Variant I Pseudocode: Adam Can play any strategy

Karan Muvvala

April 2020

1 Introduction

Algorithm 1 computes the Reg of a Game G for a given value of b and a payoff function Val. The construction of $\hat{G}$ depends on computing W' which is primarily computing $cVal$ for each edge $\{e = (u, v) \in G \mid \text{u} \in V_\exists\}$ such that Eve plays an alternate strategy σ' and τ for Adam that maximizes the payoff for any of the alternative strategy σ' except the original strategy σ.

We then use values($b \in W'$) to construct G^b with the new weight function $\hat{w} = w(e) - b$ to construct $\hat{G}$. On this game it is sufficient to play a memoryless/positional strategies for either players to ensure an antagonistic value of at least(resp. at most) $aVal(\hat{G})$. From **Section 3 Claim 1.** we conclude that $Reg(G) = -aVal(\hat{G})$. Algorithm 4 initially computes the memoryless strategy for Eve and Adam where Eve looks for the edge with the highest weight and Adam plays adversarially looking for the edge with the minimum weight, to compute the $aVal$ of the game $\hat{G}$. We then can compute σ and τ for the game G as there is an one-to-one correspondence with strategies $\hat{\sigma}$ and $\hat{\tau}$ of the game $\hat{G}$.

2 Algorithms

Algorithm 1: Compute Regret for Eve given Adam can play any strategy

Input : A weighted game $G = (V, V_\exists, v_I, E, w)$, b, and Val the payoff function $\in$\{Sup, Inf, LimInf, LimSup, $\overline{MP}$, $\underline{MP}$\}

Output: Reg, σ, τ

begin
- $W^{'} \longleftarrow$ `ConstructWprime(`G`)`
- $\hat{G} \longleftarrow$ `ConstructGhat(`G, W'`)`
- $Reg, \sigma, \tau \longleftarrow$ `ComputeAVal(`$\hat{G}, b, Val$`)`

end

Algorithm 2: ConstructWprime: Construct $W^{'}$ that represents the best value obtainable for a strategy of Eve that differs at the given edge

Input : Weighted Game Arena $G = (V, V_{\exists}, v_I, E, w)$
Output: Weight function $W^{'}$

begin
  for $e = (u, v) \in E$ **do**
    if $u \in V \setminus V_{\exists}$ **then**
      $W^{'}(e) \longleftarrow -\infty$
    end
    else
      $W^{'}(e) \longleftarrow max\{cVal^{v^{'}} \ |(u,v^{'}) \in E\setminus\{e\}\}$
    end
  end
end

Algorithm 3: ConstructGhat: construct $\hat{G}$

Input : Weighted Game Arena $G = (V, V_{\exists}, v_I, E, w)$ and weight function W'
Output: $\hat{G}$

Initialize $v_o \in \hat{\mathrm{V}} \setminus \hat{\mathrm{V}}_{\exists}$
Initialize $v_1, v_{\perp} \in \hat{\mathrm{V}}_{\exists}$
for $b \in Range(W^{'}) \setminus -\infty$ **do**
  construct $G^b = (V^b, V_{\exists}^b, v_I^b, E^b, \hat{w})$ |G^b is copy of G with edges e restricted by $W^{'}(e) \leq b$
for *e in $\hat{G}$* **do**
  $\mathrm{w}(v_o,\, v_0) = 0$
  $\mathrm{w}(v_{\perp},\, v_{\perp}) = -2w_{max} - 1$
  for *v_I^b in G^b* **do**
    $\mathrm{w}(v_1,\, v_I^b) = 0$
    Initialize weight function $\hat{w}(e) \longleftarrow w(e) - b$
  for $v^b \in V^b$ *that don't have a successor* **do**
    $\mathrm{w}(v^b,\, v_{\perp}) = 0$

Algorithm 4: ComputeAVal: Compute the antagonistic value of $\hat{G}$

Input : An antagonistic game $\hat{G}$, b, and Val the payoff function $\in$ {Sup, Inf, LimInf, LimSup, $\overline{MP}$, $\underline{MP}$}

Output: Reg, σ, τ

$\hat{\sigma}(v_1) = v_I^b$ and avoids $v_\perp$
$\hat{\tau}(v_o) = v_I$
for $b \in Range(W') \setminus -\infty$ **do**
    for $u^b \in V^b$ **do**
        if $u^b \in V^b \setminus V_\exists^b$ **then**
            $\hat{\tau}(u^b) = \{v^b \mid \min(\hat{w}(\mathrm{u}^b, v^b))$ and $(u^b, v^b) \in \mathrm{E}^b\}$
        end
        else if $u^b \in V_\exists^b$ **then**
            $\hat{\sigma}(u^b) = \{v^b \mid \max(\hat{w}(\mathrm{u}^b, v^b))$ and $(u^b, v^b) \in \mathrm{E}^b\}$
        end
    end
end
$Reg = Val_{\hat{G}}^{v}(\hat{\sigma}, \hat{\tau})$
$\sigma = \hat{\sigma} \setminus v_1$
$\tau = \hat{\tau} \setminus v_o$