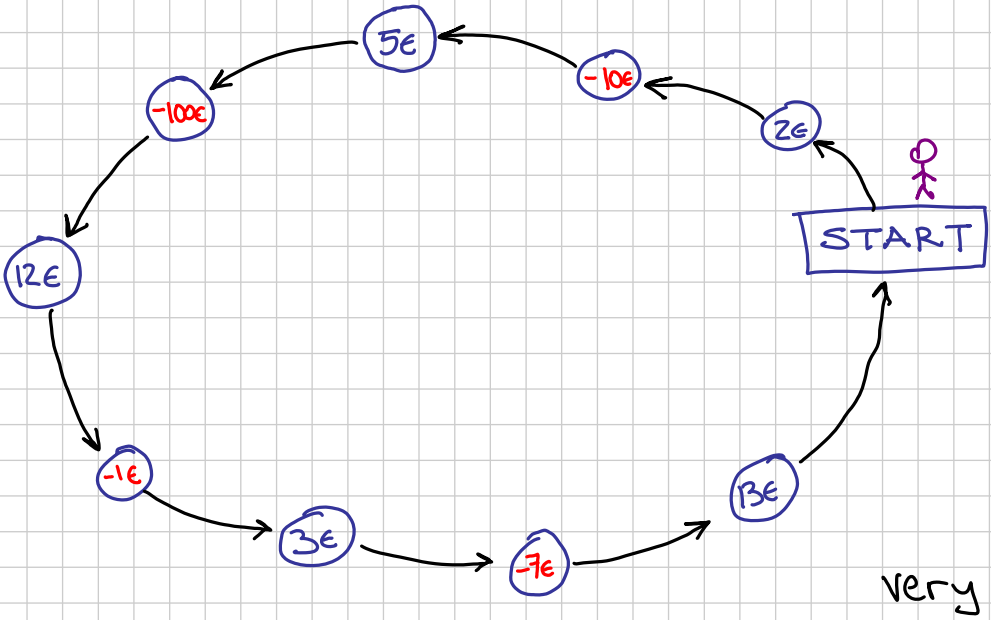
A strongly polynomial algorithm for one-dimensional Markov decision process

Guy Even: Tel-Aviv Univ.

joint work with:

Alexander Zadorojniy: Tel-Aviv Univ

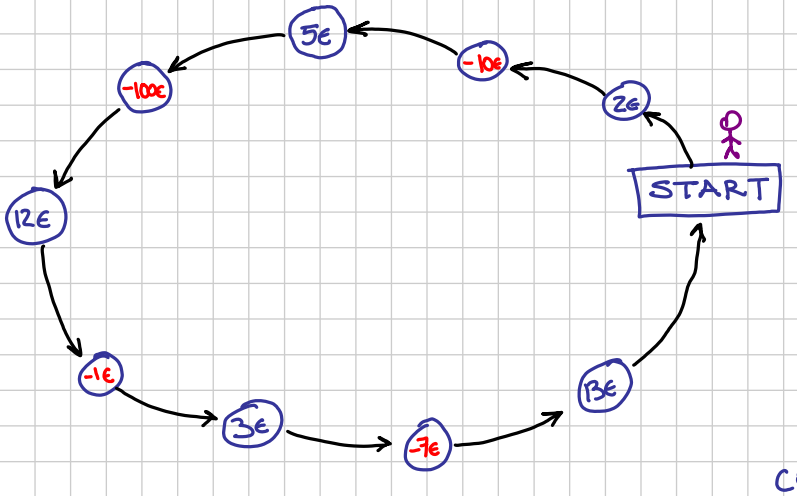
Adam Shwartz: Technion



Rules

- Throw dice
- Move accordingly
- receive / pay

very boring game:
- "easy" to compute expected payoff
- wit does not help.



Variation (Baksheesh)

before throwing dice, player may ask for biased dice.

$$\text{cost}(1,\dots,6) = 0\epsilon$$

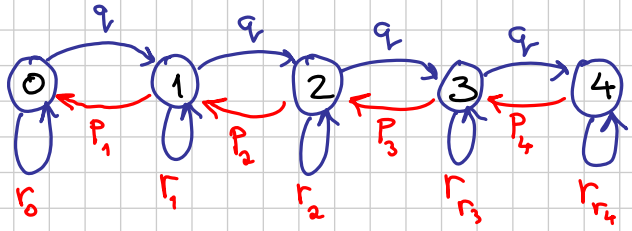
$$\text{cost}(1,2,3) = 2\epsilon$$

$$\text{cost}(4,5,6) = 1\epsilon$$

Q: find strategy that maximizes payoff.
E.g. pick which dice to throw in each state.

A Controlled Queue

* discrete-time M/M/1 queue.
* control
  $u \in \{$ hire extra people, fire people, take vacation $\}$
* arrival rate fixed
  $\forall i: \Pr(i+1 \mid i,u) = q$
* service rate depends on control
  $\forall i,u: \Pr(i-1 \mid i,u) = p_u$

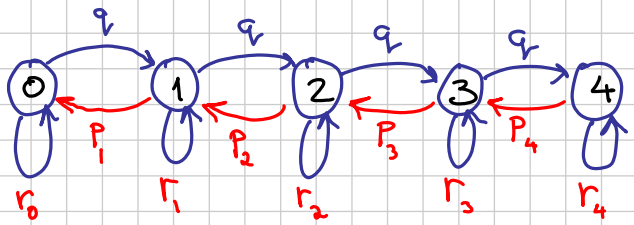


A Controlled Queue – cont

each state-control pair (x,u) has a cost $c(x,u)$.

example:
$c(x,u) = x + \text{cost}(u)$

goal: find strategy that minimizes the expected cost.



Definition: Markov Decision Process (MDP)

X – set of states $|X| = n$ $X = \{0, 1, \ldots, n-1\}$

U – set of actions $|U| = k$ $U = \{0, 1, \ldots, k-1\}$

$P: X \times X \times U \to [0,1]$ – transition probability function

$P(y \mid x, u) \triangleq$ prob. of transition $x \to y$ if action u is chosen in state x

$c: X \times U \to \mathbb{R}$ – immediate cost function, $c(x,u)$ – cost of choosing action u when in state x.

An "overlay" of k Markov chains (one per action).

MDP - definitions

stationary

policy (strategy) $\pi : X \times U \to [0,1]$ that satisfies:

$$\forall x \in X: \quad \sum_{u \in U} \pi(x,u) = 1$$

If $\exists u: \pi(x,u)=1$, then π is [redacted].

If $\forall u: \pi(x,u) \in \{0,1\}$, then π is deterministic.

If π is deterministic in all ~~states~~ but one state $\hat{x}$ and $|\{u \mid \pi(\hat{x},u) > 0\}| = 2$, then π is 1-randomized.

Every policy reduces an MDP to a Markov chain.

MDP - cost models

{ assume time is discrete

x_t - random variable equals state in time t

u_t - random variable equals action in time t

$$E^{\pi}\left(c(x_t,u_t)\right) \triangleq \sum_{x \in X, u \in U} c(x,u) \cdot P^{\pi}\left(x_t = x, u_t = u\right)$$

discounted cost: parameter $0 < \beta < 1$ $\qquad \beta = \frac{1}{1 + \text{interest rate}}$

$$C(\pi) \triangleq \sum_{t=0}^{\infty} \beta^t \cdot E^{\pi}\left(c(x_t,u_t)\right)$$

Expected average cost:

$$C(\pi) \triangleq \lim_{T \to \infty} \frac{1}{T} \cdot \sum_{t=0}^{T-1} E^{\pi}\left[c(x_t,u_t)\right]$$

Optimization Problem

Input: an MDP $\langle X, U, P, c \rangle$

Goal: compute a min-cost policy π^*

$$c(\pi^*) = \min \{ c(\pi) : \pi \text{ is a policy} \}$$

MDP Example: Online Stochastic Steiner Tree

Graph $G = (V, E)$

An online random sequence of subsets of V

$U_0, U_1, \ldots, U_t, \ldots$ (governed by a discrete stochastic process)

Rules: Compute $T_0, T_1, \ldots$ such that $T_t \subseteq E$ connects terminals in U_t.

Buy: pay b_e to add e to tree

Sell: get s_e if you no longer want e

Goal: Compute an optimal strategy for buying/selling edges.

Occupation Measure

Every policy π induces a probability distribution $\rho_\pi : X \times U \to [0,1]$.

discounted cost model:

$$\rho_\pi(x,u) \triangleq (1-\beta) \cdot \sum_{t=1}^{\infty} \beta^{t-1} \cdot P^\pi(x_t = x, u_t = u)$$

expected average cost model:

$$\rho_\pi(x,u) \triangleq \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} P^\pi(x_t = x, u_t = u)$$

occupation measure induces a policy $\pi(x,u) = \dfrac{\rho(x,u)}{\sum_{u' \in U} \rho(x,u')}$.

Unichain Assumption

Assume that, for every policy π, MDP with π is an irreducible Markov chain.

Namely, $\forall \pi \quad \forall x \in X$: $\sum_{u \in U} \rho_\pi(x,u) > 0$.

* Constrained MDP (CMDP)

an MDP + a function $d : X \times U \to \mathbb{R}$ and

a constraint $D(\pi) = \alpha$.

where $D(\pi) \triangleq \sum_{t=1}^{\infty} \beta^{t-1} \cdot E^{\pi}(d(x_t, u_t))$ ← discounted cost model

OR

$$D(\pi) \triangleq \lim_{T \to \infty} \frac{1}{T} \cdot \sum_{t=1}^{T} E^{\pi}[d(x_t, u_t)]$$ ← expected avg. cost model

- may have more than one constraint.
- can be used to model power consumption, avg. delay, etc.

Linear Programming Formulation

As in Markov chains, every policy π induces

a limit prob. distribution $\rho_\pi : X \times U \to [0, 1]$.

ρ_π satisfies 2 types of constraints: ← in exp. avg. cost model

balance constraints

$$\underbrace{\sum_u \rho(x,u)}_{\text{prob. of being in state } x} = \underbrace{\sum_{y,u} P(x|y,u) \cdot \rho(y,u)}_{\text{prob of entering state } x}$$

prob. distribution constraints

$$\begin{cases} \sum_{x,u} \rho(x,u) = 1 \\ \rho \geq 0 \end{cases}$$

Linear Programming Formulation

[D'Epenoux 60, 63
De Ghellinck 60
Manne 60
Derman 62]

discounted cost model:

$$\begin{aligned} \min \quad & c^t \rho \\ \text{s.t} \quad & A\rho = b \\ & \rho \geq 0 \\ & d^t \rho = \alpha \end{aligned}$$

extra constraint

coeff: $c, d \in \mathbb{R}^{nk}$

$A \in M_{n \times nk}$

$b \in \mathbb{R}^n$

vars: $\rho \in \mathbb{R}^{nk}$

where

$$A \triangleq \left[I - P(u_1) \;\middle|\; I - P(u_2) \;\middle|\; \cdots \;\middle|\; I - P(u_k) \right]$$

$$b \triangleq \begin{pmatrix} 1-\beta \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

each $\left[P(u) \right]_{yx} \triangleq P(y \mid x, u)$

in exp. avg. cost model: add an all-ones row to A

Theorem [D'Epenoux]: equiv. MDP $\leftrightarrow$ LP & CMDP(α) $\leftrightarrow$ LP(α)

CMDP(α) feasible $\iff$ LP(α) feasible

π^* opt. policy $\implies$ ρ_{π^*} opt. solution for LP(α)

π_{ρ^*} opt policy $\impliedby$ ρ^* opt. solution for LP(α)

Moreover: cost is preserved: $C(\pi) = c^t \cdot \rho_\pi$.

$\implies$ find opt. policies for MDP & CMDP using linear prog.

other methods: value iteration & policy iteration (only for MDP).

OPEN QUESTION: strongly polynomial algorithm for MDP.

[Ye 2005] strongly poly. interior point alg for discount cost.

Optimal Solutions

Theorem (60's follows from LP formulation + Blackwell)

Every MDP has an optimal deterministic (stationary) policy.

If CMDP(α) is feasible, then it has an optimal policy that is either deterministic or 1-randomized.

proof: (MDP) consider LP.

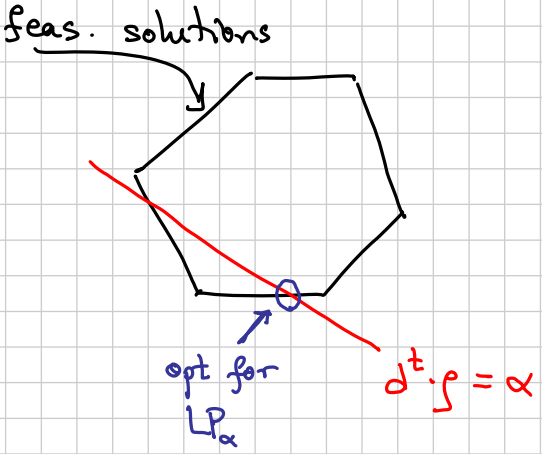
unichain assumption $\Rightarrow$ (at least) one column per state in basis

$\Rightarrow$ $\forall$ basis has exactly one column per state

$\Rightarrow$ $\forall$ basic feasible solution ρ induces a det. policy.

Limit search to det. & 1-rand. policies!

Overview of Algorithm



Scan polytope of LP by adding a new constraint

$$d^t \cdot \rho = \alpha.$$

$\forall \alpha$: Find det. or 1-rand. opt solution $\rho(\alpha)$ for $LP(\alpha)$

Return: $\operatorname{argmin} \{ c^t \cdot \rho(\alpha) \}$

Policy Graph

Vertices: det. policies

Edges: between det. policies that disagree in 1 state.

$\Rightarrow$ For 2 actions ($k=2$), policy graph $\approx$ hypercube

every 1-rand. policy π is a convex combination of 2 neighboring det. policies π^0 & π'.

$$\pi = \lambda \cdot \pi^0 + (1-\lambda) \cdot \pi^1$$

In this case we say: π lies on the edge (π^0, π')

Structure: Optimal policies of CMDP(α)

$\Gamma \triangleq \{\pi \mid \pi$ is a det. or 1-rand. policy $\}$

$\Gamma^* \triangleq \{\pi \in \Gamma \mid \exists \alpha : \pi$ is an opt. policy for CMDP(α)$\}$

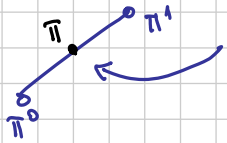
NEW

Theorem: Γ^* is a path in the policy graph

previous (ZS):

: 1) $\forall \pi$ det. : $\pi \in \Gamma^* \iff \pi$ is a vertex in P

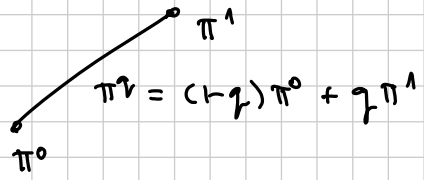
2) $\forall \pi$ 1-rand : $\pi \in \Gamma^* \iff$



whole edge is in P

proof sketch

* Γ^* is the union of vertices & edges.

* 

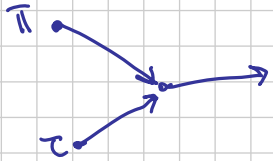
$D(q) \triangleq D(\pi^q)$

$C(q) \triangleq C(\pi^q)$

Claim: $D(q)$, $C(q)$ are Möbius transformations

$$D(q) = \frac{aq+b}{eq+f}. \Rightarrow \text{monotone!}$$

* direct edges in increasing $D(q)$ direction



either π or τ not [redacted]

uniqueness $\Rightarrow$ in-degree, out-degree $= 1$.

Uniqueness Assumption

Consider an MDP $= \langle X, U, P, c \rangle$ and a function $d: X \times U \to \mathbb{R}$.

Uniqueness assumption (for MDP & d): For every α, if CMDP(α) has an optimal det. policy, then this is the only opt. policy in Γ^* for CMDP(α).

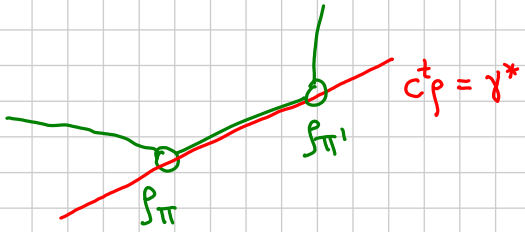
Can be obtained whp by random perturbation of c or d.

If does not hold:

$\exists\ \pi$ det. & optimal

$\exists\ \pi'$ det / 1-rand neighbor

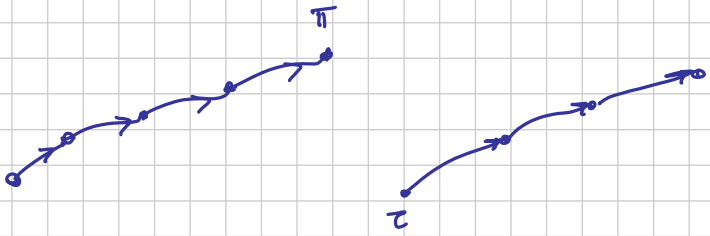
$c^t \cdot \rho_\pi = c^t \rho_{\pi'} = \gamma^*$



proof sketch (cont)

* Γ^* is a union of paths.

* Γ^* is connected $\Rightarrow$ path, as required.



$D(\pi) < D(\tau)$ (otherwise τ not unique).

but $\pi^q = (1-q)\pi + q\tau$ cont. changes from $D(\pi)$ to $D(\tau)$.

From Structure to Algorithm

Γ^* is a path in the policy graph

$\Rightarrow$ det. policies in Γ^* lie on a path in policy graph.

One of these policies is opt for MDP:

$$\pi_{opt} \triangleq \operatorname{argmin} \{ C(\pi) \mid \pi \text{ det. } \& \ \pi \in \Gamma^* \}$$

Alg: compute path of opt. det. policies and return one with cheapest cost $C(\pi)$.

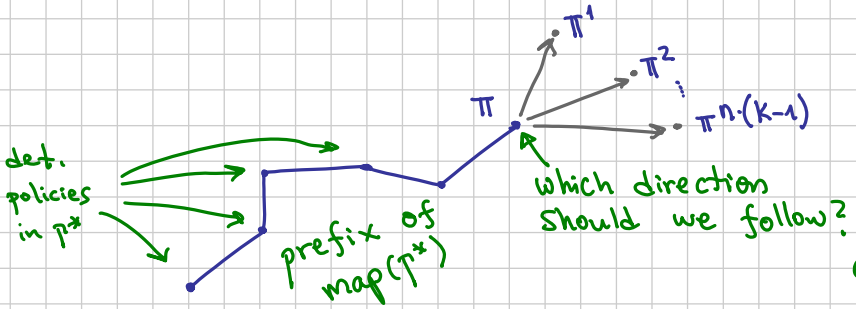
How? mimic gradient descent alg.

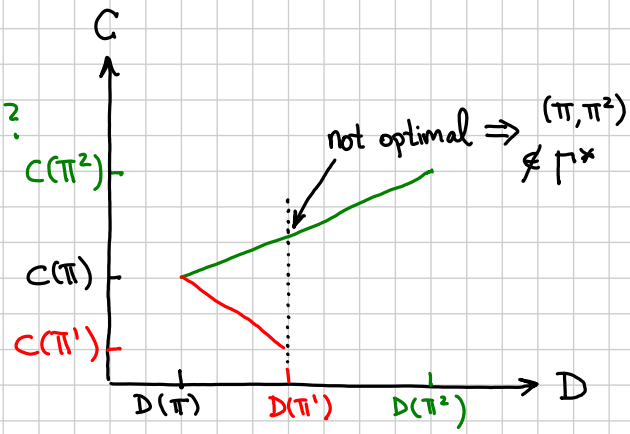
Computing map (Γ^*)

Structural Lemmas:

1) $D(\pi)$ is strictly monotone along Γ^*

2) $C(\pi)$ is linear in $D(\pi)$ along edges





⇒ Rule: follow edge (π,π^j) that minimizes ratio $\frac{C(\pi^j)-C(\pi)}{D(\pi^j)-D(\pi)}$

provided that $D(\pi^j) > D(\pi)$.

Alg Issues

1) Start & end of alg.

2) How to compute $\frac{C(\pi^j)-C(\pi)}{D(\pi^j)-D(\pi)}$

3) Running time

Starting point

How do find first endpoint of path $map(\tau^*)$?

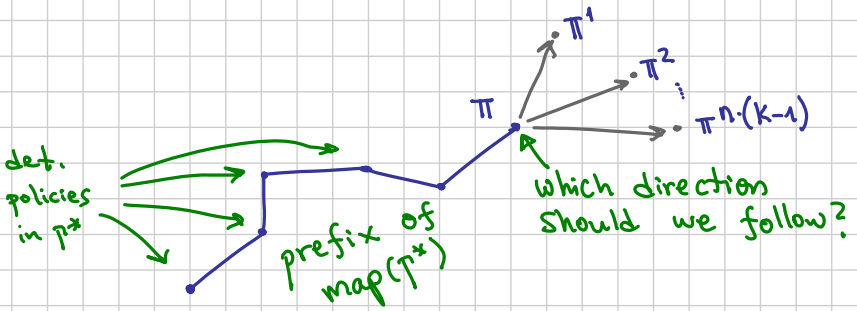
Define function $d: X \times U$ so that it is trivial !

$$d(x,u) = \begin{cases} 0 & \text{if } u=0 \\ 1 & \text{otherwise} \end{cases}$$

if $\pi(x) \equiv 0$, then $D(\pi) = 0$, and π is opt for $cMDP(0)$.

Moreover, $D(\pi)$ is minimum, hence π is the first endpoint.

Stopping the alg



path continues only if $\exists j: D(\pi^j) > D(\pi)$.

Computing next node

Augmentation by one edge requires computing

$$\min \left\{ \frac{C(\pi^j) - C(\pi)}{D(\pi^j) - D(\pi)} \;\middle|\; \text{where } D(\pi^j) > D(\pi) \right\}$$

Computation of $C(\pi)$ requires inverting a basis matrix....

π is a bet. policy. f_π is a bfs.

$$A f_\pi = b$$

$$B_\pi \cdot f_\pi = b \qquad (B_\pi \text{ basis matrix})$$

$$f_\pi = B_\pi^{-1} \cdot b$$

$$C(\pi) = c^t \cdot f_\pi .$$

← Every step is polynomial

Running Time

Number of steps = length of path.

Do not know how to bound

$\Rightarrow$

possibly non-poly. running time....

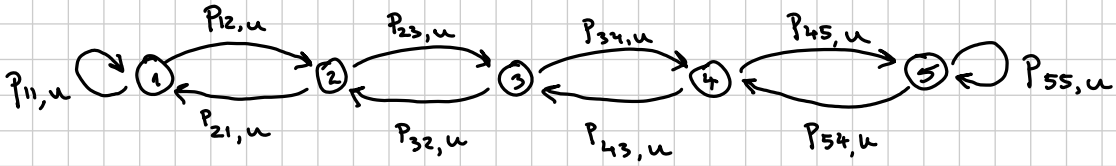
Getting poly. running time

* assume order $\preceq_x$ between actions ($\forall$ state x).
* def. part. order $\preceq$ over det. policies:

$$\pi \preceq \tau \iff \forall x: \quad \pi(x) \preceq_x \tau(x).$$

* length of longest chain $\leq nk$.
* Prove that policies along path γ^* form a chain.
  * M/M/1 controlled queue using a "coupling property"

Guaranteeing a short path: coupling property



action u affects transition probability.

example: two actions $\{0,1\}$ & $p_{i,i-1,1} > p_{i,i-1,0}$.

$\Rightarrow$ If π & τ are neighboring det. policies s.t.

$$\pi(3) = 0 \quad \& \quad \tau(3) = 1$$

then

$$\forall u: \quad \rho_\pi(1,u) < \rho_\tau(1,u)$$

$$\rho_\pi(2,u) < \rho_\tau(2,u)$$

τ "pushes" to the left more than π in state 3

Coupling Property

Def: Let π denote a det. policy.

Let $i \in X$, $j \in U$.

Then $\pi^{i,j}$ denotes the neighboring det. policy such that

$$\pi^{i,j}(x) \triangleq \begin{cases} \pi(x) & \text{if } x \neq i \\ j & \text{if } x = i \end{cases}$$

Assume $X = \{0, 1, \ldots, n-1\}$.

Let $\leq_i$ denote a linear order over U, for each $i \in X$.

Def: The coupling property holds if

$\forall$ det. π $\forall i \in X$ $\forall j \in U$:

$$\pi(i) \leq_i j \implies \forall x < i : \rho_\pi(x,u) \leq \rho_{\pi^{i,j}}(x,u)$$

How does the coupling property help?

Assign $d : X \times U \to \mathbb{R}$ so that if $j \leq_i \pi(i)$ then

$D(\pi^{i,j}) < D(\pi)$.

$\implies$ $\pi^{i,j}$ cannot appear after π along path.

$\implies$ length of path $\leq nk$.

Summary

- new type of alg. for finding optimal policy for MDP.
- For general MDP - not proven to be poly.
- With additional property (coupling): strongly poly. running time: $O((Kn)^2 \cdot M(n))$.

Further work

- extend results to CMDPs.

Open

- find an MDP for which "path" is long.
- prove poly-time ....