A New Algorithm for Mean Payoff Games

Yury Lifshits[1] and Dmitri Pavlov[2]

[1]Steklov Institute of Mathematics at St.Petersburg,
yura@logic.pdmi.ras.ru
[2]St.Petersburg Institute of Fine Mechanics and Optics,
pavlov@rain.ifmo.ru

EPIT 2006

Outline of the Talk

1. Rules of Mean Payoff Games
2. Computing Winning Strategies in Mean Payoff Games
3. Conclusions

Outline of the Talk

1. Rules of Mean Payoff Games
2. Computing Winning Strategies in Mean Payoff Games
3. Conclusions

1.1. Rules of mean payoff games

Input for a **mean payoff game**:

- Weighted directed graph (integer weights)
- Graph does not contain simple cycles with zero sum
- Vertices are divided into disjoint sets A and B
- The starting vertex

1.1. Rules of mean payoff games

Rules for mean payoff games:

- Two players: Alice and Bob
- Players move the token over arcs
- Game starts from the starting vertex and it is infinite
- Alice plays from vertices of A, Bob from these of B
- Alice wins if the sum of already passed arcs goes to $+infty$
- Bob wins if the sum of already passed arcs goes to $-infty$

1.1. Rules of mean payoff games

Rules for mean payoff games:

- Two players: Alice and Bob
- Players move the token over arcs
- Game starts from the starting vertex and it is infinite
- Alice plays from vertices of A, Bob from these of B
- Alice wins if the sum of already passed arcs goes to $+infty$
- Bob wins if the sum of already passed arcs goes to $-infty$

Computational task: given a game graph with an A, B decomposition and a starting vertex to determine the winner (and find the winning strategy)

1.2. MPG is Very Challenging

Mean Payoff Game Problem belongs to NP$\cap$co-NP
Mean Payoff Games have applications in μ-calculus verification

Known algorithms:

- Naive algorithm, n^n in the worst case
- Strategy improvement by Jurdziński, n^n in the worst case
- Linear programming based algorithm by Björklund, Sandberg and Vorobyov, $2^{\sqrt{n}}$ expected time, n^n in the worst case

1.2. MPG is Very Challenging

Mean Payoff Game Problem belongs to NP∩co-NP
Mean Payoff Games have applications in μ-calculus verification

Known algorithms:

- Naive algorithm, n^n in the worst case
- Strategy improvement by Jurdziński, n^n in the worst case
- Linear programming based algorithm by Björklund, Sandberg and Vorobyov, $2^{\sqrt{n}}$ expected time, n^n in the worst case

Our result: $O^*(2^n)$ deterministic algorithm

Outline of the Talk

1. Rules of Mean Payoff Games
2. Computing Winning Strategies in Mean Payoff Games
3. Conclusions

2.0. Our Small Plan

1. Define potentials
2. Prove their properties
3. Compute potentials
4. Derive winners and strategies from potentials

2.1. Definition of Potentials

"Money explanation": Let's assume that game started from vertex u with $X\$$, every positive arc increase the account, every negative decrease.

2.1. Definition of Potentials

"Money explanation": Let's assume that game started from vertex u with X\$, every positive arc increase the account, every negative decrease.

The **Alice's potential of** u is the minimal X such that Alice can enforce nonnegative balance through all the game

2.1. Definition of Potentials

"Money explanation": Let's assume that game started from vertex u with $X\$$, every positive arc increase the account, every negative decrease.

The **Alice's potential of** u is the minimal X such that Alice can enforce nonnegative balance through all the game

The **Bob's potential of** u is the minimal $-X$ such that Bob can enforce nonpositive balance through all the game

2.2. Properties of Potentials

The vertex is a **endpoint**, if the only outgoing arc is the self-loop

Introduce an endpoint means take some vertex and replace all outgoing edges by either $+1$ or -1 self-loop

2.2. Properties of Potentials

The vertex is a **endpoint**, if the only outgoing arc is the self-loop

Introduce an endpoint means take some vertex and replace all outgoing edges by either $+1$ or -1 self-loop

1. Every game graph with an endpoint has a **non-significant** vertex
2. For every graph we can introduce an endpoint without changing potentials
3. We can check "are these numbers true potentials?" in polynomial time

2.3. Computing Potentials

Were are going to compute potentials for

- Initial game graph G
- All subgraphs of G
- All subgraphs with one introduced endpoint

Totally for about $(2n+1)2^n$ graphs!

Method: dynamic programming from smaller graphs to bigger ones

2.3. Computing Potentials cont.

One step of dynamic programming:

- For graphs **with endpoint**:
  - Through one vertex away
  - Take the rest potentials from already computed subgraph
  - Put the deleted vertex back and check for current graph
  - Must work by property 1
- For graph **without endpoint**:
  - Just check potentials for all versions with introduced endpoint
  - Must work by property 2

2.4. Getting Strategies from Potentials

Lemma 1: Exactly one potential is finite for every vertex. Alice wins iff Alice's potential is finite on the starting vertex

2.4. Getting Strategies from Potentials

Lemma 1: Exactly one potential is finite for every vertex. Alice wins iff Alice's potential is finite on the starting vertex

Lemma 2: Strategy that minimize the "weight of the edge - difference of potentials" is the winning one.

Outline of the Talk

Summary

Main points:

- Computational problem of Mean Payoff Games/ given a game graph and a starting vertex do determine the winner

Summary

Main points:

- Computational problem of Mean Payoff Games/ given a game graph and a starting vertex do determine the winner
- Idea of new algorithm: compute potentials via dynamic programming over all subgraphs

Summary

Main points:

- Computational problem of Mean Payoff Games/ given a game graph and a starting vertex do determine the winner
- Idea of new algorithm: compute potentials via dynamic programming over all subgraphs
- Main trick: existence of a non-significant vertex

Summary

Main points:

- Computational problem of Mean Payoff Games/ given a game graph and a starting vertex do determine the winner
- Idea of new algorithm: compute potentials via dynamic programming over all subgraphs
- Main trick: existence of a non-significant vertex

Open Problem:

- Solve MPG in polynomial time!!!

Last Slide

Yury Lifshits
yura@logic.pdmi.ras.ru
http://logic.pdmi.ras.ru/~yura/

Yury Lifshits, Dmitri Pavlov
Fast Exponential Deterministic Algorithm for Mean Payoff Games.
Submitted, 2006.

Last Slide

Yury Lifshits
yura@logic.pdmi.ras.ru
http://logic.pdmi.ras.ru/~yura/

Yury Lifshits, Dmitri Pavlov
Fast Exponential Deterministic Algorithm for Mean Payoff Games.
Submitted, 2006.

Thanks for attention.

Last Slide

Yury Lifshits
yura@logic.pdmi.ras.ru
http://logic.pdmi.ras.ru/~yura/

Yury Lifshits, Dmitri Pavlov
Fast Exponential Deterministic Algorithm for Mean Payoff Games.
Submitted, 2006.

Thanks for attention. **Questions?**