Piotr Danilewski

Algorithms in Parity Games

1 Summary

Parity Game is a game played by two players on a graph, using a token or a pawn. Its rules are relatively easy, yet it is general enough so that various other problems can be reduced to parity games, for example - a minimal/maximal function fixpoint computation tasks.

First I will define formally what parity game is. Later I will show some basic facts about strategies which can be used in such game. Then I will describe an algorithm which can be used for finding optimal strategies for such game.

2 Parity Game

Parity game is played by two players on a finite graph $G = (V, E)$. Priority $p : V \longrightarrow \mathbb{N}$ is assigned to each vertex, represented by a natural number. For simplicity we assume that there are no double edges or dead ends in the graph. Let c denote maximal priority assigned to a vertex. Algorithm complexity will highly depend on this value.

In the game a token is given, initially over one of the vertices v_0. Over time the token will change its position by moving along the edges. Vertices are partitioned into two sets: V_A and V_B which are under control by players A and B respectively, that is that player may choose an edge and pefrom a single step with the token.

By $\pi = (v_0, v_1, v_2, ...)$ we define a string of visited vertices during an infinite play performed by the players. Let P denote value of minimal priority that repeated itself infinitely many times during the play π. If P is even then player B wins; otherwise player A wins.

3 Various facts of parity games

Given the game we want to know how and when we can win it. We assume that our opponent plays the best of his possibility and we will not count on mistakes he might make.

If the token is at some given vertex v and we know how to play in order to win from that position, we say that we have a winning strategy for v. Note that the strategy will not change whether v is an initial vertex or we reached it after finite number of steps. If we have winning strategy when v is initial state, we can play in exactly the same way after reaching v in some steps. Clearly, we do not depend our strategy on the previous path. Such strategy is called a *memoryless strategy* and can be represented by functions: $s_A : V_A \longrightarrow V$, $s_B : V_B \longrightarrow V$ - pointing at a successor which we pick, once we have such opportunity.

We call graph G_A a strategy-graph for player A if we take initial graph G and limit edges outgoing from V_A to only those chosen by s_A. Similarly we define G_B. Winning strategy for player A, given vertex $v \in V$ is equivalent that all reachable cycles in G_A from v are odd, that is - the minimal priority in the cycle is odd. Similarly winning strategy for player B, given vertex $v \in V$ is equivalent to even cycles.

There exists partition of V into winning sets W_A, W_B. There are no vertices where no one can win or both players can win.

4 Algorithms for parity game

In general there are two ways to approach the problem of finding optimal strategies. First one is to enrich the graph with some additional information and then deduct the desired strategy. Other approach is to define some strategies for players and then improve them until no improvement can be done. In practise the latter seems to be efficient but little is known about its complexity. Here I will focus on the first approach which has stronger theoretical background.

Naive approach

Consider all strategies s_A. For each we consider all possible counter-strategies of the opponent s_B and see how badly we fail with s_A. Finally we pick s_A which happen to be the least bad.

Although the algorithm is obviously correct, its complexity is horrible: $O\left(\prod_{v \in V} \deg v\right)$, which in case of full graph is equal to $O\left(|V|^{|V|}\right)$.

In the following sections we will focus on one of the fastest algorithms we know so far - Jurdzinski's algorithm.

5 Parity Progress Measures

Parity Progress Measure is an additional information assigned to a graph, which helps us find the strategies.

First, we store additional tuple $\mathbb{N}^{c+1}$ at each vertex ($\mathbb{N}_x := \{0, 1, ..., x\}$). We define comparison on the tuples: $<_i, \leq_i, =_i, \geq_i, >_i$ which are lexical operators, but take only $i+1$ first elements into account.

Function $\rho : V \longrightarrow \mathbb{N}^{c+1}$ is parity progress measure if $\forall (v, w) \in E : \rho(v) \geq_{p(v)} \rho(w)$ and inequality is strict if $p(v)$ is odd. Thus following the edge reduces ρ on first $p(v)+1$ positions, but there can be anything on the other positions.

If there is such function for given parity graph G then all cycles are even. Otherwise if minimal priority i is odd at vertex v_1 we have a cycle of inequalities:

$$\rho(v_1) >_i \rho(v_2) \geq_i \rho(v_3) ... \geq_i \rho(v_1)$$

Since first inequality is strict it cannot be true.

Our codomain is infinite, we want to reduce it. Let $V_i := p^{-1}(i)$ is a set of all vertices with priority i. We limit the codomain of parity progress measure to a set:

$$M_G = \mathbb{N}_0 \times \mathbb{N}_{|V_1|} \times \mathbb{N}_0 \times \mathbb{N}_{|V_3|} \times \mathbb{N}_0 \times ... \times \mathbb{N}_{|V_c|}$$

(or $\mathbb{N}_0$ at the end if c is even).

We want to proof that such restriction will not affect the existence of the parity progress measure. If all cycles in a parity graph G are even then we can still find $\rho : V \longrightarrow M_G$. Additionally we want that if $p(v)$ is odd then $\rho(v) >_{p(v)} (0, 0, ..., 0)$.

We will prove it by induction over the number of vertices.

Initial step - one-vertex graph - is trivial.

In second step of the induction we consider three cases:

- If $V_0 = \emptyset \wedge V_1 = \emptyset$ without loss of generality we can reduce all priorities by 2.
- $V_0 \neq \emptyset$. By induction we know that reduced graph $V \setminus V_0$ has function ρ and no odd vertices have assigned a value of $(0, 0, ..., 0)$. We can easily apply $\rho(v) = (0, 0, ..., 0)$ for all $v \in V_0$.
- $V_0 = \emptyset \wedge V_1 \neq \emptyset$
  Let R_1 denote a set of reachable vertices from any element of V_1 in at least 1 step. Neither of vertices of set V_1 may lie on a cycle, therefore we can sort them in topological order and the first vertex cannot be reached from any of set V_1 including itself. Therefore $V_1 \setminus R_1$ must be nonempty.

  $G^R := G \cap R_1$, $G^{NR} := G \setminus R_1$ are nonempty. By induction there are parity progress measures ρ^R and ρ^{NR} for these graphs.

  $$\rho := \rho_R \cup \left(\rho_{NR} + \left(0, \left|V_1^R\right|, 0, \left|V_3^R\right|, ..., \left|V_c^R\right|\right)\right)$$

  In other words: The G^R subgraph of G remains the same. Parity progress measure for G^{NR} however is increased by the maximal value of G^R. Since in the original G^{NR} no odd values had $\rho(v) = (0, 0, ..., 0)$ then all inequalities between G^R and G^{NR} subgraphs will conform the parity progress measures constraints.

  Maximal value at odd position i of ρ^{NR} is, by induction $\left|V_i^{NR}\right|$ therefore value of ρ at the position i cannot exceed $\left|V_i^{NR}\right| + \left|V_i^R\right| = |V_i|$.

6 Game Parity Progress Measure

So far we assumed that are only even cycles and we can find the parity progress measure. Now we will extend our codomain by another, highest element T, which will intuitively mean that for given vertex we cannot compute it. $M_G^T = M_G \cup T$.

We now define a small progress function $Prog(\rho, n, m)$ as the least $m \in M_G^T : m \geq_{p(v)} \rho(w)$ and if $p(v)$ is odd then either inequality is strict or $m = \rho(w) = T$. In other words - $Prog(\rho, n, m)$ is the minimal value at vertex v so that edge (v, w) is valid according to definition of parity progress measure and current function ρ.

We define Game Parity Progress Measure as a function $\rho : V \longrightarrow M_G^T$ if for all v we have:

- if $v \in V_A : \rho(v) \geq_{p(v)} Prog(\rho, v, w)$ for all $(v, w) \in E$ - no matter what player A chooses, it will be still an even cycle.
- if $v \in V_B : \rho(v) \geq_{p(v)} Prog(\rho, v, w)$ for some $(v, w) \in E$ - player B can choose route to an even cycle.

Strategy for player B $s_\rho : V_B \longrightarrow V$ is to pick successor which minimises the function ρ.

If $\rho(v)$ is not T we can reach some even cycle and successor will be lower than T as well; $v \in W_B$. If $\rho(v) = T$ then we are doomed; player A wins no matter what we do, $v \in W_A$.

7 Algorithm for finding minimal Game Parity Progress Measure

Although $\rho(v) = T \forall v \in V$ is a valid function, we are looking for *minimal* which will maximise the W_B set.

For all game parity progress measure functions f, g we define $f \sqsubseteq g$ if $\forall v \in V : f(v) \leq g(v)$. Since it is a complete lattice and it is finite, we know there exists the least element, which we will try to find in our algorithm.

At the beginning we start with value $(0, 0, ..., 0)$ in all vertices. Most likely it will not be a valid game parity progress measure and some values will have to be increased. Therefore we define a lifting function which will solve this problem:

$$Lift(\rho, v)(u) := \begin{cases} \rho(u) \Leftarrow u \neq v \\ \max\left(\rho(v), \min_{(v,w)\in E} Prog(\rho, v, w)\right) \Leftarrow u = v \in V_B \\ \max\left(\rho(v), \max_{(v,w)\in E} Prog(\rho, v, w)\right) \Leftarrow u = v \in V_A \end{cases}$$

Lift operation is monotonic. Once we reach fix point: $\forall v \in V : \rho = Lift(\rho, v)$, the function ρ is valid Game Parity Progress Measure, and it is the minimal function with that property.

Strategy for player B is defined as picking the minimal vertex from those reachable along a single edge from given vertex v.

Space usage is $O(c\,|V|)$ - we have to remember GPPM tuples in each vertex of the graph

Time complexity is harder to compute. Lift operation for given v can be implemented in time $O(c \deg v)$ There can be at most $|M_G|$ lifts for given vertex v, thus whole algorithm will work in $O(\sum_{v\in V} |M_G|\, c \deg v) = O\left(c\,|E|\,|M_G|\right)$. Where $\deg v$ denotes number of outgoing edges.

$$|M_G| = \prod_{i=1}^{\lceil \frac{c}{2} \rceil} (|V_{2i-1}| + 1) < \left(\frac{\sum_{i=1}^{\lceil \frac{c}{2} \rceil} (|V_{2i-1}| + 1)}{\lceil \frac{c}{2} \rceil} \right)^{\lceil \frac{c}{2} \rceil} \leq \left(\frac{|V|}{\lceil \frac{c}{2} \rceil} \right)^{\lceil \frac{c}{2} \rceil}$$

Finally

$$O\left(c\,|E| \left(\frac{|V|}{\lceil \frac{c}{2} \rceil} \right)^{\lceil \frac{c}{2} \rceil} \right)$$

Other algorithms

Of course it is not the only algorithm. There are also noticeable works of McNaughton and its further upgrade developed by Sven. Unfortunately I have to little time to describe all those algorithms in detail.

Complexity of the problem

The problem of finding strategy in Parity Games is a NP ∩ co-NP problem. What is more, it is UP ∩ co-UP problem, meaning there exists non deterministic Turing machine solving the problem in polynomial time, but for given input word there exists at most one run accepting it.

Currently, all known algorithms have exponential time complexity. Yet it is not believed to be an NP-complete problem and strong research is being made to find an algorithm which would solve it in polynomial time.