Radovan Červený

cervera3@fit.cvut.cz

presented:

On the Shortest Path Game

Andreas Darmann, Ulrich Pferschy, Joachim Schauer

http://www.sciencedirect.com/science/article/pii/S0166218X15003959

Definitions

Shortest Path Game

- game setting: (un)directed graph $G = (V, E)$; edges have non-negative cost $c(e) : E \to R_0^+$; two designated vertices $s, t \in V(G)$
- two players A, B start in s; players move together along the edges and take turns in selecting the next vertex to be visited; the deciding player pays the cost of the edge; total costs payed by players are given by $C(A)$, $C(B)$
- A starts i.e. A has first decision in s, game ends the first time A and B step on t

⇒ each player selfishly minimizes his total cost payed

Additional rules

(R1) no player can select an edge which does not permit a path to vertex t

(R2) the players cannot select an edge which implies necessarily a cycle of even length

⇒ gives us *perfect information finite game in extensive form*

- *subgame perfect equilibrium (spe)* is a strategy that is a Nash equilibrium for every subgame of the original game
- *spe-path* is a unique path from s to t in G corresponding to the unique subgame perfect equilibrium

Problems

Shortest Path Game

Input: edge-weighted graph $G = (V, E)$, vertices s, t, values $C_A, C_B \in R_0^+$

Decide: does spe-path yield costs $c(A) \leq C_A$ and $c(B) \leq C_B$?

Quantified 3-SAT

Input: set $X = x_1, \ldots, x_n$ of variables, quantified formula $F = (\exists x_1)(\forall x_2)(\exists x_3)\ldots(\forall x_n)\phi(x_1, \ldots, x_n)$ where ϕ is a 3-CNF formula over X

Decide: is F true?

Theorems

Theorem 1. Shortest Path Game is *PSPACE-complete* even for bipartite directed graphs.

Theorem 2. The spe-path of Shortest Path Game on DAGs can be computed in $O(|E|)$ time.

Theorem 3. Shortest Path Game on undirected graphs is *PSPACE-complete* even for bipartite graphs.

Theorem 4. The spe-path of Shortest Path Game on undirected cactus graphs can be computed in $O(|V|^2)$ time.

Cactus algorithm

Let $G = (V, E)$ is a undirected *cactus graph* – each edge is part of at most one simple cycle.

1. Recognize *connection strip* G' in G
2. Recognize cycles in $G \setminus G'$ and determine order in which we need to preprocess them.
3. Preprocess $G \setminus G'$ in a bottom-up fashion computing *swap options* to be used in G'
4. Compute spe-path in G' taking swap options into account.

Let $d_p^\pm(i)$ be general distance d from vertex i to a fixed vertex, where $p \in \{d, f\}$ denotes *decider*, *follower* role and $\pm \in \{+, -\}$. If $\pm = +$ player deciding in i is the same as player deciding in $v_0 = 0$, otherwise $\pm = -$.

Step 3 dynamic programming arrays

- $tc_p^\pm(i) :=$ min. cost to move from i back to 0 if a turn around is possible.
- $rc_p^\pm(i) :=$ min. cost to move from i back to 0 with no turn possible and the path goes around the cycle.