Hamiltonian Cycle Problem, Markov Decision Processes and Interior Point Methods

Jacek Gondzio

Dept. of Maths & Stats, Univ. of Edinburgh
Email: gondzio@maths.ed.ac.uk
URL: http://www.maths.ed.ac.uk/~gondzio

McMaster University, 24 May, 2002

In collaboration with:
Vladimir Ejov and Jerzy Filar
University of South Australia, Adelaide

- Hamiltonian Cycle Problem
- Unorthodox approach
  - Markov Decision Processes
  - indefinite QPs
- Use IPMs to solve the problem
  - IPM for QP;
  - sparsity issues (separable QPs);
  - numerical issues (non-convex QPs);
- Test problems
  - Knight Tour Problem, $k \times k$ chessboard
  - Randomly generated problems
- What to do at a local minimum
  - Branching (implemented)
  - Decomposition (not implemented)
  - Cuts (not implemented)
- Conclusions

Hamiltonian Cycle Problem

Notation:

$\mathcal{G} = (V, E)$ is a directed graph with **nodes** V and (directed) **arcs** $E \subset \{(i,j) : i \in V, j \in V, i \neq j\}$.
\# of nodes: $m = |V|$, \# of arcs: $n = |E|$.
Let $\mathcal{A}(i)$ be a set of arcs emanating from node i.

The *Hamiltonian Cycle Problem* (HCP) consists in finding a cycle in a directed graph that enters every node exactly once, or determine that no such cycle exists.

In other words, we look for a cycle
$(i_1, i_2), (i_2, i_3), \ldots, (i_{m-1}, i_m), (i_m, i_1)$
such that $(i_{k-1}, i_k) \in E$ and $i_k \neq i_l$ for $k \neq l$.

HCP is an *NP-complete* problem.

Suppose we additionally associate a cost c_{ij} of traversing an arc (i,j). c_{ij} can be, for example, a distance from node i to node j.
The *Traveling Salesman Problem* (TSP), known to be very hard, consists in finding an HCP of the minimum cost.

From HCP via MDP to QP

Given an **HCP**,
embed the graph problem into an **MDP**,
perturb the MDP to make it unichain,
characterize unichain MDP via LP.

Solve a mixed integer LP:
either as an MIP,
or as a non-convex **QP**.

What to do in a local minimum?
Use a heuristic to reduce the graph.

A (finite state) Markov chain is a system which can be in a certain finite number of states labeled $V = \{1, 2, \ldots, m\}$. At time $t = 1, 2, \ldots$, it moves from its current state i to a new state j. The probabilities p_{ij} determine the moves:

$$p_{ij} = \mathrm{P}\ (\text{system moves from state } i \text{ to state } j),$$

independently of time.
We have $p_{ij} \geq 0$, $\forall i, j \in V$ and $\sum_j p_{ij} = 1$, $\forall i \in V$. The *transition probabilities* are gathered in a matrix

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mm} \end{bmatrix},$$

called the *probability transition matrix* of MC.
Lack of memory: Knowing P, the only information we need to determine the probability of the MC being in any given state after the next transition is its current state; history is irrelevant.
Example

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ q & 0 & p & 0 \\ 0 & q & 0 & p \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Sets of states $C_1 = \{1\}$ and $C_2 = \{4\}$ are *closed* and *irreducible*. States $v \in \{2, 3\}$ are *transient*.

An MDP , is observed at discrete time points $t = 1, 2, \ldots$ Its *state space* is denoted by $V = \{1, 2, \ldots, m\}$. With each state $i \in V$ a set of *actions* $\mathcal{A}(i)$ is associated.
At time t the system is in state i and an action $a \in \mathcal{A}(i)$ is chosen by a decision maker. This gains reward r_{ia} and the process moves to a state $j \in V$ with the probability $p_{iaj} \geq 0$, where $\sum_{j=1}^{m} p_{iaj} = 1$.

A *decision rule* f^t at time t is a function which assigns a probability to the event that action a is taken at time t. A *policy* $f = (f^1, f^2, \ldots, f^t, \ldots)$, is a sequence of decision rules. A policy is called **stationary** if all its decision rules are identical and depend only on the current state. A policy is called **deterministic** if it is stationary and has nonrandomized decision rules.

Let $\gamma = (\gamma_1, \ldots, \gamma_m)$ be the initial distribution of the states of , : $\gamma_i = P(X_1 = i)$ and $\sum_{i=1}^{m} = 1$. Given a stationary policy f, let

$$p_{ij}(f) = \sum_{a \in \mathcal{A}(i)} p_{iaj} f(i, a).$$

Now the policy f defines a Markov chain with the probability transition matrix

$$P(f) = [p_{ij}(f)]_{i,j=1}^{m}.$$

Long-run expected state-action frequency

For any stationary policy f, initial distribution γ, $j \in V$ and $a \in \mathcal{A}(j)$, define

$$x_{ja}^{T}(f) = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{m} \gamma_i P_f(X_t = j, Y_t = a | X_1 = i).$$

We define a vector $x^T(f)$ in a space of dimension $\sum_{j=1}^{m} |\mathcal{A}(j)|$. Let $X(f)$ denote the set of all limit points of the vectors $\{x^T(f) | T = 1, 2, \ldots\}$ as $T \to \infty$. The limit $x(f)$ is called the **long-run expected state-action frequency vector** induced by the policy f. Similarly, the long-run expected frequencies of visits to any state $j \in V$ under the policy f are given by

$$x_j(f) = \sum_{a \in \mathcal{A}(j)} x_{ja}(f).$$

The Markov Decision Process is called **unichain** if for any deterministic policy f, the Markov chain induced by $P(f)$ has one ergodic set plus a (possibly empty) set of transient states.

Unichain MDP and Linear Constraints

It is known that for a **unichain** MDP, the set

$$X = \{x(f) | f \text{ is a stationary policy}\}$$

is fully characterized by the linear constraints

$$\sum_{i \in V} \sum_{a \in \mathcal{A}(i)} (\delta_{ij} - p_{iaj})\, x_{ia} = 0, \quad j \in V,$$
$$\sum_{i \in V} \sum_{a \in \mathcal{A}(i)} x_{ia} = 1,$$
$$x_{ia} \geq 0, \quad i \in V,\ a \in \mathcal{A}(i).$$

Let $C(S)$ be the class of stationary strategies of the unichain MDP. A map $T : X \mapsto C(S)$, where $T(x) = f_x$ is defined as follows

$$f_x(i, a) = \begin{cases} x_{ia}/x_i & \text{if } x_i = \sum_{a \in \mathcal{A}(i)} x_{ia} > 0, \\ 1 & \text{if } x_i = 0 \text{ and } a = a_1, \\ 0 & \text{if } x_i = 0 \text{ and } a \neq a_1, \end{cases}$$

where a_1 is the first available action in a given state according to some fixed ordering.
Another map: $\hat{T} : C(S) \mapsto X$, where $\hat{T}(f) = x(f)$:

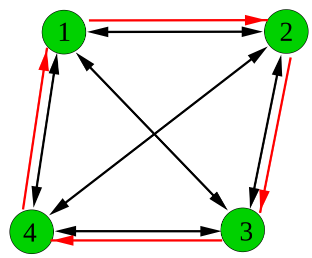
$$x_{ia}(f) = \pi_i^*(f)\, f(i, a), \quad i \in V,\ a \in \mathcal{A}(i).$$

Here π_i^* is the ith entry of the unique fixed probability vector (stationary distribution) of $P(f)$.

Derman '70, Kallenberg '83, Filar and Krass '94:
Theorem: Let , be a unichain MDP.
If $L(S) = \{x(f) | f \in C(S)\}$, then $X = L(S)$, and the extreme points of X are those x for which f_x is a deterministic policy.
In general, for a deterministic policy f, $x(f)$ is not an extreme point of X.

A *node* of $\mathcal{G}$ is a *state* in MDP.
An *arc* of $\mathcal{G}$ is an *action* in MDP.
An *HC* in $\mathcal{G}$ is a *deterministic policy* in MDP.

Consider a 4-node complete graph



and a Hamiltonian cycle $\mathcal{H} = \{(1,2)(2,3)(3,4)(4,1)\}$.

The Hamiltonian cycle $\mathcal{H}$ induces a Markov chain with the following transition matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

This is an *irreducible* Markov chain, i.e., all its states belong to one ergodic class.

Consider another Markov chain with the following transition matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

that has two distinct ergodic classes.

Suppose we **perturb** the arc $(3,4)$ and replace it with two arcs $(3,4)$ and $(3,1)$. Similarly, we replace the arc $(4,3)$ with two arcs $(4,3)$ and $(4,1)$. We thus allow to return from any node to a **home** node 1.
We have two **fictitious** arcs: $(3,1)$ and $(4,1)$.

The new transition matrix has the form

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 1-\varepsilon \\ \varepsilon & 0 & 1-\varepsilon & 0 \end{bmatrix}.$$

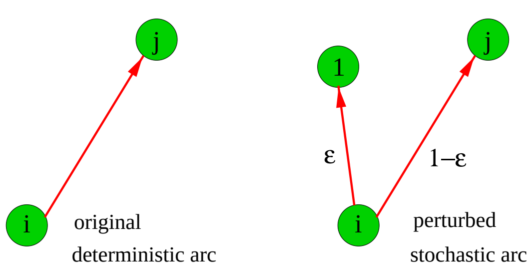
The Markov control problem changes now to a perturbed one with the property that any Markov chain induced by a stationary policy possesses a single ergodic class and a (possibly empty) set of transient states.

ε-perturbed Embedding

The embedding of the graph $\mathcal{G}$ in an MDP , suggests the analysis to be carried out in the space X of the long-run state-action frequencies, the union of $\{x(f)\}$ over all policies f.

Advantage: polyhedral characterization of X.

Note that the perturbation ensures that , is unichain, if we assign an $\varepsilon > 0$ probability to the "going home" arcs.



Perturbed MDP

An ε-perturbed MDP , (ε) is "close" to the original one , (for a small ε) and it is **unichain**.

Filar and Krass 1994
define a smaller polyhedron $\bar{X}(\varepsilon)$ that satisfies the following linear constraints:

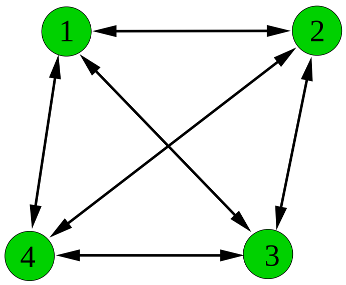
$$\begin{aligned}
&\textstyle\sum_{i \in V} \sum_{a \in \mathcal{A}(i)} (\delta_{ij} - p_{iaj}(\varepsilon))\, x_{ia} = 0, \quad j \in V, \\
&\textstyle\sum_{i \in V} \sum_{a \in \mathcal{A}(i)} x_{ia} = 1, \\
&\textstyle\sum_{a \in \mathcal{A}(i)} x_{ia} \geq c(\varepsilon), \quad j \in V, \\
&\textstyle\sum_{j \in V} x_{1j} = c(\varepsilon), \\
&x_{ia} \geq 0, \quad i \in V,\ a \in \mathcal{A}(i),
\end{aligned}$$

where $c(\varepsilon)$ is a given number.

Theorem: Hamiltonian cycles of a graph $\mathcal{G}$ are in 1:1 correspondence with vectors x satisfying:

$(i)\quad x \in \bar{X}(\varepsilon)$
$(ii)\quad x_{ia} / \sum_{a \in \mathcal{A}(i)} x_{ia} \in \{0,1\},\ i \in V,\ a \in \mathcal{A}(i).$

Further, if x is such a vector,
then f_x is a Hamiltonian cycle of $\mathcal{G}$.



The node-arc incidence matrix $A \in \mathcal{R}^{4\times 12}$ of this graph has the form

$$A=\begin{bmatrix} 1 & 1 & 1 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 1 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 1 & 1 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 1 & 1 \end{bmatrix}.$$

The constraint matrix in the definition of the polyhedron $\bar{X}(\varepsilon)$ has for $\varepsilon = 0.1$ the form

$$A=\left[\begin{array}{cccccccccccc} 1 & 1 & 1 & -1 & -.1 & -.1 & -1 & -.1 & -.1 & -1 & -.1 & -.1 \\ -1 & 0 & 0 & 1 & 1 & 1 & 0 & -.9 & 0 & 0 & -.9 & 0 \\ 0 & -1 & 0 & 0 & -.9 & 0 & 1 & 1 & 1 & 0 & 0 & -.9 \\ 0 & 0 & -1 & 0 & 0 & -.9 & 0 & 0 & -.9 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right].$$

Define $s := \sum_{a\in\mathcal{A}(i)} x_{ia}$ and $\hat{x}_{ik} = x_{ik}/s$.
We wish $\hat{x}_{ik}$ to be **binary**.

Find an HC in the graph, i.e., for a given node i choose **one** of the outgoing arcs $(i,k) \in \mathcal{A}(i)$.

Two possibilities:

1. Define **binary** variables: $\hat{x}_{ik},\ k \in \mathcal{A}(i)$.
$\hat{x}_{ik}$ is equal to 1 iff arc (i,k) is chosen.
Choose one element from the set:

$$\sum_{k\in\mathcal{A}(i)} \hat{x}_{ik} = 1.$$

2. Define **continuous** variables: $0 \le \hat{x}_{ik} \le 1$ for $k \in \mathcal{A}(i)$ and

$$\begin{array}{ll} \min & q_i(x) = (\sum_k \hat{x}_{ik})^2 - \sum_k \hat{x}_{ik}^2 \\ \text{s.t.} & \sum_k \hat{x}_{ik} = 1. \end{array}$$

Observe that

$$q_i(x) = \sum_{k\neq l} \hat{x}_{ik}\hat{x}_{il} \ge 0$$

and it is equal to zero iff at most one of the variables $\hat{x}_{ik}$ is nonzero.

From LP to QP

QP problem

$$\begin{array}{ll} \min & c^T x + \frac{1}{2} x^T Q x \\ \text{s.t.} & Ax = b, \\ & x \ge 0. \end{array}$$

First order conditions (for barrier problem)

$$\begin{array}{rcl} Ax & = & b, \\ A^T y + s - Qx & = & c, \\ XSe & = & \mu e. \end{array}$$

Newton direction

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

where

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

Augmented system

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

Conclusion:
QP is a natural extension of LP.

IPMs: LP vs QP

Augmented system in **LP**

$$\begin{bmatrix} -\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1}\xi_\mu \\ \xi_p \end{bmatrix}.$$

Eliminate Δx from the first equation and get normal equations

$$(A\Theta A^T)\Delta y = g.$$

Augmented system in **QP**

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

Eliminate Δx from the first equation and get normal equations

$$(A(Q + \Theta^{-1})^{-1}A^T)\Delta y = g.$$

One can use normal equations in LP, but not in QP. Normal equations in QP may become almost completely dense even for sparse matrices A and Q. Thus, in QP, usually the indefinite augmented system form is used.

Example

$$\begin{bmatrix} 1 & 1 & & & \\ 1 & 2 & 1 & & \\ & 1 & 2 & 1 & \\ & & 1 & 2 & 1 \\ & & & 1 & 2 \end{bmatrix}^{-1} = \left(\begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & 1 & \\ & & & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & 1 & \\ & & & 1 & 1 \\ & & & & 1 \end{bmatrix} \right)^{-1}$$

$$= \begin{bmatrix} 1 & -1 & 1 & -1 & 1 \\ & 1 & -1 & 1 & -1 \\ & & 1 & -1 & 1 \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ 1 & -1 & 1 & & \\ -1 & 1 & -1 & 1 & \\ 1 & -1 & 1 & -1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 5 & -4 & 3 & -2 & 1 \\ -4 & 4 & -3 & 2 & -1 \\ 3 & -3 & 3 & -2 & 1 \\ -2 & 2 & -2 & 2 & -1 \\ 1 & -1 & 1 & -1 & 1 \end{bmatrix}.$$

Conclusion:
the inverse of the sparse matrix may be dense.

IPMs for QP:
Do not explicitly invert the matrix $Q + \Theta^{-1}$ in the matrix $A(Q + \Theta^{-1})^{-1}A^T$.
Use the **augmented system** instead.

Regarding the computations involved, a quadratic program with diagonal matrix $Q = D$:

$$\begin{array}{ll} \text{min} & c^T x + \frac{1}{2} x^T D x \\ \text{s.t.} & Ax = b, \\ & x \geq 0, \end{array}$$

is as easy as a linear program.
Indeed, in this case, the Newton equation system can be reduced to the following normal equation system:

$$(A(D + \Theta^{-1})^{-1} A^T) \Delta y \; = \; g.$$

Since $\tilde{\Theta}^{-1} = D + \Theta^{-1}$ is a diagonal matrix, this system is not more difficult to solve than a usual system arising in LP:

$$(A \Theta A^T) \Delta y \; = \; g.$$

Conclusion:
If you can formulate the QP as a **separable** problem, then it's usually worth a try.

Sparsity Issues

We consider a "subproblem" for node i.

The function $q_i(x)$ can be rewritten as follows

$$q_i(x) = (\sum_k \hat{x}_{ik})^2 - \sum_k \hat{x}_{ik}^2 = x^T Q_i \, x,$$

where

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

Regarding the computations, this is possibly a very demanding form because Q_i is a completely dense matrix.

Let us introduce an additional variable

$$y_i = \sum_k \hat{x}_{ik},$$

and rewrite the problem

$$\begin{array}{ll} \text{min} & q_i(x) = y_i^2 - \sum_k \hat{x}_{ik}^2 \\ \text{s.t.} & \sum_k \hat{x}_{ik} = y_i, \\ & \sum_k \hat{x}_{ik} = 1. \end{array}$$

with a **diagonal** quadratic form.
It is a **separable non-convex QP**.

Quasidefinite Matrices

Symmetric matrix is called **quasidefinite** if

$$K = \begin{bmatrix} -E & A^T \\ A & F \end{bmatrix},$$

where $E \in \mathcal{R}^{n \times n}$ and
$F \in \mathcal{R}^{m \times m}$ are positive definite, and
$A \in \mathcal{R}^{m \times n}$ has full row rank.

Symmetric nonsingular matrix K is *factorizable* if there exists a diagonal matrix D and a unit lower triangular matrix L such that $K = LDL^T$.

The symmetric matrix K is *strongly factorizable* if for any permutation matrix P a factorization $PKP^T = LDL^T$ exists.

Vanderbei (1995) proved that
Symmetric QDFM's are strongly factorizable.

For any quasidefinite matrix
there exists a **Cholesky-like** factorization

$$\bar{H} = LDL^T,$$

where
D is **diagonal** but **not positive definite**:
n negative pivots;
m positive pivots.

Indefinite matrix

$$H = \begin{bmatrix} -Q-\Theta^{-1} & A^T \\ A & 0 \end{bmatrix}.$$

Vanderbei (1995): replace $Ax = b$ with $Ax \le b$

$$H_V = \begin{bmatrix} -\Theta_s^{-1} & 0 & I \\ 0 & -Q-\Theta^{-1} & A^T \\ I & A & 0 \end{bmatrix}$$

and eliminate Θ_s^{-1}

$$K = \begin{bmatrix} -Q-\Theta^{-1} & A^T \\ A & \Theta_s \end{bmatrix}.$$

Saunders (1996):

$$H_S = \begin{bmatrix} -Q-\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} + \begin{bmatrix} -\gamma^2 I_n & 0 \\ 0 & \delta^2 I_m \end{bmatrix},$$

where

$$\gamma\delta \ge \sqrt{\varepsilon} = 10^{-8}.$$

A & G (1999): use **dynamic regularization**

$$\bar{H} = \begin{bmatrix} -\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} + \begin{bmatrix} -R_p & 0 \\ 0 & R_d \end{bmatrix},$$

$R_p \in \mathcal{R}^{n\times n}$ is a *primal* regularization
$R_d \in \mathcal{R}^{m\times m}$ is a *dual* regularization.

Having introduced m additional variables

$$y_i = \sum_k \hat{x}_{ik}, \quad i = 1, 2, ..., m$$

we transform the block-diagonal quadratic form diag$(Q_1, Q_2, ..., Q_m)$ to a diagonal one.

We deal with the quadratic program

$$\begin{array}{ll} \text{min} & c^T x + \frac{1}{2} x^T Q x \\ \text{s.t.} & Ax = b, \\ & x \ge 0, \end{array}$$

in which Q is **diagonal** and **indefinite**.
We solve it with **HOPDM**.

If we find a solution such that $x^T Q x = 0$, then we get a Hamiltonian cycle.

In general, we cannot expect this to happen.

What can we learn from the local solution?

We have implemented a heuristic to eliminate arcs that are not used by the flow x and a simple branching strategy.

Heuristics

Suppose a local solution of the QP has been found but $x^T Q x \ne 0$.

Define $s = \sum_{k \in \mathcal{A}(i)} x_{ik}$ and $\hat{x}_{ik} = x_{ik}/s$.

We would like to have only one arc outgoing the node i, i.e., only one $\hat{x}_{ik} > 0$.

Arc Elimination

If $\hat{x}_{ik} < \delta$, then remove arc (i,k).

Branching

If there is no arc that can be eliminated, then choose a node i and branch on all outgoing arcs. That is, consider $|\mathcal{A}(i)|$ new problems, each corresponding to a graph with only one of arcs from $\mathcal{A}(i)$ left.

Depth-first search

If branching happens we analyse all $|\mathcal{A}(i)|$ children nodes and continue with the one for which the largest number of arcs can be eliminated.

Numerical Results

Knight Tour Problem

Given a $k \times k$ chessboard, find a tour of the Knight to visit each square of the board exactly once.

200 MHz Pentium III PC, Linux.

Problem	Nodes	Arcs	time
chess6	36	160	1.25
chess8	64	336	3.35
chess10	100	576	29.77
chess12	144	880	33.58
chess14	196	1248	456.01
chess20	400	2736	1203.61
chess32	1024	7440	11 hrs

Solution for 6×6 chessboard:

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

Randomly generated problems

200 MHz Pentium III PC, Linux.

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

The approach works well unless the tree grows too large.

The largest trees reached 62 and 59 for chessboard 32×32 and rand10 problems, respectively.

- Embedding HCP into MDP.
- Perturbed MDP $\rightarrow$ unichain MDP:
  - Polyhedral representation
  - Integer LP
  - Non-convex QP
- Heuristics needed:
  - Branching (implemented)
- Medium-scale problems solved
  - Knight Tour Problems
  - Randomly generated problems
- Further research:
  - Decomposition
  - Cuts