A New Algorithm for Mean Payoff Games

Yury Lifshits¹ and Dmitri Pavlov²

1Steklov Institute of Mathematics at St.Petersburg, yura@logic.pdmi.ras.ru
2St.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

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 is infinite
- 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 $\mu$-calculus verification

Known algorithms:
- Naive algorithm, $n^6$ in the worst case
- Strategy improvement by Jurdziński, $n^6$ in the worst case
- Linear programming based algorithm by Björklund, Sandberg and Vorobyov, $2^{\sqrt{n}}$ expected time, $n^6$ in the worst case

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

2.0. Our Small Plan

Define potentials
Prove their properties
Compute potentials
Derive winners and strategies from potentials

Outline of the Talk

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

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

2.3. Computing Potentials

We are going to compute potentials for
- Initial game graph \( G \)
- All subgraphs of \( G \)
- All subgraphs with one introduced endpoint

Totally for about \( (2^n + 1)^2 \) 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.

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

Outline of the Talk

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

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.

Thanks for attention. Questions?