Computing Hamiltonian cycles in random graphs

Michael Saunders
Stanford University

Joint work with Ali Eshragh
University of Newcastle, NSW, Australia

The Fifth International Conference on
Numerical Analysis and Optimization (NAOV)

Muscat, Sultanate of Oman   Jan 6–9, 2020

Abstract

A Hamiltonian Cycle is a path that passes once through each node of a graph and returns to the starting node. The HC Problem is a special case of the Traveling Salesman Problem in that it seeks any such path, while the TSP seeks the shortest path. The HCP can be reduced to finding particular vertices of a certain polytope associated with the input graph.

Eshragh et al. (MOR 2019) implemented a simplex-type algorithm to find an HC by moving from a feasible vertex to an adjacent feasible vertex at random. To handle larger graphs, we modified the simplex algorithm in MINOS to do the same. The only change to MINOS is that Phase 2 chooses a random nonbasic variable to enter the basis. (Thus, dual variables are not required in Phase 2.)

The polytope constraints depend on a parameter beta, and the probability of finding an HC depends on beta being close to 1.

With double-precision MINOS we have used beta = 1 - 1e-8.

The quad-precision version of MINOS allows beta = 1 - 1e-16 (say).

We report success rates for random graphs of varying size.

Abstract

A Hamiltonian Cycle is a path that passes once through each node of a graph and returns to the starting node. The HC Problem is a special case of the Traveling Salesman Problem in that it seeks any such path, while the TSP seeks the shortest path. The HCP can be reduced to finding particular vertices of a certain polytope associated with the input graph.

Eshragh et al. (MOR 2019) implemented a simplex-type algorithm to find an HC by moving from a feasible vertex to an adjacent feasible vertex at random. To handle larger graphs, we modified the simplex algorithm in MINOS to do the same. The only change to MINOS is that Phase 2 chooses a random nonbasic variable to enter the basis. (Thus, dual variables are not required in Phase 2.)

The polytope constraints depend on a parameter beta, and the probability of finding an HC depends on beta being close to 1.
With double-precision MINOS we have used beta = 1 - 1e-8.
The quad-precision version of MINOS allows beta = 1 - 1e-16 (say).
We report success rates for random graphs of varying size.

Abstract

A Hamiltonian Cycle is a path that passes once through each node of a graph and returns to the starting node. The HC Problem is a special case of the Traveling Salesman Problem in that it seeks any such path, while the TSP seeks the shortest path. The HCP can be reduced to finding particular vertices of a certain polytope associated with the input graph.

Eshragh et al. (MOR 2019) implemented a simplex-type algorithm to find an HC by moving from a feasible vertex to an adjacent feasible vertex at random. To handle larger graphs, we modified the simplex algorithm in MINOS to do the same. The only change to MINOS is that Phase 2 chooses a random nonbasic variable to enter the basis. (Thus, dual variables are not required in Phase 2.)

The polytope constraints depend on a parameter beta, and the probability of finding an HC depends on beta being close to 1.

With double-precision MINOS we have used beta = 1 - 1e-8.

The quad-precision version of MINOS allows beta = 1 - 1e-16 (say).

We report success rates for random graphs of varying size.

Definition

Hamiltonian Cycle (HC)

Given a graph G, a simple **path** that starts from an arbitrary node, visits all nodes exactly **once** and **returns** to the initial node is called a **Hamiltonian Cycle**.

Hamiltonian Cycle Problem (HCP)

Given a graph G, **determine** whether it contains at least **one** HC or **not**.

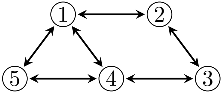
Definition

Hamiltonian Cycle (HC)

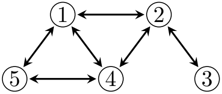
Given a graph G, a simple **path** that starts from an arbitrary node, visits all nodes exactly **once** and **returns** to the initial node is called a **Hamiltonian Cycle**.

Hamiltonian Cycle Problem (HCP)

Given a graph G, **determine** whether it contains at least **one** HC or **not**.



(a) A Hamiltonian graph



(b) A non-Hamiltonian graph

Notation

HCP in this talk

- There is a **directed** graph G on n nodes with no self-loops.
- $\mathcal{S} = \{1, 2, \ldots, n\}$ is the set of all **nodes** and $\mathscr{A}$ is the set of all **arcs** in this graph.
- For each node i, we define **two** subsets

$$\begin{cases} \mathcal{A}(i) = \{a \in \mathcal{S} \mid (i,a) \in \mathscr{A}\} \\ \mathcal{B}(i) = \{b \in \mathcal{S} \mid (b,i) \in \mathscr{A}\} \end{cases}$$

Notation

HCP in this talk

- There is a **directed** graph G on n nodes with no self-loops.
- $\mathcal{S} = \{1, 2, \ldots, n\}$ is the set of all **nodes** and $\mathscr{A}$ is the set of all **arcs** in this graph.
- For each node i, we define **two** subsets

$$\begin{cases} \mathcal{A}(i) = \{a \in \mathcal{S} \mid (i, a) \in \mathscr{A}\} \\ \mathcal{B}(i) = \{b \in \mathcal{S} \mid (b, i) \in \mathscr{A}\} \end{cases}$$

Notation

HCP in this talk

- There is a **directed** graph G on n nodes with no self-loops.
- $\mathcal{S} = \{1, 2, \ldots, n\}$ is the set of all **nodes** and $\mathscr{A}$ is the set of all **arcs** in this graph.
- For each node i, we define **two** subsets

$$\begin{cases} \mathcal{A}(i) = \{a \in \mathcal{S} \mid (i, a) \in \mathscr{A}\} \\ \mathcal{B}(i) = \{b \in \mathcal{S} \mid (b, i) \in \mathscr{A}\} \end{cases}$$

Embedding in Markov Decision Processes (MDPs)

- In 1994, Filar and Krass developed a model for HCP by embedding it in a **perturbed Markov decision process**.
- They converted the deterministic HCP to a particular **average-reward Markov decision process**.
- In 2000, Feinberg converted the HCP to a class of Markov decision processes, the so-called **weighted discounted Markov decision processes**.
- MDP embedding implies that you can search for a Hamiltonian cycle in a nicely structured **polyhedral domain of discounted occupational measures**.

Embedding in Markov Decision Processes (MDPs)

- In 1994, Filar and Krass developed a model for HCP by embedding it in a **perturbed Markov decision process**.
- They converted the deterministic HCP to a particular **average-reward Markov decision process**.
- In 2000, Feinberg converted the HCP to a class of Markov decision processes, the so-called **weighted discounted Markov decision processes**.
- MDP embedding implies that you can search for a Hamiltonian cycle in a nicely structured **polyhedral domain of discounted occupational measures**.

Embedding in Markov Decision Processes (MDPs)

- In 1994, Filar and Krass developed a model for HCP by embedding it in a **perturbed Markov decision process**.
- They converted the deterministic HCP to a particular **average-reward Markov decision process**.
- In 2000, Feinberg converted the HCP to a class of Markov decision processes, the so-called **weighted discounted Markov decision processes**.
- MDP embedding implies that you can search for a Hamiltonian cycle in a nicely structured **polyhedral domain of discounted occupational measures**.

Embedding in Markov Decision Processes (MDPs)

- In 1994, Filar and Krass developed a model for HCP by embedding it in a **perturbed Markov decision process**.
- They converted the deterministic HCP to a particular **average-reward Markov decision process**.
- In 2000, Feinberg converted the HCP to a class of Markov decision processes, the so-called **weighted discounted Markov decision processes**.
- MDP embedding implies that you can search for a Hamiltonian cycle in a nicely structured **polyhedral domain of discounted occupational measures**.

Domain of discounted occupational measures

$\mathcal{H}_\beta(G)$ Polytope
associated with graph G and discount factor $\beta \in (0,1)$

$$\sum_{a\in\mathcal{A}(1)} x_{1a} - \beta \sum_{b\in\mathcal{B}(1)} x_{b1} = 1 - \beta^n$$

$$\sum_{a\in\mathcal{A}(i)} x_{ia} - \beta \sum_{b\in\mathcal{B}(i)} x_{bi} = 0, \quad i = 2, 3, \dots, n$$

$$\sum_{a\in\mathcal{A}(1)} x_{1a} = 1$$

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

Hamiltonian extreme points

Theorem (Feinberg 2000)

If the graph G is Hamiltonian, then corresponding to each **HC** *in the graph, there exists an* **extreme point** *of polytope $\mathcal{H}_\beta(G)$, called* **Hamiltonian extreme point***.*

If x is a Hamiltonian extreme point, then it has exactly n **positive coordinates** *tracing out a Hamiltonian cycle in G.*

Otherwise, that is if an extreme point does **not** *have n positive coordinates, it is called a* **non-Hamiltonian extreme point***.*

Hamiltonian extreme points

Theorem (Feinberg 2000)

If the graph G is Hamiltonian, then corresponding to each **HC** *in the graph, there exists an* **extreme point** *of polytope $\mathcal{H}_\beta(G)$, called* **Hamiltonian extreme point**.

If x is a Hamiltonian extreme point, then it has exactly n **positive coordinates** *tracing out a Hamiltonian cycle in G.*

Otherwise, that is if an extreme point does **not** *have n positive coordinates, it is called a* **non-Hamiltonian extreme point**.

Hamiltonian extreme points

Theorem (Feinberg 2000)

If the graph G is Hamiltonian, then corresponding to each **HC** *in the graph, there exists an* **extreme point** *of polytope $\mathcal{H}_\beta(G)$, called* **Hamiltonian extreme point**.

If x is a Hamiltonian extreme point, then it has exactly n **positive coordinates** *tracing out a Hamiltonian cycle in G.*

Otherwise, that is if an extreme point does **not** *have n positive coordinates, it is called a* **non-Hamiltonian extreme point**.

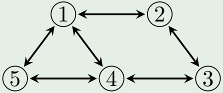
Illustration

Example

$$x_{12} + x_{14} + x_{15} - \beta(x_{21} + x_{41} + x_{51}) = 1 - \beta^5$$

$$x_{21} + x_{23} - \beta(x_{12} + x_{32}) = 0$$
$$x_{32} + x_{34} - \beta(x_{23} + x_{43}) = 0$$
$$x_{41} + x_{43} + x_{45} - \beta(x_{14} + x_{34} + x_{54}) = 0$$
$$x_{51} + x_{54} - \beta(x_{15} + x_{45}) = 0$$

$$x_{12} + x_{14} + x_{15} = 1$$
$$x_{12}, x_{14}, \ldots, x_{54} \geq 0$$



Example (cont.)

- One particular **Hamiltonian extreme point:**

$$x_{12} = 1,\ x_{23} = \beta,\ x_{34} = \beta^2,\ x_{45} = \beta^3, x_{51} = \beta^4$$

$$x_{ia} = 0 \ \text{ for all other possible values}$$

- It traces out the **HC**
$1 \to 2 \to 3 \to 4 \to 5 \to 1$

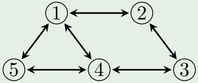
Example (cont.)

- One particular **Hamiltonian extreme point:**

$$x_{12} = 1,\ x_{23} = \beta,\ x_{34} = \beta^2,\ x_{45} = \beta^3, x_{51} = \beta^4$$

$x_{ia} = 0$ for all other possible values

- It traces out the **HC**
$1 \to 2 \to 3 \to 4 \to 5 \to 1$



A new random search algorithm

- The correspondence between the **HCs** in graph G and **extreme points** of polytope $\mathcal{H}_\beta(G)$ can be exploited to develop an **algorithm** that **searches** for Hamiltonian cycles.
- As the polytope $\mathcal{H}_\beta(G)$ might have many **degenerate** extreme points, it would be easier to run such a search algorithm on the **feasible bases** of $\mathcal{H}_\beta(G)$.
- As $\mathcal{H}_\beta(G)$ has $n+1$ non-redundant equality **constraints**, an extreme point of this polytope is called **degenerate** if it has **less** than $n+1$ non-zero components. Otherwise (if it has **exactly** $n+1$ non-zero components), it is **non-degenerate**.
- Analogously, we can define **Hamiltonian** and **non-Hamiltonian bases** corresponding to extreme points of $\mathcal{H}_\beta(G)$.
- Thus, a key **issue** influencing the **efficiency** of such a search algorithm is the existence of a **sufficiently large** number of **Hamiltonian bases**.

A new random search algorithm

- The correspondence between the **HCs** in graph G and **extreme points** of polytope $\mathcal{H}_\beta(G)$ can be exploited to develop an **algorithm** that **searches** for Hamiltonian cycles.
- As the polytope $\mathcal{H}_\beta(G)$ might have many **degenerate** extreme points, it would be easier to run such a search algorithm on the **feasible bases** of $\mathcal{H}_\beta(G)$.
- As $\mathcal{H}_\beta(G)$ has $n+1$ non-redundant equality **constraints**, an extreme point of this polytope is called **degenerate** if it has **less** than $n+1$ non-zero components. Otherwise (if it has **exactly** $n+1$ non-zero components), it is **non-degenerate**.
- Analogously, we can define **Hamiltonian** and **non-Hamiltonian bases** corresponding to extreme points of $\mathcal{H}_\beta(G)$.
- Thus, a key **issue** influencing the **efficiency** of such a search algorithm is the existence of a **sufficiently large** number of **Hamiltonian bases**.

A new random search algorithm

- The correspondence between the **HCs** in graph G and **extreme points** of polytope $\mathcal{H}_\beta(G)$ can be exploited to develop an **algorithm** that **searches** for Hamiltonian cycles.
- As the polytope $\mathcal{H}_\beta(G)$ might have many **degenerate** extreme points, it would be easier to run such a search algorithm on the **feasible bases** of $\mathcal{H}_\beta(G)$.
- As $\mathcal{H}_\beta(G)$ has $n+1$ non-redundant equality **constraints**, an extreme point of this polytope is called **degenerate** if it has **less** than $n+1$ non-zero components. Otherwise (if it has **exactly** $n+1$ non-zero components), it is **non-degenerate**.
- Analogously, we can define **Hamiltonian** and **non-Hamiltonian bases** corresponding to extreme points of $\mathcal{H}_\beta(G)$.
- Thus, a key **issue** influencing the **efficiency** of such a search algorithm is the existence of a **sufficiently large** number of **Hamiltonian bases**.

A new random search algorithm

- The correspondence between the **HCs** in graph G and **extreme points** of polytope $\mathcal{H}_\beta(G)$ can be exploited to develop an **algorithm** that **searches** for Hamiltonian cycles.
- As the polytope $\mathcal{H}_\beta(G)$ might have many **degenerate** extreme points, it would be easier to run such a search algorithm on the **feasible bases** of $\mathcal{H}_\beta(G)$.
- As $\mathcal{H}_\beta(G)$ has $n+1$ non-redundant equality **constraints**, an extreme point of this polytope is called **degenerate** if it has **less** than $n+1$ non-zero components. Otherwise (if it has **exactly** $n+1$ non-zero components), it is **non-degenerate**.
- Analogously, we can define **Hamiltonian** and **non-Hamiltonian bases** corresponding to extreme points of $\mathcal{H}_\beta(G)$.
- Thus, a key **issue** influencing the **efficiency** of such a search algorithm is the existence of a **sufficiently large** number of **Hamiltonian bases**.

A new random search algorithm

- The correspondence between the **HCs** in graph G and **extreme points** of polytope $\mathcal{H}_\beta(G)$ can be exploited to develop an **algorithm** that **searches** for Hamiltonian cycles.
- As the polytope $\mathcal{H}_\beta(G)$ might have many **degenerate** extreme points, it would be easier to run such a search algorithm on the **feasible bases** of $\mathcal{H}_\beta(G)$.
- As $\mathcal{H}_\beta(G)$ has $n+1$ non-redundant equality **constraints**, an extreme point of this polytope is called **degenerate** if it has **less** than $n+1$ non-zero components. Otherwise (if it has **exactly** $n+1$ non-zero components), it is **non-degenerate**.
- Analogously, we can define **Hamiltonian** and **non-Hamiltonian bases** corresponding to extreme points of $\mathcal{H}_\beta(G)$.
- Thus, a key **issue** influencing the **efficiency** of such a search algorithm is the existence of a **sufficiently large** number of **Hamiltonian bases**.

Definitions

- Let x be an extreme point of $\mathcal{H}_\beta(G)$. The **support** of x, denoted by $S(G, x)$, is defined to be a **subgraph** of G with **node set** $\mathcal{S}$ and **arc set** $\{(i, a) \in \mathscr{A} : x_{ia} > 0\}$.
- Clearly, if x is a Hamiltonian extreme point, the support graph $S(G, x)$ is a **Hamiltonian cycle**.
- A simple path that starts from node 1 and returns to it in fewer than n arcs is called a **short cycle**.
- A **noose path** is a simple path that starts from node 1 and returns to some node other than node 1.
- In the following graph, arcs $(1, 4), (4, 5), (5, 1)$ form a **short cycle**, and arcs $(1, 2), (2, 3), (3, 2)$ form a **noose path**.

Definitions

- Let x be an extreme point of $\mathcal{H}_\beta(G)$. The **support** of x, denoted by $S(G, x)$, is defined to be a **subgraph** of G with **node set** $\mathcal{S}$ and **arc set** $\{(i,a) \in \mathscr{A} : x_{ia} > 0\}$.
- Clearly, if x is a Hamiltonian extreme point, the support graph $S(G, x)$ is a **Hamiltonian cycle**.
- A simple path that starts from node 1 and returns to it in fewer than n arcs is called a **short cycle**.
- A **noose path** is a simple path that starts from node 1 and returns to some node other than node 1.
- In the following graph, arcs $(1,4),(4,5),(5,1)$ form a **short cycle**, and arcs $(1,2),(2,3),(3,2)$ form a **noose path**.

Definitions

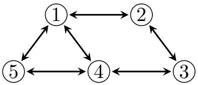
- Let x be an extreme point of $\mathcal{H}_\beta(G)$. The **support** of x, denoted by $S(G,x)$, is defined to be a **subgraph** of G with **node set** $\mathcal{S}$ and **arc set** $\{(i,a) \in \mathscr{A} : x_{ia} > 0\}$.
- Clearly, if x is a Hamiltonian extreme point, the support graph $S(G,x)$ is a **Hamiltonian cycle**.
- A simple path that starts from node 1 and returns to it in fewer than n arcs is called a **short cycle**.
- A **noose path** is a simple path that starts from node 1 and returns to some node other than node 1.
- In the following graph, arcs $(1,4),(4,5),(5,1)$ form a **short cycle**, and arcs $(1,2),(2,3),(3,2)$ form a **noose path**.

Definitions

- Let $\boldsymbol{x}$ be an extreme point of $\mathcal{H}_\beta(G)$. The **support** of $\boldsymbol{x}$, denoted by $S(G,\boldsymbol{x})$, is defined to be a **subgraph** of G with **node set** $\mathcal{S}$ and **arc set** $\{(i,a) \in \mathscr{A} : x_{ia} > 0\}$.
- Clearly, if $\boldsymbol{x}$ is a Hamiltonian extreme point, the support graph $S(G,\boldsymbol{x})$ is a **Hamiltonian cycle**.
- A simple path that starts from node 1 and returns to it in fewer than n arcs is called a **short cycle**.
- A **noose path** is a simple path that starts from node 1 and returns to some node other than node 1.
- In the following graph, arcs $(1,4),(4,5),(5,1)$ form a **short cycle**, and arcs $(1,2),(2,3),(3,2)$ form a **noose path**.

Definitions

- Let $\boldsymbol{x}$ be an extreme point of $\mathcal{H}_\beta(G)$. The **support** of $\boldsymbol{x}$, denoted by $S(G, \boldsymbol{x})$, is defined to be a **subgraph** of G with **node set** $\mathcal{S}$ and **arc set** $\{(i,a) \in \mathscr{A} : x_{ia} > 0\}$.
- Clearly, if $\boldsymbol{x}$ is a Hamiltonian extreme point, the support graph $S(G, \boldsymbol{x})$ is a **Hamiltonian cycle**.
- A simple path that starts from node 1 and returns to it in fewer than n arcs is called a **short cycle**.
- A **noose path** is a simple path that starts from node 1 and returns to some node other than node 1.
- In the following graph, arcs $(1,4), (4,5), (5,1)$ form a **short cycle**, and arcs $(1,2), (2,3), (3,2)$ form a **noose path**.



Hamiltonian and non-Ham. extreme points of $\mathcal{H}_\beta(G)$

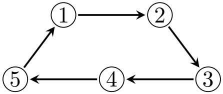
Theorem (Ejov et al. 2009)

Consider a graph G and the corresponding polytope $\mathcal{H}_\beta(G)$. If $\boldsymbol{x}$ is an **extreme point** *of the polytope $\mathcal{H}_\beta(G)$, then the* **support graph** $S(G, \boldsymbol{x})$ *is either a* **Hamiltonian cycle** *or the* **union** *of a* **short cycle** *and a* **noose path**.

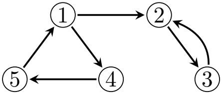
Hamiltonian and non-Ham. extreme points of $\mathcal{H}_\beta(G)$

Theorem (Ejov et al. 2009)

Consider a graph G and the corresponding polytope $\mathcal{H}_\beta(G)$. If $\boldsymbol{x}$ is an **extreme point** *of the polytope $\mathcal{H}_\beta(G)$, then the* **support graph** *$S(G, \boldsymbol{x})$ is either a* **Hamiltonian cycle** *or the* **union** *of a* **short cycle** *and a* **noose path**.

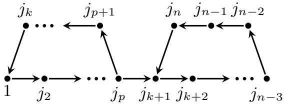


(a) Hamiltonian extreme point

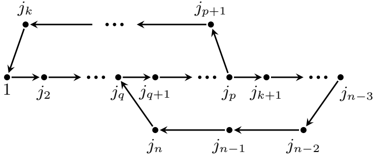


(b) Non-Hamiltonian extreme point

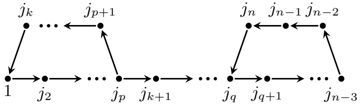
Non-Ham. extreme point-supports [Eshragh & Filar 2011]



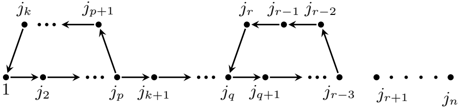
(a) Type 1



(c) Type 3



(b) Type 2



(d) Type 4

- While non-Hamiltonian extreme points of types **1**, **2** and **3** are **non-degenerate**, **Hamiltonian** as well as non-Hamiltonian extreme points of Type **4** are **degenerate**.

The prevalence of Hamiltonian bases

- What is the **number** of each class of **feasible bases** of the polytope $\mathcal{H}_\beta(G)$?
- We utilize **Binomial Random Graphs** $G_{n,p}$.

The prevalence of Hamiltonian bases

- What is the **number** of each class of **feasible bases** of the polytope $\mathcal{H}_\beta(G)$?
- We utilize **Binomial Random Graphs** $G_{n,p}$.

Expected number of feasible bases

Theorem (Eshragh et al. 2019)

Consider the binomial **random graph** $G_{n,p}$ *and the corresponding random polytope* $\mathcal{H}_\beta(G_{n,p})$. *The* **expected number** *of*

1. **Hamiltonian** *bases is* $(n-2)n!p^{n+1}$
2. *non-Hamiltonian bases of*
   - *Type* **1** *is* $\frac{1}{2}(n-3)n!p^{n+1}$
   - *Type* **2** *is* $\frac{1}{6}(n-4)(n-3)(n+1)(n-1)!p^{n+1}$
   - *Type* **3** *is* $\frac{1}{6}(n-2)(n-1)n!p^{n+1}$
   - *Type* **4** *is at least* $(n-1)(n-2)(n-3)^{n-5}2^{n-4}p^{n+1}$

Expected number of feasible bases

Corollary (Eshragh et al. 2019)

In the **random polytope** $\mathcal{H}_\beta(G_{n,p})$, *for sufficiently large* n, *we have*

$$\frac{E\left[\textit{Number of feasible bases of Type 4}\right]}{E\left[\textit{Total number of feasible bases }\right]} \geq 1 - \frac{n^{11/2}}{e^n 2^{n-9}}$$

$$\frac{E\left[\textit{Number of Hamiltonian bases}\right]}{E\left[\textit{Total number of feasible bases}\right]} \leq \frac{n^{9/2}}{e^{n-1} 2^{n-9}}$$

Reducing the feasible region

$\mathcal{H}_\beta(G)$ Polytope

$$\sum_{a\in\mathcal{A}(1)} x_{1a} - \beta \sum_{b\in\mathcal{B}(1)} x_{b1} = 1 - \beta^n$$

$$\sum_{a\in\mathcal{A}(i)} x_{ia} - \beta \sum_{b\in\mathcal{B}(i)} x_{bi} = 0, \quad i = 2, 3, \dots, n$$

$$\sum_{a\in\mathcal{A}(1)} x_{1a} = 1$$

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

Wedge constraints [Eshragh et al. 2011]

$$\beta^{n-1} \leq \sum_{a\in\mathcal{A}(i)} x_{ia} \leq \beta, \quad i = 2, 3, \dots, n$$

$$\mathcal{WH}_\beta(G) = \mathcal{H}_\beta(G) + \text{wedge constraints}$$

The intersection of extreme points

Theorem (Eshragh and Filar 2011)

Consider the graph G and polytopes $\mathcal{H}_\beta(G)$ and $\mathcal{WH}_\beta(G)$. For $\beta \in \left((1-\frac{1}{n-2})^{\frac{1}{n-2}}, 1\right)$, the **intersection** *of extreme points of these two polytopes can be* **partitioned** *into two disjoint (possibly empty) subsets:*

1. **Hamiltonian** *extreme points*
2. **non-Hamiltonian** *extreme points of Type 1*

Investigating HCs through a Simple Random Walk

A Random Walk Algorithm

1. Start from a **feasible basis** of polytope $\mathcal{WH}_\beta$
2. **Uniformly**, choose one of the **adjacent** feasible bases at random and move to that one
3. If the current feasible basis is **Hamiltonian**, then **stop** else **return to** Step 2

Investigating HCs through a Simple Random Walk

A Random Walk Algorithm

1. Start from a **feasible basis** of polytope $\mathcal{WH}_\beta$
2. **Uniformly**, choose one of the **adjacent** feasible bases at random and move to that one
3. If the current feasible basis is **Hamiltonian**, then **stop** else **return to** Step 2

Investigating HCs through a Simple Random Walk

A Random Walk Algorithm

1. Start from a **feasible basis** of polytope $\mathcal{WH}_\beta$
2. **Uniformly**, choose one of the **adjacent** feasible bases at random and move to that one
3. If the current feasible basis is **Hamiltonian**, then **stop** else **return to** Step 2

Numerical results for fixed $\beta = 0.9999$

Iterations to find a Hamiltonian graph

Nodes	Iterations
6	1
10	1
20	10
30	12
40	10
50	2
60	27
80	11
100	29
150	34
200	37
400	52
800	67

Dependence of the Random Walk Algorithm on β

- Random walk with 1000 steps on **feasible bases** of the polytope $\mathcal{WH}_\beta(G)$ for an input sparse Hamiltonian graph G on 30 nodes

β	Number of Hamiltonian bases
0.1	0
0.5	0
0.8	0
0.9	0
0.95	0
0.97	0
0.98	0
0.99	7
0.995	27
0.999	41
0.9999	67
0.99999	70

Dependence of the Random Walk Algorithm on β

- Random walk with 1000 steps on **feasible bases** of the polytope $\mathcal{WH}_\beta(G)$ for an input sparse Hamiltonian graph G on 30 nodes

β	**Number of Hamiltonian bases**
0.1	0
0.5	0
0.8	0
0.9	0
0.95	0
0.97	0
0.98	0
0.99	7
0.995	27
0.999	41
0.9999	67
0.99999	70

Why not just set $\beta = 1$?

The matrix of the polytope $\mathcal{WH}_\beta(G)$ has **singularity** at $\beta = 1$

- Let **polytope** $\mathcal{P}_\epsilon$ be the non-negative points (x_1, x_2) satisfying

$$\begin{cases} x_1 + x_2 = 1 \\ (1+\epsilon)x_1 + (1+2\epsilon)x_2 = 1 + \epsilon \end{cases}$$

- $\mathcal{P}_\epsilon$ has a **singularity** at $\epsilon = 0$ and $\lim_{\epsilon \searrow 0} \mathcal{P}_\epsilon \neq \mathcal{P}_0$

$$\begin{array}{lll} \epsilon > 0 & \Rightarrow & \max x_2 = 0 \\ \epsilon = 0 & \Rightarrow & \max x_2 = 1 \end{array}$$

Why not just set $\beta = 1$?

The matrix of the polytope $\mathcal{WH}_\beta(G)$ has **singularity** at $\beta = 1$

- Let **polytope** $\mathcal{P}_\epsilon$ be the non-negative points (x_1, x_2) satisfying

$$\begin{cases} x_1 + x_2 = 1 \\ (1+\epsilon)x_1 + (1+2\epsilon)x_2 = 1 + \epsilon \end{cases}$$

- $\mathcal{P}_\epsilon$ has a **singularity** at $\epsilon = 0$ and $\lim_{\epsilon \searrow 0} \mathcal{P}_\epsilon \neq \mathcal{P}_0$

$$\begin{aligned} \epsilon > 0 \quad &\Rightarrow \quad \max x_2 = 0 \\ \epsilon = 0 \quad &\Rightarrow \quad \max x_2 = 1 \end{aligned}$$

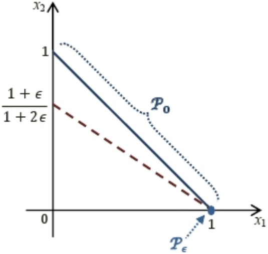
Why not just set $\beta = 1$?

The matrix of the polytope $\mathcal{WH}_\beta(G)$ has **singularity** at $\beta = 1$

- Let **polytope** $\mathcal{P}_\epsilon$ be the non-negative points (x_1, x_2) satisfying

$$\begin{cases} x_1 + x_2 = 1 \\ (1+\epsilon)x_1 + (1+2\epsilon)x_2 = 1+\epsilon \end{cases}$$

- $\mathcal{P}_\epsilon$ has a **singularity** at $\epsilon = 0$ and $\lim_{\epsilon \searrow 0} \mathcal{P}_\epsilon \neq \mathcal{P}_0$



$$\begin{array}{lcl} \epsilon > 0 & \Rightarrow & \max x_2 = 0 \\ \epsilon = 0 & \Rightarrow & \max x_2 = 1 \end{array}$$

Modified simplex method

Implemented in MINOS

1. Phase 1: same as always
2. Phase 2: Replace “price” routine
   - Choose a random nonbasic variable to enter the basis
   - Dual variables π not needed
   - If the current feasible basis is **Hamiltonian**, then **stop**
3. Speed per iteration vs number of iterations
   - Double-precision MINOS is fast
   - Quad-precision MINOS allows β to be very close to 1

Modified simplex method

Implemented in MINOS

1. Phase 1: same as always
2. Phase 2: Replace "price" routine
   - Choose a random nonbasic variable to enter the basis
   - Dual variables π not needed
   - If the current feasible basis is **Hamiltonian**, then **stop**
3. Speed per iteration vs number of iterations
   - Double-precision MINOS is fast
   - Quad-precision MINOS allows β to be very close to 1

Modified simplex method

Implemented in MINOS

1. Phase 1: same as always
2. Phase 2: Replace "price" routine
    - Choose a random nonbasic variable to enter the basis
    - Dual variables π not needed
    - If the current feasible basis is **Hamiltonian**, then **stop**
3. Speed per iteration vs number of iterations
    - Double-precision MINOS is fast
    - Quad-precision MINOS allows β to be very close to 1

Double vs Quad MINOS

	Double	Quad
β	1 - 1e-8	1 - 1e-16
Featol	1e-9	1e-18

Random graphs $p = 0.1$

Nodes	Itns	Itns	Time	Time
100	27997	6017	1	1
200	88109	60802	4	9
300	238857	113929	12	27
400	79383	370891	6	127
500	338272	200321	31	98
600	269592	596965	32	380
700	74212	1838550	11	1493
800	1044635	1072930	184	1107
900	483490	3066025	102	3948
1000	846332	1835241	212	3260
1500	2428213	2732446	1190	8418
2000	1384168	10000000*	1129	49254
2500	7578426	5536333	11484	42096
3000	10000000*	9780955	20258	97673
3500	1375760	9116285	3944	119274
4000	9980379!	10000000*	36500	165062
5000	8479705	10000000*	46541	251378

Double vs Quad MINOS

	Double	Quad
β	1 - 1e-8	1 - 1e-16
Featol	1e-9	1e-18

Random graphs $p = 0.1$

Nodes	Itns	Itns	Time	Time
100	27997	6017	1	1
200	88109	60802	4	9
300	238857	113929	12	27
400	79383	370891	6	127
500	338272	200321	31	98
600	269592	596965	32	380
700	74212	1838550	11	1493
800	1044635	1072930	184	1107
900	483490	3066025	102	3948
1000	846332	1835241	212	3260
1500	2428213	2732446	1190	8418
2000	1384168	10000000*	1129	49254
2500	7578426	5536333	11484	42096
3000	10000000*	9780955	20258	97673
3500	1375760	9116285	3944	119274
4000	9980379!	10000000*	36500	165062
5000	8479705	10000000*	46541	251378

Further development

Main reference

- Ali Eshragh, Jerzy Filar, Thomas Kalinowski, Sogol Mohammadian (2019). Hamiltonian cycles and subsets of discounted occupational measures. *Mathematics of Operations Research*.

Conjecture

- $\exists$ *positive* c, δ, k *such that for all* $\beta \in (1 - e^{-cn}, 1)$, *with high probability, the expected proportion of feasible bases of* $\mathcal{WH}_\beta(G_{n,p})$ *that are quasi-Hamiltonian is at least* δ/n^k.

New

- Thomas Kalinowski and Sogol Mohammadian (2019). Feasible bases for a polytope related to the Hamilton cycle problem. arXiv.org:1907.12691.

The set of feasible bases is independent of β when it is close to 1.

Further development

Main reference

- Ali Eshragh, Jerzy Filar, Thomas Kalinowski, Sogol Mohammadian (2019). Hamiltonian cycles and subsets of discounted occupational measures. *Mathematics of Operations Research*.

Conjecture

- $\exists$ *positive* c, δ, k *such that for all* $\beta \in (1 - e^{-cn}, 1)$, *with high probability, the expected proportion of feasible bases of* $\mathcal{WH}_\beta(G_{n,p})$ *that are quasi-Hamiltonian is at least* δ/n^k.

New

- Thomas Kalinowski and Sogol Mohammadian (2019). Feasible bases for a polytope related to the Hamilton cycle problem. arXiv.org:1907.12691.

  The set of feasible bases is independent of β when it is close to 1.

Conclusions

$\mathcal{WH}_\beta(G)$ Polytope

$$\sum_{a\in\mathcal{A}(1)} x_{1a} - \beta \sum_{b\in\mathcal{B}(1)} x_{b1} = 1 - \beta^n$$

$$\sum_{a\in\mathcal{A}(i)} x_{ia} - \beta \sum_{b\in\mathcal{B}(i)} x_{bi} = 0, \quad i = 2, 3, \dots, n$$

$$\sum_{a\in\mathcal{A}(1)} x_{1a} = 1$$

$$x_{ia} \ge 0 \quad \forall\, i \in \mathcal{S},\ a \in \mathcal{A}(i)$$

$$\beta^{n-1} \le \sum_{a\in\mathcal{A}(i)} x_{ia} \le \beta, \quad i = 2, 3, \dots, n$$

- Even though $\beta = 1 \Rightarrow$ singularity,
  $B = LU$ in MINOS says all basis matrices are extremely well-conditioned!
- Double $\beta = 1$ - 1e-8 and Quad $\beta = 1$ - e-16 are both reliable
- Quad MINOS is (only) 5–10 times slower per iteration (but not really needed)
- The apparent need for Quad MINOS brought us together (in Berkeley and Muscat!)

Conclusions

$\mathcal{WH}_\beta(G)$ Polytope

$$\sum_{a\in\mathcal{A}(1)} x_{1a} - \beta \sum_{b\in\mathcal{B}(1)} x_{b1} = 1 - \beta^n$$

$$\sum_{a\in\mathcal{A}(i)} x_{ia} - \beta \sum_{b\in\mathcal{B}(i)} x_{bi} = 0, \quad i = 2,3,\dots,n$$

$$\sum_{a\in\mathcal{A}(1)} x_{1a} = 1$$

$$x_{ia} \ge 0 \quad \forall\, i \in \mathcal{S},\ a \in \mathcal{A}(i)$$

$$\beta^{n-1} \le \sum_{a\in\mathcal{A}(i)} x_{ia} \le \beta, \quad i = 2,3,\dots,n$$

- Even though $\beta = 1 \Rightarrow$ singularity,
  $B = LU$ in MINOS says all basis matrices are extremely well-conditioned!
- Double $\beta = 1$ - 1e-8 and Quad $\beta = 1$ - e-16 are both reliable
- Quad MINOS is (only) 5–10 times slower per iteration (but not really needed)
- The apparent need for Quad MINOS brought us together (in Berkeley and Muscat!)

Conclusions

$\mathcal{WH}_\beta(G)$ Polytope

$$\sum_{a\in\mathcal{A}(1)} x_{1a} - \beta \sum_{b\in\mathcal{B}(1)} x_{b1} = 1 - \beta^n$$

$$\sum_{a\in\mathcal{A}(i)} x_{ia} - \beta \sum_{b\in\mathcal{B}(i)} x_{bi} = 0, \quad i = 2,3,\ldots,n$$

$$\sum_{a\in\mathcal{A}(1)} x_{1a} = 1$$

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

$$\beta^{n-1} \leq \sum_{a\in\mathcal{A}(i)} x_{ia} \leq \beta, \quad i = 2,3,\ldots,n$$

- Even though $\beta = 1 \Rightarrow$ singularity,
  $B = LU$ in MINOS says all basis matrices are extremely well-conditioned!
- Double $\beta = 1$ - 1e-8 and Quad $\beta = 1$ - e-16 are both reliable
- Quad MINOS is (only) 5–10 times slower per iteration (but not really needed)
- The apparent need for Quad MINOS brought us together (in Berkeley and Muscat!)

Conclusions

$\mathcal{WH}_\beta(G)$ Polytope

$$\sum_{a\in\mathcal{A}(1)} x_{1a} - \beta \sum_{b\in\mathcal{B}(1)} x_{b1} = 1-\beta^n$$

$$\sum_{a\in\mathcal{A}(i)} x_{ia} - \beta \sum_{b\in\mathcal{B}(i)} x_{bi} = 0, \quad i=2,3,\dots,n$$

$$\sum_{a\in\mathcal{A}(1)} x_{1a} = 1$$

$$x_{ia} \ge 0 \quad \forall\, i\in\mathcal{S},\ a\in\mathcal{A}(i)$$

$$\beta^{n-1} \le \sum_{a\in\mathcal{A}(i)} x_{ia} \le \beta, \quad i=2,3,\dots,n$$

- Even though $\beta = 1 \Rightarrow$ singularity,
  $B = LU$ in MINOS says all basis matrices are extremely well-conditioned!
- Double $\beta = 1$ - 1e-8 and Quad $\beta = 1$ - e-16 are both reliable
- Quad MINOS is (only) 5–10 times slower per iteration (but not really needed)
- The apparent need for Quad MINOS brought us together (in Berkeley and Muscat!)

Ali Eshragh, Stanford, Oct 2018

Ali Eshragh and friends, Berkeley, Dec 2019

NAOIV Jan 2017

NAOIV Jan 2017

NAOIV Jan 2017