HC)) or prove that one does not exist.

- The essence of the Hamiltonian Cycle Problem (HCP) is contained in the following - deceptively simple - single sentence statement:

Given a graph, find a simple cycle that contains all vertices of the graph (Hamiltonian cycle (HC)) or prove that one does not exist.

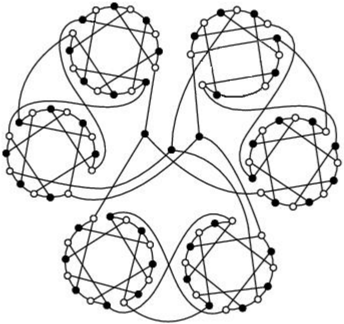
- With respect to this property - Hamiltonicity - graphs possessing a HC are called Hamiltonian.
- The name is due to the fact that Sir William Hamilton investigated the existence of such cycles on the dodecahedron graph.



The Hamiltonian Cycle Problem: An Introduction

- Henceforth, a graph of order N will mean a simple N-vertex graph (without self-loops) that can be both:
  - symmetric - every edge admits two-way traffic, or
  - directed - digraph, with one-way traffic along each arc.
- The HCP is NP-complete and has become a challenge both in its own right and because of its close relationship to the famous Travelling Salesman Problem (TSP).
- An efficient solution of TSP would have an enormous impact in operations research, optimization and computer science.
- I claim that the underlying difficulty of the TSP is, perhaps, hidden in the Hamiltonian Cycle Problem and focus on the latter.

- Just to demonstrate the connection of HCP with “rare events”
- I claim that even some “simple” Hamiltonian Graphs, can be tough! For example, consider the “Horton94 Graph”.



Horton Graph

- The standard, Horton graph is a 96-node cubic graph and is non-Hamiltonian. The “Horton94” graph is a simple 94-node modification (Ejov) which has only “a few” Hamiltonian cycles: (exactly?) $76,800$.

Horton Graph

- The standard, Horton graph is a 96-node cubic graph and is non-Hamiltonian. The "Horton94" graph is a simple 94-node modification (Ejov) which has only "a few" Hamiltonian cycles: (exactly?) 76, 800.
- This sounds like a lot, but there are 3^{94} possible sub-graphs. So, finding a Hamiltonian cycle "by chance", as percentage, is

$$\frac{76,800}{3^{94}} \times 100 = 1.08633 \times 10^{-38},$$

not terribly likely, is it?
- Like looking for a proverbial "needle in a haystack"...

- We consider a moving object tracing out a directed path on the graph Γ with its movement "controlled" by a function

$$f : \mathcal{V} = \mathcal{V}(\Gamma) = \{1, 2 \ldots, N\} \to \mathcal{A} = \mathcal{A}(\Gamma)$$

mapping the set of nodes of Γ into its set of arcs.

- We think of $\mathcal{V}$ as the state space of a controlled Markov chain, where for each state/node i, the action space

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

is in $1:1$ correspondence with the set of arcs emanating from that node or, equivalently, with the set of endpoints ("heads") of those arcs.

Stochastic, Dynamic, Embedding of the HCP

Example (1)

- Consider the complete graph Γ_4 on four nodes (with no self-loops). In a natural way, the Hamiltonian cycle

$$c_1 \ : \ 1 \to 2 \to 3 \to 4 \to 1$$

corresponds to the "deterministic (stationary) control"

$$\mathbf{f}_1 \ : \ \{1, 2, 3, 4\} \to \{2, 3, 4, 1\},$$

where $\mathbf{f}_1(2) = 3 \Leftrightarrow$ to the controller choosing arc $(2,3)$ in state 2 with probability 1.

- A Markov chain induced by $\mathbf{f}_1$ is given by the " zero-one" transition matrix

$$P(\mathbf{f}_1) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix},$$

an irreducible, stochastic matrix.

Example (1, cont.)

- However, the union of two sub-cycles:

$$1 \to 2 \to 1 \text{ and } 3 \to 4 \to 3$$

corresponds to the deterministic control

$$\mathbf{f}_2 \; : \; \{1, 2, 3, 4\} \to \{2, 1, 4, 3\}$$

which identifies the Markov chain transition matrix

$$P(\mathbf{f}_2) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

containing two distinct ergodic classes.

Performance of a Policy

- This leads to a natural embedding of the HCP in a controlled Markov chain.

Performance of a Policy

- This leads to a natural embedding of the HCP in a controlled Markov chain.
- Typically, the performance of a policy $\mathbf{f}$ is evaluated either by
  - a limiting average criterion

$$\mathbf{v}(\mathbf{f}) = \left[\lim_{T \to \infty} \frac{1}{1+T} \sum_{t=0}^{T} P(\mathbf{f})^t \right] \mathbf{r}(\mathbf{f}) = P(\mathbf{f})^* \mathbf{r}(\mathbf{f}), \text{ or}$$

Performance of a Policy

- This leads to a natural embedding of the HCP in a controlled Markov chain.
- Typically, the performance of a policy $\mathbf{f}$ is evaluated either by
  - a limiting average criterion

$$\mathbf{v}(\mathbf{f}) = \left[\lim_{T \to \infty} \frac{1}{1+T} \sum_{t=0}^{T} P(\mathbf{f})^t \right] \mathbf{r}(\mathbf{f}) = P(\mathbf{f})^* \mathbf{r}(\mathbf{f}), \text{ or}$$

  - a discounted criterion (for a "discount factor" $\beta \in [0,1)$)

$$\mathbf{v}_\beta(\mathbf{f}) = \sum_{t=0}^{\infty} \beta^t P(\mathbf{f})^t \mathbf{r}(\mathbf{f}) = [I - \beta P(\mathbf{f})]^{-1} \mathbf{r}(\mathbf{f}).$$

Performance of a Policy

- This leads to a natural embedding of the HCP in a controlled Markov chain.
- Typically, the performance of a policy $\mathbf{f}$ is evaluated either by
  - a limiting average criterion

$$\mathbf{v}(\mathbf{f}) = \left[\lim_{T \to \infty} \frac{1}{1+T} \sum_{t=0}^{T} P(\mathbf{f})^t \right] \mathbf{r}(\mathbf{f}) = P(\mathbf{f})^* \mathbf{r}(\mathbf{f}), \text{ or}$$

  - a discounted criterion (for a "discount factor" $\beta \in [0,1)$)

$$\mathbf{v}_\beta(\mathbf{f}) = \sum_{t=0}^{\infty} \beta^t P(\mathbf{f})^t \mathbf{r}(\mathbf{f}) = [I - \beta P(\mathbf{f})]^{-1} \mathbf{r}(\mathbf{f}).$$

- The immediate reward/cost vector $\mathbf{r}(\mathbf{f})$ will be set to

$$\mathbf{r}(\mathbf{f}) := (1, 0, \dots, 0)^T \text{ for all } \mathbf{f}.$$

Difficulties

- Such a controlled Markov chain (CMC, for short) has a multi-chain ergodic structure which complicates the analysis.
- This is essential because multiple ergodic classes correspond to sub-cycles in the original graph that are the "mother of all difficulties" in the TSP (and hence also in the HCP).

Difficulties

- Such a controlled Markov chain (CMC, for short) has a multi-chain ergodic structure which complicates the analysis.
- This is essential because multiple ergodic classes correspond to sub-cycles in the original graph that are the "mother of all difficulties" in the TSP (and hence also in the HCP).
- However, from Markov chains' perspective, there is an opportunity to parametrically differentiate between controls inducing the multi-chain and the uni-chain ergodic structures.

Difficulties

- Such a controlled Markov chain (CMC, for short) has a multi-chain ergodic structure which complicates the analysis.
- This is essential because multiple ergodic classes correspond to sub-cycles in the original graph that are the "mother of all difficulties" in the TSP (and hence also in the HCP).
- However, from Markov chains' perspective, there is an opportunity to parametrically differentiate between controls inducing the multi-chain and the uni-chain ergodic structures.
- Indeed, we have at least two ways of achieving such a differentiation.

Method 1: Perturbation

- Multiple chains are "disguised" with the help of a singular perturbation. For instance, we can replace $P(\mathbf{f}_2)$ with

$$P_\varepsilon(\mathbf{f}_2) = \begin{pmatrix} \varepsilon & 1-3\varepsilon & \varepsilon & \varepsilon \\ 1-3\varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1-3\varepsilon \\ \varepsilon & \varepsilon & 1-3\varepsilon & \varepsilon \end{pmatrix}.$$

Method 1: Perturbation

- Multiple chains are "disguised" with the help of a singular perturbation. For instance, we can replace $P(\mathbf{f}_2)$ with

$$P_\varepsilon(\mathbf{f}_2) = \begin{pmatrix} \varepsilon & 1-3\varepsilon & \varepsilon & \varepsilon \\ 1-3\varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1-3\varepsilon \\ \varepsilon & \varepsilon & 1-3\varepsilon & \varepsilon \end{pmatrix}.$$

- This perturbation is singular as it alters the ergodic structure by changing the Markov chain $P(\mathbf{f}_2)$ to one that is irreducible. Its stationary distribution matrix is

$$P^*_\varepsilon(\mathbf{f}_2) = \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix},$$

where $\pi^*_\varepsilon(\mathbf{f}_2) = (0.25, 0.25, 0.25, 0.25)$ is the unique invariant measure of $P_\varepsilon(\mathbf{f}_2)$.

Method 2: Discount

- Discount factor β is used to differentiate between the multi- and uni-chain cases.
- For instance,
  - for Hamiltonian cycle policy $\mathbf{f}_1$

$$[\mathbf{v}_\beta(\mathbf{f}_1)]_1 = 1 + \beta^4 + \beta^8 + \ldots... = \frac{1}{1-\beta^4},$$

Method 2: Discount

- Discount factor β is used to differentiate between the multi- and uni-chain cases.
- For instance,
  - for Hamiltonian cycle policy $\mathbf{f}_1$
  $$[\mathbf{v}_\beta(\mathbf{f}_1)]_1 = 1 + \beta^4 + \beta^8 + \ldots\ldots = \frac{1}{1-\beta^4},$$
  - for the 2-cycle policy $\mathbf{f}_2$
  $$[\mathbf{v}_\beta(\mathbf{f}_2)]_1 = 1 + \beta^2 + \beta^4 + \ldots\ldots = \frac{1}{1-\beta^2}.$$

Method 2: Discount

- Discount factor β is used to differentiate between the multi- and uni-chain cases.
- For instance,
  - for Hamiltonian cycle policy $\mathbf{f}_1$

$$[\mathbf{v}_\beta(\mathbf{f}_1)]_1 = 1 + \beta^4 + \beta^8 + \ldots\ldots = \frac{1}{1-\beta^4},$$

  - for the 2-cycle policy $\mathbf{f}_2$

$$[\mathbf{v}_\beta(\mathbf{f}_2)]_1 = 1 + \beta^2 + \beta^4 + \ldots\ldots = \frac{1}{1-\beta^2}.$$

- Further, for a noose cycle control

$$\mathbf{f}_3 \ : \ \{1,2,3,4\} \to \{2,3,4,3\},$$

$$[\mathbf{v}_\beta(\mathbf{f}_3)]_1 = 1 + 0 + 0 + \ldots\ldots = 1.$$

Method 2: Discount

- Discount factor β is used to differentiate between the multi- and uni-chain cases.
- For instance,
  - for Hamiltonian cycle policy $\mathbf{f}_1$
  $$[\mathbf{v}_\beta(\mathbf{f}_1)]_1 = 1 + \beta^4 + \beta^8 + \ldots\ldots = \frac{1}{1-\beta^4},$$
  - for the 2-cycle policy $\mathbf{f}_2$
  $$[\mathbf{v}_\beta(\mathbf{f}_2)]_1 = 1 + \beta^2 + \beta^4 + \ldots\ldots = \frac{1}{1-\beta^2}.$$
- Further, for a noose cycle control
$$\mathbf{f}_3 \ : \ \{1,2,3,4\} \rightarrow \{2,3,4,3\},$$
$$[\mathbf{v}_\beta(\mathbf{f}_3)]_1 = 1 + 0 + 0 + \ldots\ldots = 1.$$
- Since $\beta \in [0,1)$
$$\frac{1}{1-\beta^2} > \frac{1}{1-\beta^4} > 1.$$

Randomization

- Next, consider a Markov chain induced by a (stationary) randomized control.
- The latter is an $N \times N$ matrix $\mathbf{f}$ with entries representing probabilities $f(i, a)$ of choosing a possible arc a at state i whenever this state is visited; with $f(i, a) = 0$, whenever $a \notin \mathcal{A}(i)$.
- Our control space $\mathcal{F}_S = \{\text{randomized controls}\}$.

Randomization

- Next, consider a Markov chain induced by a (stationary) randomized control.
- The latter is an $N \times N$ matrix $\mathbf{f}$ with entries representing probabilities $f(i, a)$ of choosing a possible arc a at state i whenever this state is visited; with $f(i, a) = 0$, whenever $a \notin \mathcal{A}(i)$.
- Our control space $\mathcal{F}_S = \{\text{randomized controls}\}$.
- Hitherto, we considered only special paths which our moving object can trace out in the graph Γ. These corresponded to the subspace $\mathcal{F}_D \subset \mathcal{F}_S$ of deterministic controls.

Randomization

- Next, consider a Markov chain induced by a (stationary) randomized control.
- The latter is an $N \times N$ matrix $\mathbf{f}$ with entries representing probabilities $f(i, a)$ of choosing a possible arc a at state i whenever this state is visited; with $f(i, a) = 0$, whenever $a \notin \mathcal{A}(i)$.
- Our control space $\mathcal{F}_S = \{\text{randomized controls}\}$.
- Hitherto, we considered only special paths which our moving object can trace out in the graph Γ. These corresponded to the subspace $\mathcal{F}_D \subset \mathcal{F}_S$ of deterministic controls.
- Passing from $\mathcal{F}_D$ to $\mathcal{F}_S$ constitutes a "continuous relaxation" of the HCP.

Example (1, cont.)

- Consider the case where $\mathbf{f}_\lambda$ is obtained from the 2-cycle deterministic control $\mathbf{f}_2$ by the "controller" randomizing at node 4 as follows. He/she chooses:
    - arc $(4, 2)$ with probability $f(4, 2) = \lambda$, and
    - arc $(4, 3)$ with probability $f(4, 3) = (1 - \lambda)$.
- The resulting policy $\mathbf{f}_\lambda$ induces a MC

$$P(\mathbf{f}_\lambda) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \lambda & (1-\lambda) & 0 \end{pmatrix}.$$

Note that...

- The following phrases and terms are used interchangeably:
  control & policy,
  controlled Markov chain & Markov Decision Process (MDP).

Note that...

- The following phrases and terms are used interchangeably:
  control & policy,
  controlled Markov chain & Markov Decision Process (MDP).
- The space of objects where we search for a solution has been changed from

A discrete set of sub-graphs of the given graph Γ

to

Note that...

- The following phrases and terms are used interchangeably:
  control & policy,
  controlled Markov chain & Markov Decision Process (MDP).
- The space of objects where we search for a solution has been changed from

A discrete set of sub-graphs of the given graph Γ

to

A convex control/policy space of a MDP, or
A convex frequency space of such a process.

Optimization in the Frequency Space

- The discounted frequency space is the set

$$X_\beta := \{\mathbf{x}(\mathbf{f}) | \mathbf{f} \in \mathcal{F}_S\}$$

of vectors $\mathbf{x}(\mathbf{f})$ whose entries are defined by

$$x_{ia}(\mathbf{f}) := \sum_{t=0}^{\infty} \beta^t Pr(X_t = i, A_t = a | \mathbf{f}).$$

Optimization in the Frequency Space

- The discounted frequency space is the set
$$X_\beta := \{\mathbf{x}(\mathbf{f}) | \mathbf{f} \in \mathcal{F}_S\}$$
of vectors $\mathbf{x}(\mathbf{f})$ whose entries are defined by
$$x_{ia}(\mathbf{f}) := \sum_{t=0}^{\infty} \beta^t Pr(X_t = i, A_t = a | \mathbf{f}).$$
- The construction of $\mathbf{x}$ defines a map M of the policy space $\mathcal{F}_S$ into $\mathbb{R}^m$ by
$$M(\mathbf{f}) := \mathbf{x}(\mathbf{f}).$$

Optimization in the Frequency Space

- The discounted frequency space is the set

$$X_\beta := \{\mathbf{x}(\mathbf{f}) | \mathbf{f} \in \mathcal{F}_S\}$$

of vectors $\mathbf{x}(\mathbf{f})$ whose entries are defined by

$$x_{ia}(\mathbf{f}) := \sum_{t=0}^{\infty} \beta^t Pr(X_t = i, A_t = a | \mathbf{f}).$$

- The construction of $\mathbf{x}$ defines a map M of the policy space $\mathcal{F}_S$ into $\mathbb{R}^m$ by

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

- The map M is invertible and its inverse M^{-1} is defined by

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

- The above discounted frequency space X_β is actually a linear polytope characterized by the transportation-like constraints:

$$\sum_{i=1}^{N}\sum_{a=1}^{N}(\delta_{ij} - \beta p(j|i,a))x_{ia} = \delta_{1j},\ \forall j = 1, \ldots, N, \quad (1)$$

$$x_{ia} \geq 0,\ \forall\ i, a. \quad (2)$$

- The above discounted frequency space X_β is actually a linear polytope characterized by the transportation-like constraints:

$$\sum_{i=1}^{N}\sum_{a=1}^{N}(\delta_{ij} - \beta p(j|i,a))x_{ia} = \delta_{1j},\ \forall j = 1,\dots,N, \quad (1)$$

$$x_{ia} \geq 0,\ \forall\ i, a. \quad (2)$$

Theorem (Feinberg, 2000)

At a control $\mathbf{f}$ *that is a Hamiltonian cycle,* $\mathbf{x} = \mathbf{x}(\mathbf{f})$ *satisfies*

$$\sum_{a \in A(1)} x_{1a} = \frac{1}{1-\beta^N}. \quad (3)$$

- The above discounted frequency space X_β is actually a linear polytope characterized by the transportation-like constraints:

$$\sum_{i=1}^{N}\sum_{a=1}^{N}(\delta_{ij} - \beta p(j|i,a))x_{ia} = \delta_{1j}, \forall j = 1, \ldots, N, \quad (1)$$

$$x_{ia} \geq 0, \forall\, i, a. \quad (2)$$

Theorem (Feinberg, 2000)

At a control $\mathbf{f}$ *that is a Hamiltonian cycle,* $\mathbf{x} = \mathbf{x}(\mathbf{f})$ *satisfies*

$$\sum_{a \in A(1)} x_{1a} = \frac{1}{1 - \beta^N}. \quad (3)$$

- Now, let

$$W := \{\mathbf{x} | (1) - (3) \text{ hold}\} = \{\mathbf{x} | A(\beta)\mathbf{x} = \mathbf{b};\ \mathbf{x} \geq \mathbf{0}\}.$$

Theorem (Chen & Filar, 1992; Feinberg, 2000)

Let $\mathbf{x} \in W$ *be such that for all* i *and* a*:*

$$M^{-1}(\mathbf{x})[i,a] = \mathbf{f}_x(i,a) := \frac{x_{ia}}{\sum_a x_{ia}} \in \{0,1\}.$$

Then $\mathbf{f}_x$ *traces out a Hamiltonian cycle in* Γ*.*

Theorem (Chen & Filar, 1992; Feinberg, 2000)

Let $\mathbf{x} \in W$ be such that for all i and a:

$$M^{-1}(\mathbf{x})[i,a] = \mathbf{f}_x(i,a) := \frac{x_{ia}}{\sum_a x_{ia}} \in \{0,1\}.$$

Then $\mathbf{f}_x$ traces out a Hamiltonian cycle in Γ.

- So, HCP is equivalent to feasibility problem: Find $\mathbf{x}$ satisfying:

$$\sum_{i=1}^{N}\sum_{a=1}^{N}(\delta_{ij} - \beta p(j|i,a))x_{ia} = \delta_{1j},\ j = 1,\dots,N, \tag{4}$$

$$x_{ia} \geq 0 \ \forall\ i,a, \tag{5}$$

$$\sum_{a \in A(1)} x_{1a} = \frac{1}{1-\beta^N}, \tag{6}$$

Theorem (Chen & Filar, 1992; Feinberg, 2000)

Let $\mathbf{x} \in W$ *be such that for all* i *and* a*:*

$$M^{-1}(\mathbf{x})[i,a] = \mathbf{f}_x(i,a) := \frac{x_{ia}}{\sum_a x_{ia}} \in \{0,1\}.$$

Then $\mathbf{f}_x$ *traces out a Hamiltonian cycle in* Γ*.*

- So, HCP is equivalent to feasibility problem: Find $\mathbf{x}$ satisfying:

$$\sum_{i=1}^{N}\sum_{a=1}^{N}(\delta_{ij} - \beta p(j|i,a))x_{ia} = \delta_{1j},\ j = 1,\ldots,N, \quad (4)$$

$$x_{ia} \geq 0 \quad \forall\ i, a, \quad (5)$$

$$\sum_{a \in A(1)} x_{1a} = \frac{1}{1-\beta^N}, \quad (6)$$

$$(\text{\& sadly})\ \frac{x_{ia}}{\sum_{a \in A(i)} x_{ia}} \in \{0,1\}\ \forall\ i, a. \quad (7)$$

- Meanwhile, in 1998, Jean Lasserre (CNRS) and I considered a version of the original question:

Does $\exists\ \mathbf{x} \in W$ such that $M^{-1}(\mathbf{x}) = \mathbf{f}_{\mathbf{x}}$ is a HC?

- Or, equivalently, does $\exists\ \mathbf{x}$ such that

$$A(\beta)\mathbf{x} = \mathbf{b};\ \mathbf{x} \geq \mathbf{0}\ \ \&\ \ \mathbf{f}_{\mathbf{x}} \in \mathcal{F}_D?$$

- Meanwhile, in 1998, Jean Lasserre (CNRS) and I considered a version of the original question:

Does $\exists\ \mathbf{x} \in W$ such that $M^{-1}(\mathbf{x}) = \mathbf{f}_{\mathbf{x}}$ is a HC?

- Or, equivalently, does $\exists\ \mathbf{x}$ such that

$$A(\beta)\mathbf{x} = \mathbf{b};\ \mathbf{x} \geq \mathbf{0}\ \ \&\ \ \mathbf{f}_{\mathbf{x}} \in \mathcal{F}_D?$$

- Recall,

$$\mathbf{f}_{\mathbf{x}} \in \mathcal{F}_D \iff \frac{x_{ia}}{\sum\limits_{a} x_{ia}} \in \{0, 1\}\ \ \forall i, a.$$

- Meanwhile, in 1998, Jean Lasserre (CNRS) and I considered a version of the original question:

Does $\exists\ \mathbf{x} \in W$ such that $M^{-1}(\mathbf{x}) = \mathbf{f_x}$ is a HC?

- Or, equivalently, does $\exists\ \mathbf{x}$ such that

$$A(\beta)\mathbf{x} = \mathbf{b};\ \mathbf{x} \geq \mathbf{0}\ \ \&\ \ \mathbf{f_x} \in \mathcal{F}_D?$$

- Recall,

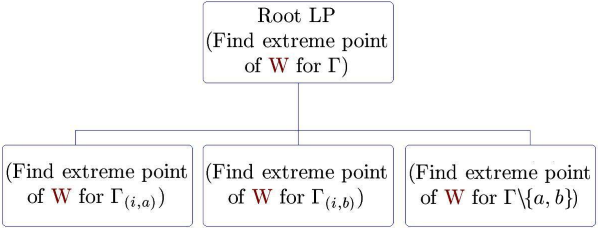
$$\mathbf{f_x} \in \mathcal{F}_D \iff \frac{x_{ia}}{\sum_a x_{ia}} \in \{0, 1\}\ \ \forall i, a.$$

- We observed that if $\mathbf{x}$ is an extreme point of W, then either
  - $\mathbf{f_x} \in \mathcal{F}_D$ (and we're done), or
  - $\exists!$ node i and a pair of arcs $(i, a), (i, b)$ such that

$$\mathbf{f_x}(i, a)\ \&\ \mathbf{f_x}(i, b) > 0 \quad \Leftrightarrow \quad x_{ia}\ \&\ x_{ib} > 0.$$

Branch & Fix Algorithm

- Hence, we proposed - but did not implement - a "Branch & Fix" algorithm with the logical structure:



- Note that at each sub-node of this B& F tree we are solving the feasibility problem for a smaller graph.

Branch & Fix Algorithm

- However, we were suspicious that the B & F tree would grow so fast that the method would be essentially a brute force enumeration of all subgraphs of Γ (all deterministic controls).
- Only in 2006 I asked Giang Nguyen and Michael Haythorpe to try it out.

Branch & Fix Algorithm

- However, we were suspicious that the B & F tree would grow so fast that the method would be essentially a brute force enumeration of all subgraphs of Γ (all deterministic controls).
- Only in 2006 I asked Giang Nguyen and Michael Haythorpe to try it out.
- The preliminary results were surprisingly good and we're beginning to understand why!

Graph	Branches	Time
Hamiltonian (24 N 72 A)	394	0:04
Dodecahedron (20 N 60 A)	65	0:01
Chess8 (64 N 336A)	1097	0:38
NH: Petersen (10 N 30 A)	154	0:01
NH: Coxeter (28 N 84 A)	41268	8:23

- Even $41,268 \ll 3^{28} = 22,876,792,454,961$; right?!

- At each sub-problem in B & F, we find an extreme point of $W = \{\mathbf{x} | A(\beta)\mathbf{x} = \mathbf{b};\ \mathbf{x} \geq \mathbf{0}\}$ for that sub-problem and check if:

$$\mathbf{f_x} \in \mathcal{F}_D \iff \frac{x_{ia}}{\sum_a x_{ia}} \in \{0, 1\} \quad \forall i, a \ ?$$

    - If answer YES, $\mathbf{f_x}$ is a HC.
    - What if answer is NO?

- At each sub-problem in B & F, we find an extreme point of $W = \{\mathbf{x} | A(\beta)\mathbf{x} = \mathbf{b};\ \mathbf{x} \geq \mathbf{0}\}$ for that sub-problem and check if:

$$\mathbf{f_x} \in \mathcal{F}_D \iff \frac{x_{ia}}{\sum_a x_{ia}} \in \{0, 1\} \ \ \forall i, a \ \ ?$$

  - If answer YES, $\mathbf{f_x}$ is a HC.
  - What if answer is NO?

- Well, $\mathbf{f_x} \in \mathcal{F}_S$ and $\mathbf{x}$ satisfies:

$$\sum_{a \in \mathcal{A}(1)} x_{1a} = \frac{1}{1 - \beta^N}. \tag{8}$$

- At each sub-problem in B & F, we find an extreme point of $W = \{\mathbf{x} | A(\beta)\mathbf{x} = \mathbf{b};\ \mathbf{x} \geq \mathbf{0}\}$ for that sub-problem and check if:

$$\mathbf{f}_{\mathbf{x}} \in \mathcal{F}_D \iff \frac{x_{ia}}{\sum_a x_{ia}} \in \{0,1\} \quad \forall i, a \ \ ?$$

  - If answer YES, $\mathbf{f}_{\mathbf{x}}$ is a HC.
  - What if answer is NO?

- Well, $\mathbf{f}_{\mathbf{x}} \in \mathcal{F}_S$ and $\mathbf{x}$ satisfies:

$$\sum_{a \in \mathcal{A}(1)} x_{1a} = \frac{1}{1-\beta^N}. \tag{8}$$

- Recall the inequality (for $\beta \in [0,1)$ and $k < N$)

$$\frac{1}{1-\beta^k} > \frac{1}{1-\beta^N} > 1.$$

- At each sub-problem in B & F, we find an extreme point of $W = \{\mathbf{x} | A(\beta)\mathbf{x} = \mathbf{b};\ \mathbf{x} \geq \mathbf{0}\}$ for that sub-problem and check if:

$$\mathbf{f}_{\mathbf{x}} \in \mathcal{F}_D \iff \frac{x_{ia}}{\sum_a x_{ia}} \in \{0, 1\} \quad \forall i, a \quad ?$$

  - If answer YES, $\mathbf{f}_{\mathbf{x}}$ is a HC.
  - What if answer is NO?

- Well, $\mathbf{f}_{\mathbf{x}} \in \mathcal{F}_S$ and $\mathbf{x}$ satisfies:

$$\sum_{a \in \mathcal{A}(1)} x_{1a} = \frac{1}{1 - \beta^N}. \tag{8}$$

- Recall the inequality (for $\beta \in [0, 1)$ and $k < N$)

$$\frac{1}{1 - \beta^k} > \frac{1}{1 - \beta^N} > 1.$$

- How does $\mathbf{f}_{\mathbf{x}} \in \mathcal{F}_S$ "fool us" and manage to satisfy (8)?

Proposition (Nguyen, 2007)

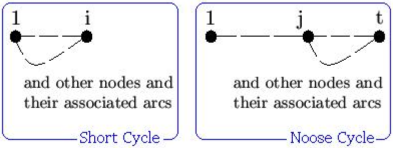
If an extreme point $\mathbf{x}$ of W induces a randomized control

$$\mathbf{f_x} \in \mathcal{F}_S \text{ and } \sum_{a \in \mathcal{A}(1)} x_{1a} = \frac{1}{1-\beta^N} \text{ holds,}$$

then $\exists$ a short-cycle control $\mathbf{f}_2$ and a noose-cycle policy $\mathbf{f}_3$ and $\lambda \in (0,1)$ such that

$$\mathbf{f_x} = \lambda \mathbf{f}_2 + (1-\lambda)\mathbf{f}_3.$$

Also, $\mathbf{f}_2$ & $\mathbf{f}_3$ coincide except at one node.



- So, perhaps, it's not so easy for $\mathbf{f_x}$ to "pretend" to be a HC......Certainly, it cannot be done by mixing just any two deterministic controls.....

A Faster Heuristic

- Here the cut frequency space is constrained further

$$
\begin{aligned}
\sum_{i=1}^{N} \sum_{a \in A(i)} (\delta_{ij} - \beta p(j|i,a))\, x_{ia} &= \delta_{1j}(1-\beta^N), \ \forall\, j, \\
\sum_{a \in A(1)} x_{1a} &= 1, \\
\text{box-constraints } \beta^{N-1} \le \sum_{a \in A(i)} x_{ia} &\le \beta, \ \forall\ i \neq 1, \\
x_{ia} &\ge 0, \ \forall\, i, a.
\end{aligned}
$$

- The new x_{ia}'s are the old x_{ia}'s multiplied by $(1-\beta^N)$.
- The box-constraints can be narrowed by increasing $\beta \uparrow 1$.

A Faster Heuristic

- Here the cut frequency space is constrained further

$$\begin{aligned}
\sum_{i=1}^{N} \sum_{a \in A(i)} (\delta_{ij} - \beta p(j|i,a))\, x_{ia} &= \delta_{1j}(1-\beta^N), \ \ \forall\, j, \\
\sum_{a \in A(1)} x_{1a} &= 1, \\
\text{box-constraints } \beta^{N-1} \le \sum_{a \in A(i)} x_{ia} &\le \beta, \ \ \forall \ i \ne 1, \\
x_{ia} &\ge 0, \ \ \forall\, i, a.
\end{aligned}$$

- The new x_{ia}'s are the old x_{ia}'s multiplied by $(1-\beta^N)$.
- The box-constraints can be narrowed by increasing $\beta \uparrow 1$.
- We still need

$$x_{ia} x_{ib} = 0 \quad \text{for all } i, \ \ a \ne b.$$

A Faster Heuristic

- A recent heuristic simply searches for a feasible solution of the above using the simplex-based OPL-CPLEX.
- Preliminary results were very encouraging.

A Faster Heuristic

- A recent heuristic simply searches for a feasible solution of the above using the simplex-based OPL-CPLEX.
- Preliminary results were very encouraging.
- In particular, two well known "difficult" graphs, listed on the University of Heidelberg's web site successfully solved.

Graph	Time
8x8 Knight's Tour Problem (64 nodes, 336 arcs)	1 sec
Perturbed Horton Graph (94 nodes, 282 arcs)	2 sec
20x20 Knight's Tour Problem (400 nodes, 2736 arcs)	11 min
1000-node Heidelberg Graph (1000 nodes, 3996 arcs)	24 min
2000-node Heidelberg Graph (2000 nodes, 7992 arcs)	46 hrs

- A choice of β strongly influences the numerical performance.

Optimization in the Policy Space

- Finally, why not get rid of noose-cycle controls since they permit randomized controls to pretend to be a HC?

- Finally, why not get rid of noose-cycle controls since they permit randomized controls to pretend to be a HC?
- This immediately leads to the observation that, perhaps,

> the correct convex domain where HC's should be sought, is the set $\mathcal{DS}$ of doubly stochastic matrices induced by a given graph.

- Finally, why not get rid of noose-cycle controls since they permit randomized controls to pretend to be a HC?
- This immediately leads to the observation that, perhaps,

the correct convex domain where HC's should be sought, is the set $\mathcal{DS}$ of doubly stochastic matrices induced by a given graph.

- A square non-negative matrix is doubly stochastic if both its row-sums and column-sums are equal to 1.

Optimization in the Policy Space

- Finally, why not get rid of noose-cycle controls since they permit randomized controls to pretend to be a HC?
- This immediately leads to the observation that, perhaps,

the correct convex domain where HC's should be sought, is the set $\mathcal{DS}$ of doubly stochastic matrices induced by a given graph.

- A square non-negative matrix is doubly stochastic if both its row-sums and column-sums are equal to 1.

Theorem (Birkhoff-von Neumann)

The set of all $N \times N$ doubly stochastic matrices is the convex hull of permutation matrices.

Theorem (Borkar, Ejov & Filar, 2004)

Given a graph and the $\varepsilon-$perturbed Markov Decision Process, let $\mathcal{DS}$ be the space of doubly stochastic controls. Define the r.v.

$$\tau_1 := \textit{ first hitting time of the home node } 1.$$

The Hamiltonian Cycle Problem reduces to "merely" minimizing the variance of τ_1, namely,

$$\min_{\mathbf{f} \in \mathcal{DS}} E_1 \left[(\tau_1 - N)^2 | \mathbf{f} \right].$$

- This is an, interesting, optimization problem that we are tackling, with Walter Murray's help, by interior point methods.
- Could (should?) it be also tackled by cross-entropy methods?

Connection with Cross-Entropy Methods

In Rubinstein and Kroese 2004 a cross-entropy algorithm is outlined to solve the TSP.

Connection with Cross-Entropy Methods

In Rubinstein and Kroese 2004 a cross-entropy algorithm is outlined to solve the TSP.
It associates transition probabilities to all arcs and updates these probabilities on the basis of information contained in samples of tours.

Connection with Cross-Entropy Methods

In Rubinstein and Kroese 2004 a cross-entropy algorithm is outlined to solve the TSP.
It associates transition probabilities to all arcs and updates these probabilities on the basis of information contained in samples of tours.

Step 1.

- Choose an initial transition probability matrix P_0, say with elements uniformly distributed at each row.

Connection with Cross-Entropy Methods

In Rubinstein and Kroese 2004 a cross-entropy algorithm is outlined to solve the TSP.
It associates transition probabilities to all arcs and updates these probabilities on the basis of information contained in samples of tours.
Step 1.

- Choose an initial transition probability matrix P_0, say with elements uniformly distributed at each row.
- Generate n tours $\tau_1, \tau_2, \ldots, \tau_n$ from P_0 with lengths $L(\tau_1), L(\tau_2), \ldots, L(\tau_n)$ and find:

Connection with Cross-Entropy Methods

In Rubinstein and Kroese 2004 a cross-entropy algorithm is outlined to solve the TSP.
It associates transition probabilities to all arcs and updates these probabilities on the basis of information contained in samples of tours.
Step 1.

- Choose an initial transition probability matrix P_0, say with elements uniformly distributed at each row.
- Generate n tours $\tau_1, \tau_2, \ldots, \tau_n$ from P_0 with lengths $L(\tau_1), L(\tau_2), \ldots, L(\tau_n)$ and find:

$$\min \left\{ \gamma : \ \frac{1}{n} \sum_{j=1}^{n} \exp^{\frac{-L(\tau_j)}{\gamma}} \geq \rho \right\},$$

for a fixed ρ. Denote the initial solution by γ_1^*.

Step 2.

- Use the same n tours $\tau_1, \tau_2, \ldots, \tau_n$ associated with P_0.
- Calculate P_1^* and P_1 by applying the following equations:

$$
\begin{aligned}
p_{1,ia}^* &= \frac{\sum\limits_{1 \le j \le n : \{ia\} \in \tau_j} e^{\frac{-L(\tau_j)}{\gamma_1^*}}}{\sum_{j=1}^{n} e^{\frac{-L(\tau_j)}{\gamma_1^*}}}, \\
P_1 &= (1-\alpha) P_0 + \alpha P_1^*,
\end{aligned}
$$

where α is the smoothing parameter chosen from $(0, 1)$.

Step 3.

- Generate n new tours $\tau_1, \tau_2, \ldots, \tau_n$ from P_1.
- Repeat Step 1 and Step 2 again with P_0 and P_1^* replaced with P_1 and P_2^*, respectively.
- Denote the final solution by P_2 and the corresponding solution at stage t by P_t.

CE Algorithm

Step 3.

- Generate n new tours $\tau_1, \tau_2, \ldots, \tau_n$ from P_1.
- Repeat Step 1 and Step 2 again with P_0 and P_1^* replaced with P_1 and P_2^*, respectively.
- Denote the final solution by P_2 and the corresponding solution at stage t by P_t.

Step 4.

- Let $\mathcal{L}_t$ denote the length of the shortest tour up to stage t.
- If for any $t > r$ and some r, say $r = 5$,

$$\mathcal{L}_t = \mathcal{L}_{t-1} = \ldots = \mathcal{L}_{t-r},$$

then repeat Step 3 and STOP, else repeat Steps 2 - 4 again.

CE Algorithm

- Convergence: It is shown in Margolin 2005 and Costa et al 2007 that similar Cross-Entropy algorithms converge to an optimum solution in finite number of steps with probability one.

CE Algorithm

- Convergence: It is shown in Margolin 2005 and Costa et al 2007 that similar Cross-Entropy algorithms converge to an optimum solution in finite number of steps with probability one.

- However, it is possible that the stopping criterion of Step 4 may require many iterations to satisfy.

CE Algorithm

- Convergence: It is shown in Margolin 2005 and Costa et al 2007 that similar Cross-Entropy algorithms converge to an optimum solution in finite number of steps with probability one.
- However, it is possible that the stopping criterion of Step 4 may require many iterations to satisfy.
- Is it possible that some insights from preceding results could be used in conjunction with this CE algorithm?

Nexus Between HCP and CE

Since - for the HCP - not all arcs are available in a given graph, we "fill-it-in" with artificial arcs that have sufficiently high costs that they will be identifiable.

Nexus Between HCP and CE

Since - for the HCP - not all arcs are available in a given graph, we "fill-it-in" with artificial arcs that have sufficiently high costs that they will be identifiable.

So, we solve a "TSP" and claim that the graph is (very likely) non-Hamiltonian, if the minimal tour is "too long".

Step 0: Initiation.

- Insert artificial arcs to make the given graph a complete one with no self-loops.

Step 0: Initiation.

- Insert artificial arcs to make the given graph a complete one with no self-loops.
- Assign a length to each arc as follow:
$$\begin{cases} l_{ia} \sim U(0, \omega), & \text{if } (i, a) \text{ is an authentic arc,} \\ l_{ia} \sim U(\mu, \mu + \sigma), & \text{otherwise.} \end{cases}$$
Here, $\omega, \mu,$ and σ are arbitrary positive real numbers and the only restriction is $\omega \ll \mu$.

Step 0: Initiation.

- Insert artificial arcs to make the given graph a complete one with no self-loops.
- Assign a length to each arc as follow:
$$\begin{cases} l_{ia} \sim U(0, \omega), & \text{if } (i, a) \text{ is an authentic arc,} \\ l_{ia} \sim U(\mu, \mu + \sigma), & \text{otherwise.} \end{cases}$$
Here, $\omega, \mu,$ and σ are arbitrary positive real numbers and the only restriction is $\omega \ll \mu$.
- Since all the lengths are generated from some continuous distributions, the total length $L(\tau)$ of a typical tour τ is a continuous random variable.

- Hence, for two distinct tours τ_i and τ_j, we will have
$$\Pr(L(\tau_i) = L(\tau_j)) = 0.$$

- Hence, for two distinct tours τ_i and τ_j, we will have
$$\Pr(L(\tau_i) = L(\tau_j)) = 0.$$
- This means that the shortest tour is unique and (if the graph is Hamiltonian), then its length is less than $N\omega$.

- Hence, for two distinct tours τ_i and τ_j, we will have
$$\Pr(L(\tau_i) = L(\tau_j)) = 0.$$
- This means that the shortest tour is unique and (if the graph is Hamiltonian), then its length is less than $N\omega$.
- Set up the initial transition probability matrix as below:

$$p_{0,ia} := \begin{cases} \frac{1}{|A(i)|} - \frac{N-1-|A(i)|}{|A(i)|}\varepsilon, & \text{if } (i,a) \text{ is an authentic arc,} \\ \varepsilon, & \text{otherwise.} \end{cases}$$

Here, $\varepsilon > 0$ and is sufficiently small.

- Hence, for two distinct tours τ_i and τ_j, we will have
$$\Pr(L(\tau_i) = L(\tau_j)) = 0.$$
- This means that the shortest tour is unique and (if the graph is Hamiltonian), then its length is less than $N\omega$.
- Set up the initial transition probability matrix as below:

$$p_{0,ia} := \begin{cases} \frac{1}{|A(i)|} - \frac{N-1-|A(i)|}{|A(i)|}\varepsilon, & \text{if } (i,a) \text{ is an authentic arc,} \\ \varepsilon, & \text{otherwise.} \end{cases}$$

Here, $\varepsilon > 0$ and is sufficiently small.

Step 1.

$$t := t + 1.$$

Step 2.

- Generate the random tours $\tau_1, \tau_2, \ldots, \tau_n$ from matrix $P_{t-1} = |p_{t-1,ia}|_{N \times N}$ in the following manner:

Step 2.

- Generate the random tours $\tau_1, \tau_2, \ldots, \tau_n$ from matrix $P_{t-1} = |p_{t-1,ia}|_{N \times N}$ in the following manner:
  - $100\gamma\,\%$ of them are generated based on the rows of P_{t-1}, and

Step 2.

- Generate the random tours $\tau_1, \tau_2, \ldots, \tau_n$ from matrix $P_{t-1} = |p_{t-1,ia}|_{N \times N}$ in the following manner:
  - $100\gamma\,\%$ of them are generated based on the rows of P_{t-1}, and
  - $100(1-\gamma)\,\%$ of them are generated based on the its columns.

Step 2.

- Generate the random tours $\tau_1, \tau_2, \ldots, \tau_n$ from matrix $P_{t-1} = |p_{t-1,ia}|_{N\times N}$ in the following manner:
  - $100\gamma\,\%$ of them are generated based on the rows of P_{t-1}, and
  - $100(1-\gamma)\,\%$ of them are generated based on the its columns.
- The former are called forward tours and the latter reverse tours. If a tour with total length of less than $N\omega$ has been generated; STOP; the graph is Hamiltonian.

Step 2.

- Generate the random tours $\tau_1, \tau_2, \ldots, \tau_n$ from matrix $P_{t-1} = |p_{t-1,ia}|_{N \times N}$ in the following manner:
  - $100\gamma\,\%$ of them are generated based on the rows of P_{t-1}, and
  - $100(1-\gamma)\,\%$ of them are generated based on the its columns.
- The former are called forward tours and the latter reverse tours. If a tour with total length of less than $N\omega$ has been generated; STOP; the graph is Hamiltonian.

Step 3.

- Construct updated matrix P_t with the best m generated tours as in Step 2 of the CE algorithm.

Step 2.

- Generate the random tours $\tau_1, \tau_2, \ldots, \tau_n$ from matrix $P_{t-1} = |p_{t-1,ia}|_{N\times N}$ in the following manner:
  - $100\gamma\,\%$ of them are generated based on the rows of P_{t-1}, and
  - $100(1-\gamma)\,\%$ of them are generated based on the its columns.
- The former are called forward tours and the latter reverse tours. If a tour with total length of less than $N\omega$ has been generated; STOP; the graph is Hamiltonian.

Step 3.

- Construct updated matrix P_t with the best m generated tours as in Step 2 of the CE algorithm.
- If t is "large enough" AND P_t is "not close enough" to being Hamiltonian, STOP; and claim that the graph is (likely) non-Hamiltonian.

Step 4. (NEW STEP)

- Set $\lambda_{ia} := p_{t,ia}$ for all $i = 1, 2, \ldots, N$ and $a \in A(i)$.

Step 4. (NEW STEP)

- Set $\lambda_{ia} := p_{t,ia}$ for all $i = 1, 2, \ldots, N$ and $a \in A(i)$.
- Solve the LP model with

$$\max \sum_{i=1}^{N} \sum_{a \in A(i)} \lambda_{ia} x_{ia}$$

as the objective function subject to:

Step 4. (NEW STEP)

- Set $\lambda_{ia} := p_{t,ia}$ for all $i = 1, 2, \ldots, N$ and $a \in A(i)$.
- Solve the LP model with

$$\max \sum_{i=1}^{N} \sum_{a \in A(i)} \lambda_{ia} x_{ia}$$

as the objective function subject to:

$$\begin{aligned}
\sum_{i=1}^{N} \sum_{a \in A(i)} \left(\delta_{ij} - \beta p(j|i,a)\right) x_{ia} &= \delta_{1j}(1-\beta^{N}), \quad \forall\, j, \\
\sum_{a \in A(1)} x_{1a} &= 1, \\
\text{box-constraints } \beta^{N-1} \le \sum_{a \in A(i)} x_{ia} &\le \beta, \quad \forall \ i \neq 1, \\
x_{ia} &\ge 0, \quad \forall\, i, a.
\end{aligned}$$

- Consider the dominant positive entries of the optimal solution x^*_{ia}. These trace out a subgraph $\Gamma^* \in \Gamma$

- Consider the dominant positive entries of the optimal solution x_{ia}^*. These trace out a subgraph $\Gamma^* \in \Gamma$
- If Γ^* is a tour, the original graph is Hamiltonian.

- Consider the dominant positive entries of the optimal solution x^*_{ia}. These trace out a subgraph $\Gamma^* \in \Gamma$
- If Γ^* is a tour, the original graph is Hamiltonian.
- If Γ^* is not a tour, return to Step 1 and update the transition matrix.

Numerical Results

- Many Hamiltonian graphs in the range of $6 - 324$ nodes have been tested with the above algorithm. In all cases, Hamiltonian cycles were found.

Numerical Results

- Many Hamiltonian graphs in the range of $6 - 324$ nodes have been tested with the above algorithm. In all cases, Hamiltonian cycles were found.

- When the size of graph was small, say less than 50 nodes, often termination occurred in Step 2 by generating a Hamiltonian cycle randomly.

Numerical Results

- Many Hamiltonian graphs in the range of $6 - 324$ nodes have been tested with the above algorithm. In all cases, Hamiltonian cycles were found.

- When the size of graph was small, say less than 50 nodes, often termination occurred in Step 2 by generating a Hamiltonian cycle randomly.

- However, for the larger graphs, in most tests, the algorithm terminated in Step 4 by solving the proposed LP model.

Theoretical Explanation

- Suppose the Hamiltonian graph G is given and based on Step 0 of the new algorithm, the unique shortest tour is θ^*.

Theoretical Explanation

- Suppose the Hamiltonian graph G is given and based on Step 0 of the new algorithm, the unique shortest tour is θ^*.
- Define $\mathbf{x}^*$ as a feasible solution of the linear program given in Step 4 associated with a deterministic policy corresponding to the tour θ^*.

Theoretical Explanation

- Suppose the Hamiltonian graph G is given and based on Step 0 of the new algorithm, the unique shortest tour is θ^*.
- Define $\mathbf{x}^*$ as a feasible solution of the linear program given in Step 4 associated with a deterministic policy corresponding to the tour θ^*.
- Let Λ_t be a transition matrix (obtained from P_t by suitable adjustment) comprising of values of λ_{ia} at iteration t, that is, coefficients of objective function of the linear program proposed in the Step 4.

Theoretical Explanation

- Suppose the Hamiltonian graph G is given and based on Step 0 of the new algorithm, the unique shortest tour is θ^*.
- Define $\mathbf{x}^*$ as a feasible solution of the linear program given in Step 4 associated with a deterministic policy corresponding to the tour θ^*.
- Let Λ_t be a transition matrix (obtained from P_t by suitable adjustment) comprising of values of λ_{ia} at iteration t, that is, coefficients of objective function of the linear program proposed in the Step 4.
- Consider Λ^* as the coefficient matrix associated with a deterministic policy that corresponds to θ^*, that is, elements of Λ^* which coincide with the arcs in θ^* are equal to 1 and the others are equal to 0.

Theoretical Explanation

Proposition

There exists $\delta > 0$ such that, if $\|\Lambda_t - \Lambda^*\| \leq \delta$, then an optimal solution of the linear program given in Step 4 is associated with a deterministic policy corresponding to the shortest tour θ^*.

Theoretical Explanation

Proposition

There exists $\delta > 0$ such that, if $\|\Lambda_t - \Lambda^*\| \leq \delta$, then an optimal solution of the linear program given in Step 4 is associated with a deterministic policy corresponding to the shortest tour θ^*.

Corollary

It can be shown that depending on the graph structure and for β near 1 that,

$$O(N^{-1}) \leq \delta \leq \frac{3}{7}.$$

Finally: Is HCP's NP-completeness an "anomaly"?

- For 18-node cubic graphs of which 39, 635 are Hamiltonian and 1, 666 are non-Hamiltonian ($t = 20$)