New Algorithms for Classes of Mean-Payoff Games

Anna Hauschild

Institute for Algorithms and Complexity

June 17, 2022

Contents

Introduction to Mean Payoff Games

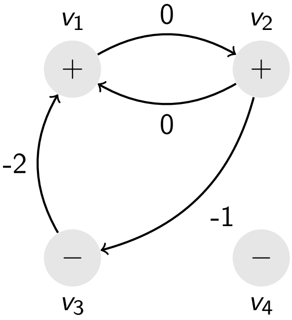


Figure: Graph G_1

- Two players *Min* and *Max*
- vertex set $V = V_+ \cup V_-$
  - $V_+ = \{v_1, v_2\}$
  - $V_- = \{v_3, v_4\}$

Introduction to Mean Payoff Games

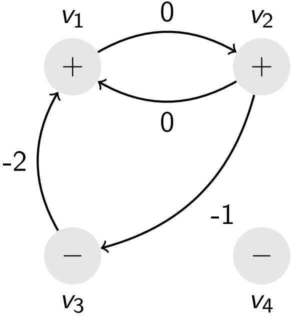


Figure: Graph G_1

- Two players *Min* and *Max*
- Vertex set $V = V_+ \cup V_-$
  - $V_+ = \{v_1, v_2\}$
  - $V_- = \{v_3, v_4\}$
- Edge set $E = \{(v_1, v_2), (v_2, v_1), (v_2, v_3), (v_3, v_1)\}$
- Cost function $\mu(v, w)$ i.e. $\mu(v_2, v_3) = -1$

Introduction to Mean Payoff Games

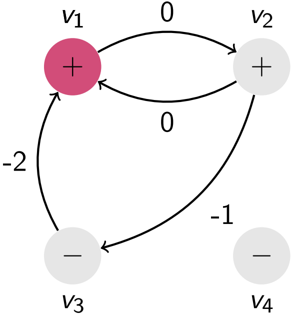


Figure: Graph G_1

- Game $\pi_1 = \{v_1\}$

Introduction to Mean Payoff Games

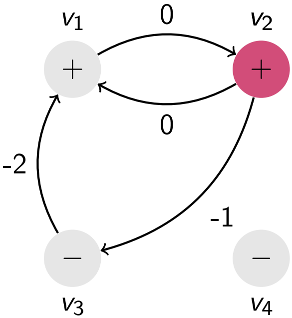


Figure: Graph G_1

- Game $\pi_1 = \{v_1, v_2\}$

Introduction to Mean Payoff Games

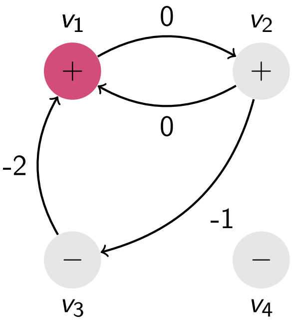


Figure: Graph G_1

- Game $\pi_1 = \{v_1, v_2, v_1, \dots\}$

Introduction to Mean Payoff Games

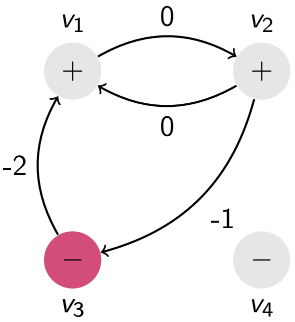


Figure: Graph G_1

- Game $\pi_1 = \{v_1, v_2, v_1, \ldots\}$
- Game $\pi_2 = \{v_3\}$

Introduction to Mean Payoff Games

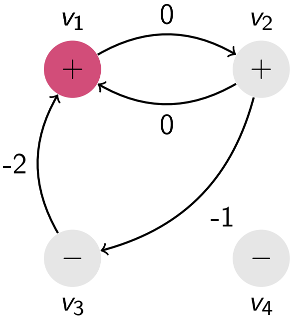


Figure: Graph G_1

- Game $\pi_1 = \{v_1, v_2, v_1, ...\}$
- Game $\pi_2 = \{v_3, v_1\}$

Introduction to Mean Payoff Games

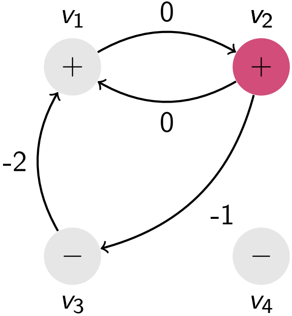


Figure: Graph G_1

- Game $\pi_1 = \{v_1, v_2, v_1, \ldots\}$
- Game $\pi_2 = \{v_3, v_1, v_2\}$

Introduction to Mean Payoff Games

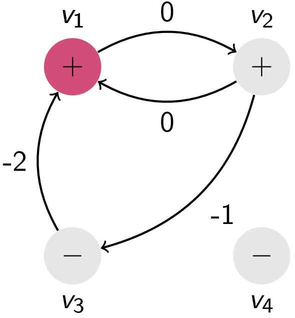


Figure: Graph G_1

- Game $\pi_1 = \{v_1, v_2, v_1, \ldots\}$
- Game $\pi_2 = \{v_3, v_1, v_2, v_1, \ldots\}$

Introduction to Mean Payoff Games

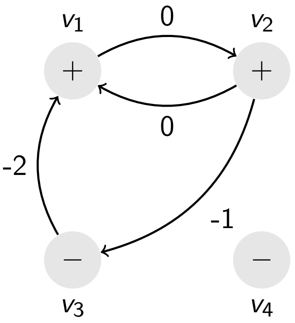


Figure: Graph G_1

- *Characteristic value*

$$\chi(\pi) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mu(v_i, v_{i+1})$$

- $\chi(\pi) > 0 \quad \Rightarrow \quad$ *Max* wins,
  $\chi(\pi) < 0 \quad \Rightarrow \quad$ *Min* wins,
  $\chi(\pi) = 0 \quad \Rightarrow \quad$ Draw

- Introduced by Ehrenfeucht and Mycielski[a]

[a] Ehrenfeucht and Mycielski, "Positional strategies for mean payoff games".

Introduction to Mean Payoff Games

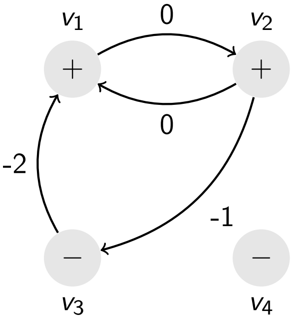


Figure: Graph G_1

- *Sink*:= vertex without outgoing edges
- $v_n \in V_+$ *Min* wins,
  $v_n \in V_-$ *Max* wins

Introduction to Mean Payoff Games

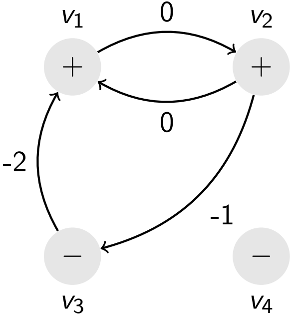


Figure: Graph G_1

- Winning sets
  - W_+ for *Max*,
  - W_- for *Min*,
  - W_0 for Draw
- $W_+ = \{v_4\}$
- $W_- = \emptyset$
- $W_0 = \{v_1, v_2, v_3\}$

Central Idea

- Existing algorithm in $\mathcal{O}(|V|^3 \cdot |E| \cdot W)$[1]
- Instead of solving the whole graph solve small part of it
- Extend winning sets by applying a reachability game
- Especially games with a signature of a potential can be solved in linear time[2]

[1] Zwick and Paterson, "The complexity of mean payoff games on graphs".
[2] Akian et al., "Mean payoff games with signature of a potential".

Functions[3]

function $\text{MAIN}(V_+, V_-, E, \mu)$
    $F := Obsolete(\#V_+\# + V_-, V_+, V_-, E, \mu)$
    while $F \neq \emptyset$ **do**
        $E := E - F$
        $F := Obsolete(\#V_+ + V_-, V_+, V_-, E, \mu)$
    end while
    $W_+ = W_- = W_0 := \emptyset$
    if $V = \emptyset$ **then**
        $r := 0$
    else
        $r := 1$
    end if
    $X_1 := V$
    while $r > 0$ **do**
        ...
    end while
    return (W_+, W_-, W_0)
end function

1. Remove obsolete edges
2. Decompose graph into components
3. Solve one component
4. Check reachability for vertices outside the winning sets

[3] Akian et al., "Mean payoff games with signature of a potential".

Functions[3]

```
function MAIN(V_+, V_-, E, μ)
    ...
    while r > 0 do
        X := X_r − (W_+ ∪ W_- ∪ W_0)
        (s, X_r, ..., X_{r+s−1}) := StrConnComp(X, E)
        r := r + s − 1
        (Y_+, Y_-, Y_0) := SolveMPG(X_r, V_+, V_-, E, μ)
        W_0 := W_0 ∪ Y_0
        for e ∈ {+, −} do
            F := {(v, w) ∈ E | v ∈ Y_e, w ∈ Y_e ∪ W_0}
            W_0 := Reach(W_0, −e, V_+, V_-, F)
            if Y_e ∩ W_0 = ∅ then
                W_e := W_e ∪ Y_e
            end if
        end for
        W_+ := Reach(W_+, +, V_+, V_-, E)
        W_- := Reach(W_-, −, V_+, V_-, E)
        while r > 0 and X_r ⊂ W_+ ∪ W_- ∪ W_0 do
            r := r − 1
        end while
    end while
    return (W_+, W_-, W_0)
end function
```

1. Remove obsolete edges
2. Decompose graph into components
3. Solve one component
4. Check reachability for vertices outside the winning sets

[3] Akian et al., "Mean payoff games with signature of a potential".

Functions[3]

function MAIN(V_+, V_-, E, μ)
  ...
  while $r > 0$ **do**
    $X := X_r - (W_+ \cup W_- \cup W_0)$
    $(s, X_r, \ldots, X_{r+s-1}) := StrConnComp(X, E)$
    $r := r + s - 1$
    $(Y_+, Y_-, Y_0) := SolveMPG(X_r, V_+, V_-, E, \mu)$
    $W_0 := W_0 \cup Y_0$
    for $e \in \{+, -\}$ **do**
      $F := \{(v, w) \in E \mid v \in Y_e, w \in Y_e \cup W_0$
      $W_0 := Reach(W_0, -e, V_+, V_-, F)$
      if $Y_e \cap W_0 = \emptyset$ **then**
        $W_e := W_e \cup Y_e$
      end if
    end for
    $W_+ := Reach(W_+, +, V_+, V_-, E)$
    $W_- := Reach(W_-, -, V_+, V_-, E)$
    while $r > 0$ and $X_r \subset W_+ \cup W_- \cup W_0$ **do**
      $r := r - 1$
    end while
  end while
  return (W_+, W_-, W_0)
end function

1. Remove obsolete edges
2. Decompose graph into components
3. Solve one component
4. Check reachability for vertices outside the winning sets

[3] Akian et al., "Mean payoff games with signature of a potential".

Functions[3]

```
function MAIN(V+, V−, E, μ)
    ...
    while r > 0 do
        X := X_r − (W+ ∪ W− ∪ W0)
        (s, X_r, ..., X_{r+s−1}) := StrConnComp(X, E)
        r := r + s − 1
        (Y+, Y−, Y0) := SolveMPG(X_r, V+, V−, E, μ)
        W0 := W0 ∪ Y0
        for e ∈ {+, −} do
            F := {(v, w) ∈ E | v ∈ Y_e, w ∈ Y_e ∪ W0
            W0 := Reach(W0, −e, V+, V−, F)
            if Y_e ∩ W0 = ∅ then
                W_e := W_e ∪ Y_e
            end if
        end for
        W+ := Reach(W+, +, V+, V−, E)
        W− := Reach(W−, −, V+, V−, E)
        while r > 0 and X_r ⊂ W+ ∪ W− ∪ W0 do
            r := r − 1
        end while
    end while
    return (W+, W−, W0)
end function
```

1. Remove obsolete edges
2. Decompose graph into components
3. Solve one component
4. Check reachability for vertices outside the winning sets

[3] Akian et al., "Mean payoff games with signature of a potential".

Obsolete edges

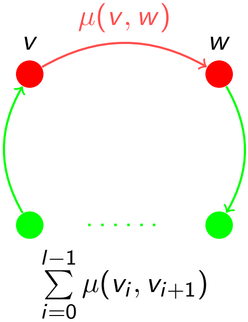


Figure: Obsolete edges

$$e(v) \sum_{i=0}^{l-1} \mu(v_i, v_{i+1}) < -e(v)\mu(v, w).$$

Obsolete edges

function $\textsc{Obsolete}(k, V_+, V_-, E, \mu)$
  $F = \emptyset$
  for $e \in \{+, -\}$ and $v \in V_e$ **do**
    for $i = 0, ..., k-1$ **do** $\eta_i(v) = 0$
    end for
    for $w \in (V_+ \cup V_-) - \{v\}$ **do** $\eta_0(w) := e \cdot \infty$
    end for
    for $i = 1, ..., k-1$ **do**
      for $w \in (V_+ \cup V_-) - \{v\}$ **do**
        $\eta_i(w) := \epsilon(w) \cdot max\{\epsilon(w) \cdot (\mu(w, u) + \eta_{i-1}(u)) \mid (w, u) \in E\}$
      end for
    end for
    for $(v, w) \in E$ **do**
      if $e \cdot \eta_{k-1}(w) < -e \cdot \mu(v, w)$ **then** $F := F \cup \{(v, w)\}$
      end if
    end for
  end for
  return F
end function

Obsolete edges

$$\eta_i(w) := \epsilon(w) \cdot max\{\epsilon(w) \cdot (\mu(w,u) + \eta_{i-1}(u)) \mid (w,u) \in E\}$$

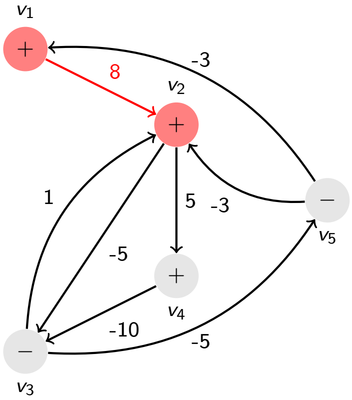


Figure: Graph G_3

$$\eta(v_1) = \begin{array}{r c} & \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ i=0 & \begin{pmatrix} 0 & \infty & \infty & \infty & \infty \\ 0 & & & & \\ 0 & & & & \\ 0 & & & & \\ 0 & & & & \end{pmatrix} \end{array}$$

Obsolete edges

$$\eta_i(w) := \epsilon(w) \cdot max\{\epsilon(w) \cdot (\mu(w,u) + \eta_{i-1}(u)) \mid (w,u) \in E\}$$

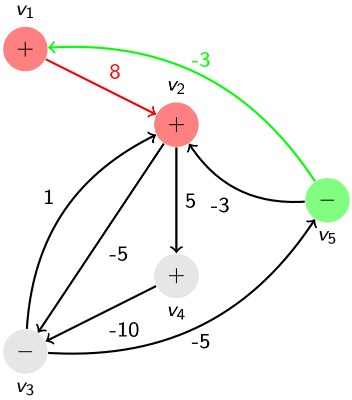


Figure: Graph G_3

$$\eta(v_1) = \begin{array}{cc} & \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{array}{c} i=0 \\ i=1 \\ \\ \\ \\ \end{array} & \begin{pmatrix} 0 & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & -3 \\ 0 & & & & \\ 0 & & & & \\ 0 & & & & \end{pmatrix} \end{array}$$

Obsolete edges

$$\eta_i(w) := \epsilon(w) \cdot max\{\epsilon(w) \cdot (\mu(w,u) + \eta_{i-1}(u)) \mid (w,u) \in E\}$$

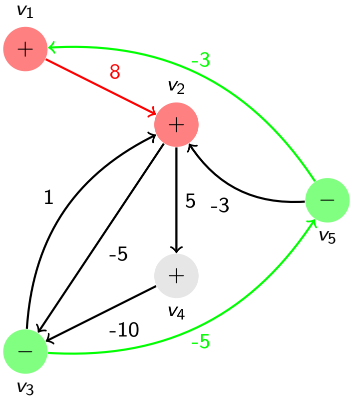


Figure: Graph G_3

$$\eta(v_1) = \begin{array}{r} \\ i=0 \\ i=1 \\ i=2 \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{pmatrix} 0 & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & -3 \\ 0 & \infty & -8 & \infty & -3 \\ 0 & & & & \\ 0 & & & & \end{pmatrix} \end{array}$$

Obsolete edges

$$\eta_i(w) := \epsilon(w) \cdot max\{\epsilon(w) \cdot (\mu(w,u) + \eta_{i-1}(u)) \mid (w,u) \in E\}$$

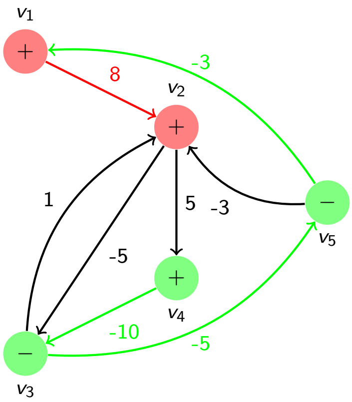


Figure: Graph G_3

$$\eta(v_1) = \begin{array}{r c} & \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{array}{r} i=0 \\ i=1 \\ i=2 \\ i=3 \\ \end{array} & \begin{pmatrix} 0 & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & -3 \\ 0 & \infty & -8 & \infty & -3 \\ 0 & \infty & -8 & -18 & -3 \\ 0 & & & & \end{pmatrix} \end{array}$$

Obsolete edges

$$\eta_i(w) := \epsilon(w) \cdot max\{\epsilon(w) \cdot (\mu(w,u) + \eta_{i-1}(u)) \mid (w,u) \in E\}$$

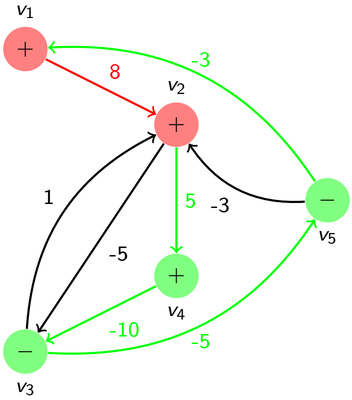


Figure: Graph G_3

$$\eta(v_1) = \begin{array}{c c} & \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{array}{c} i=0 \\ i=1 \\ i=2 \\ i=3 \\ i=4 \end{array} & \begin{pmatrix} 0 & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & -3 \\ 0 & \infty & -8 & \infty & -3 \\ 0 & \infty & -8 & -18 & -3 \\ 0 & -13 & -8 & -18 & -3 \end{pmatrix} \end{array}$$

Obsolete edges

If $e \cdot \eta_{k-1}(w) < -e \cdot \mu(v, w)$ **then** $F := F \cup \{(v, w)\}$

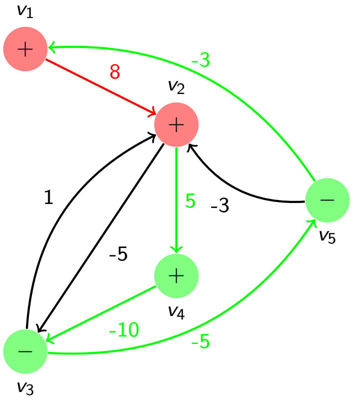


Figure: Graph G_3

$$\eta(v_1) = \begin{array}{c} \\ i=0 \\ i=1 \\ i=2 \\ i=3 \\ i=4 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{pmatrix} 0 & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & -3 \\ 0 & \infty & -8 & \infty & -3 \\ 0 & \infty & -8 & -18 & -3 \\ 0 & \mathbf{-13} & -8 & -18 & -3 \end{pmatrix} \end{array}$$

$$\begin{aligned} 1 \cdot \eta_4(v_2) &< (-1) \cdot \mu(v_1, v_2) \\ 1 \cdot (-13) &< (-1) \cdot 8 \end{aligned}$$

Obsolete edges

$F = \{(v_1, v_2), (v_2, v_3), (v_2, v_4), (v_4, v_3)\}$

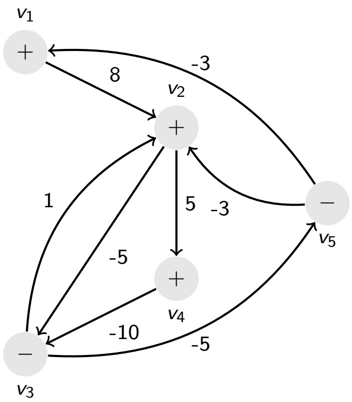


Figure: Graph G_3

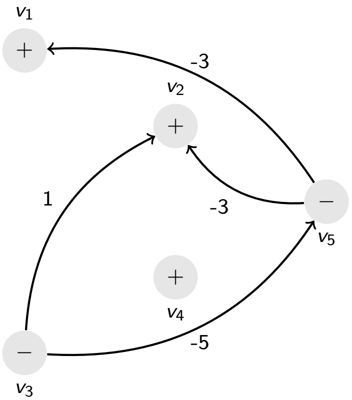


Figure: Graph G_3' after removing obsolete edges

Strongly connected components

Input: X, E

Output: $s, Y_s, ..., Y_1$

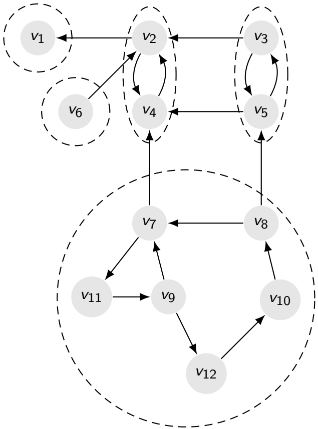
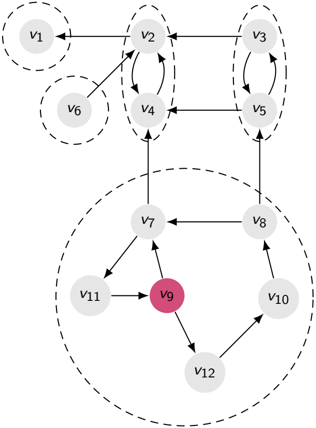


Figure: G_4 decomposed into it's strongly connected components

Strongly connected components

Input: X, E

Output: $s, Y_s, ..., Y_1$



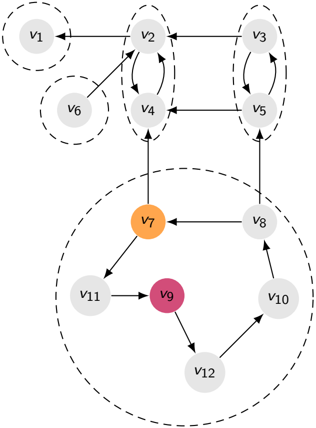
$$v = [v_9]$$

$$m = [1]$$

Figure: G_4 decomposed into its strongly connected components

Strongly connected components

Input: X, E
Output: $s, Y_s, ..., Y_1$



$v = [v_9, v_7]$

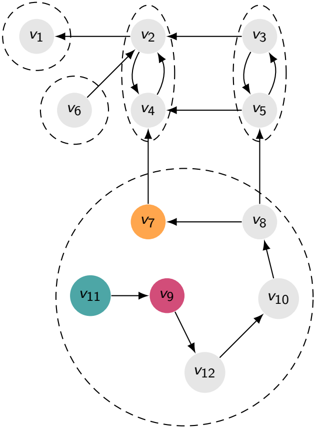
$m = [1, 2]$

Figure: G_4 decomposed into its strongly connected components

Strongly connected components

Input: X, E

Output: $s, Y_s, \ldots, Y_1$



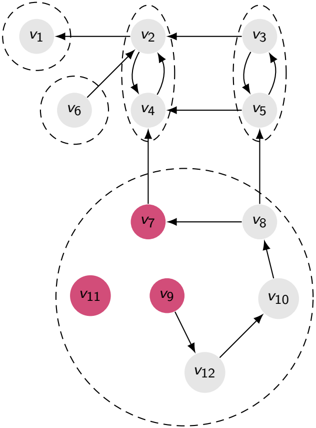
$$v = [v_9, v_7, v_{11}]$$

$$m = [1, 2, 3]$$

Figure: G_4 decomposed into its strongly connected components

Strongly connected components

Input: X, E
Output: $s, Y_s, \ldots, Y_1$



$$v = [v_9, v_7, v_{11}]$$

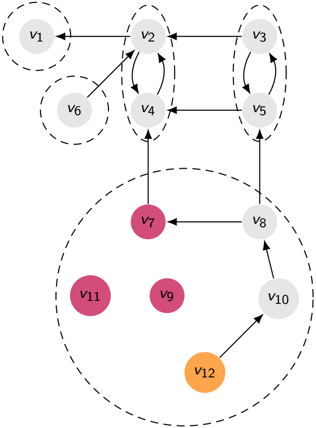
$$m = [1, 1, 1]$$

Figure: G_4 decomposed into its strongly connected components

Strongly connected components

Input: X, E

Output: $s, Y_s, ..., Y_1$



$$v = [v_9, v_7, v_{11}, v_{12}]$$

$$m = [1, 1, 1, 2]$$

Figure: G_4 decomposed into its strongly connected components

Strongly connected components

Input: X, E
Output: $s, Y_s, ..., Y_1$

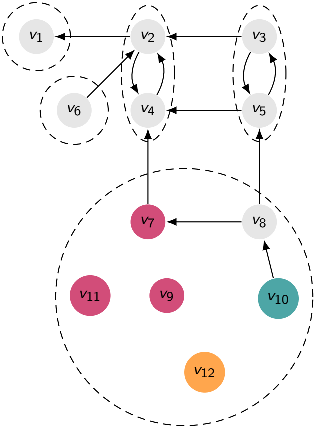


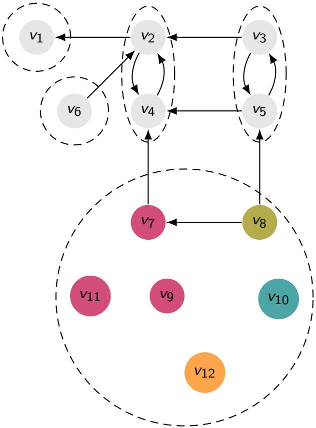
Figure: G_4 decomposed into its strongly connected components

$$v = [v_9, v_7, v_{11}, v_{12}, v_{10}]$$

$$m = [1, 1, 1, 2, 3]$$

Strongly connected components

Input: X, E
Output: $s, Y_s, \ldots, Y_1$



$$v = [v_9, v_7, v_{11}, v_{12}, v_{10}, v_8]$$

$$m = [1, 1, 1, 2, 3, 4]$$

Figure: G_4 decomposed into its strongly connected components

Strongly connected components

Input: X, E
Output: $s, Y_s, ..., Y_1$

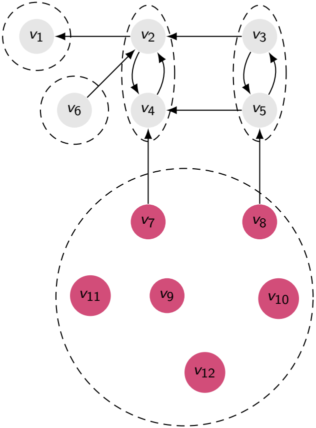


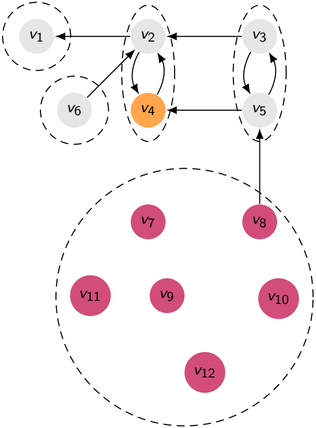
Figure: G_4 decomposed into its strongly connected components

$$v = [v_9, v_7, v_{11}, v_{12}, v_{10}, v_8]$$

$$m = [1, 1, 1, 1, 1, 1]$$

Strongly connected components

Input: X, E
Output: $s, Y_s, ..., Y_1$



$$v = [v_9, v_7, v_{11}, v_{12}, v_{10}, v_8, v_4]$$

$$m = [1, 1, 1, 1, 1, 1, 2]$$

Figure: G_4 decomposed into its strongly connected components

Strongly connected components

Input: X, E
Output: $s, Y_s, \ldots, Y_1$

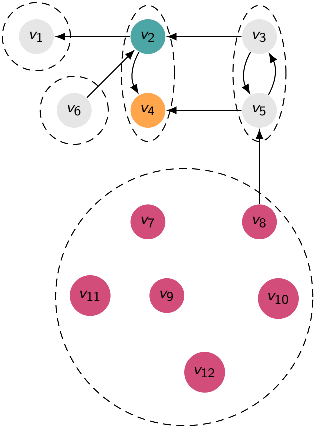


Figure: G_4 decomposed into its strongly connected components

$$v = [v_9, v_7, v_{11}, v_{12}, v_{10}, v_8, v_4, v_2]$$

$$m = [1, 1, 1, 1, 1, 1, 2, 3]$$

Strongly connected components

Input: X, E

Output: $s, Y_s, ..., Y_1$

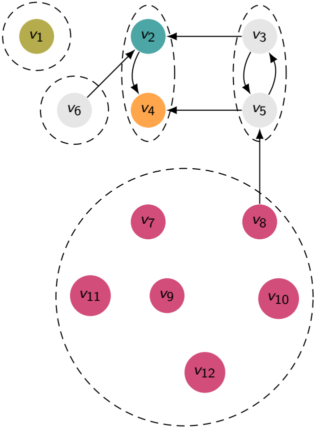


Figure: G_4 decomposed into its strongly connected components

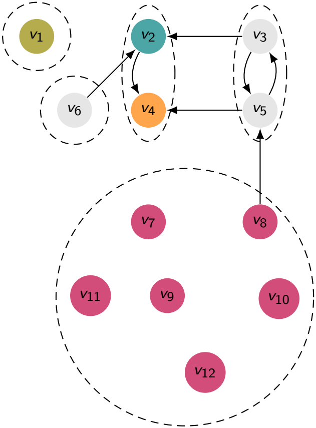
$$v = [v_9, v_7, v_{11}, v_{12}, v_{10}, v_8, v_4, v_2, v_1]$$

$$m = [1, 1, 1, 1, 1, 1, 2, 3, 4]$$

Strongly connected components

Input: X, E

Output: $s, Y_s, \ldots, Y_1$



$v = [v_9, v_7, v_{11}, v_{12}, v_{10}, v_8, v_4, v_2]$

$m = [1, 1, 1, 1, 1, 1, 2, 3]$

No outgoing edge $\Rightarrow$ First component found

$Y_1 = \{v_1\}$

Figure: G_4 decomposed into its strongly connected components

Strongly connected components

Input: X, E
Output: $s, Y_s, \ldots, Y_1$

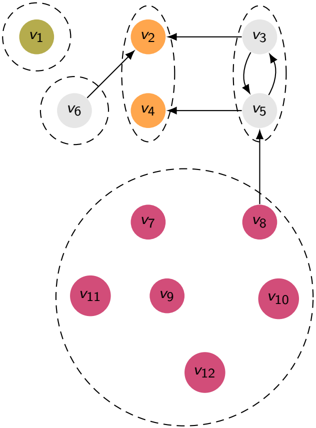


Figure: G_4 decomposed into its strongly connected components

$$v = [v_9, v_7, v_{11}, v_{12}, v_{10}, v_8, v_4, v_2]$$

$$m = [1, 1, 1, 1, 1, 1, 2, 2]$$

$$Y_1 = \{v_1\}$$

Strongly connected components

Input: X, E
Output: $s, Y_s, \ldots, Y_1$

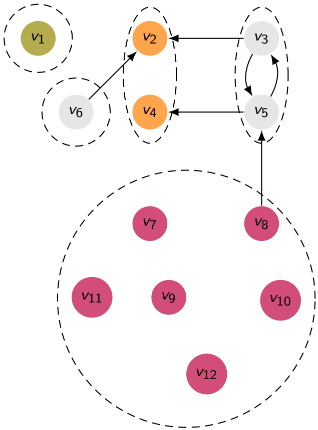


Figure: G_4 decomposed into its strongly connected components

$v = [v_9, v_7, v_{11}, v_{12}, v_{10}, v_8]$

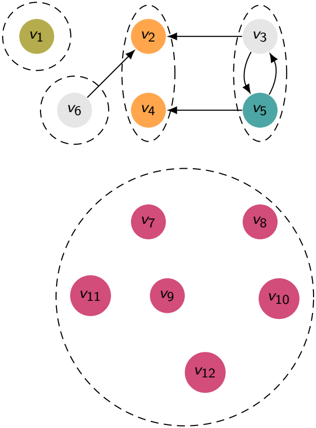
$m = [1, 1, 1, 1, 1, 1]$

No outgoing edge for component $\{v_2, v_4\} \Rightarrow$ Second component found

$Y_1 = \{v_1\}$
$Y_2 = \{v_2, v_4\}$

Strongly connected components

Input: X, E
Output: $s, Y_s, ..., Y_1$



$$v = [v_9, v_7, v_{11}, v_{12}, v_{10}, v_8, v_5]$$

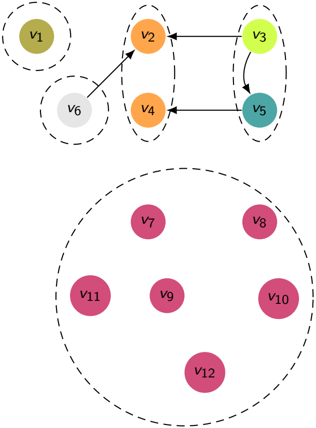
$$m = [1, 1, 1, 1, 1, 1, 2]$$

$$Y_1 = \{v_1\}$$
$$Y_2 = \{v_2, v_4\}$$

Figure: G_4 decomposed into its strongly connected components

Strongly connected components

Input: X, E
Output: $s, Y_s, \dots, Y_1$



$$v = [v_9, v_7, v_{11}, v_{12}, v_{10}, v_8, v_5, v_3]$$

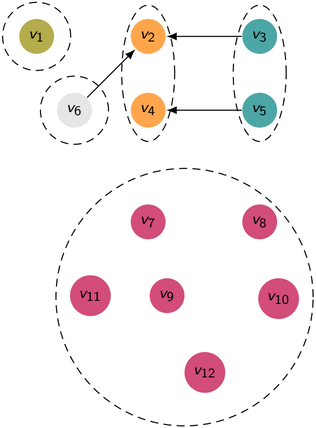
$$m = [1, 1, 1, 1, 1, 1, 2, 3]$$

$$Y_1 = \{v_1\}$$
$$Y_2 = \{v_2, v_4\}$$

Figure: G_4 decomposed into its strongly connected components

Strongly connected components

Input: X, E
Output: $s, Y_s, ..., Y_1$



$$v = [v_9, v_7, v_{11}, v_{12}, v_{10}, v_8, v_5, v_3]$$

$$m = [1, 1, 1, 1, 1, 1, 2, 2]$$

$$Y_1 = \{v_1\}$$
$$Y_2 = \{v_2, v_4\}$$

Figure: G_4 decomposed into its strongly connected components

Strongly connected components

Input: X, E
Output: $s, Y_s, ..., Y_1$

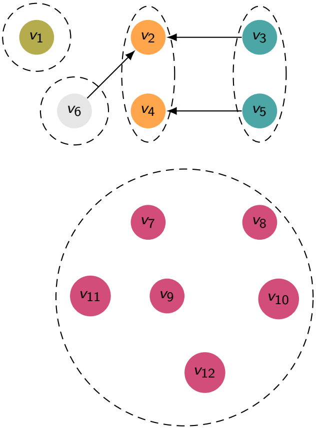


Figure: G_4 decomposed into its strongly connected components

$v = [v_9, v_7, v_{11}, v_{12}, v_{10}, v_8]$

$m = [1, 1, 1, 1, 1, 1]$

Edges lead to already allocated nodes
$\Rightarrow$ Third component found

$Y_1 = \{v_1\}$
$Y_2 = \{v_2, v_4\}$
$Y_3 = \{v_3, v_5\}$

Strongly connected components

Input: X, E

Output: $s, Y_s, \dots, Y_1$

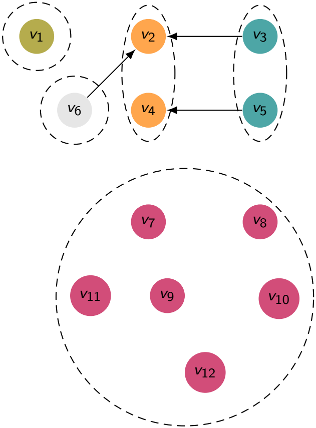


Figure: G_4 decomposed into its strongly connected components

$v = []$

$m = []$

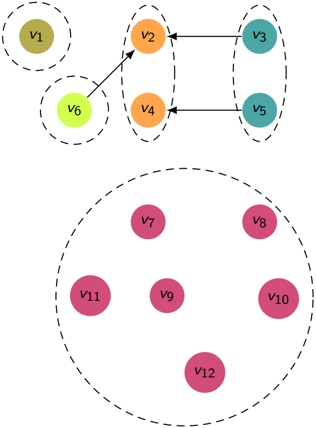
Edges lead to already allocated nodes
$\Rightarrow$ Fourth component found

$Y_1 = \{v_1\}$
$Y_2 = \{v_2, v_4\}$
$Y_3 = \{v_3, v_5\}$
$Y_4 = \{v_9, v_7, v_{11}, v_{12}, v_{10}, v_8\}$

Strongly connected components

Input: X, E

Output: $s, Y_s, \dots, Y_1$



$v = [v_6]$

$m = [1]$

$Y_1 = \{v_1\}$
$Y_2 = \{v_2, v_4\}$
$Y_3 = \{v_3, v_5\}$
$Y_4 = \{v_9, v_7, v_{11}, v_{12}, v_{10}, v_8\}$

Figure: G_4 decomposed into its strongly connected components

Strongly connected components

Input: X, E
Output: $s, Y_s, \dots, Y_1$

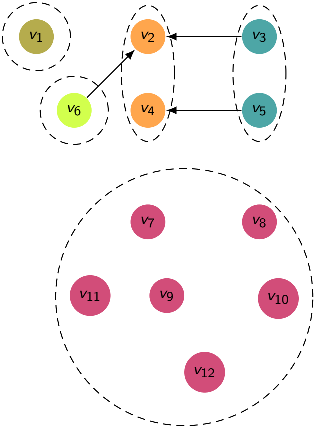


Figure: G_4 decomposed into its strongly connected components

No other vertices left
$\Rightarrow$ Fifth component found

$Y_1 = \{v_1\}$
$Y_2 = \{v_2, v_4\}$
$Y_3 = \{v_3, v_5\}$
$Y_4 = \{v_9, v_7, v_{11}, v_{12}, v_{10}, v_8\}$
$Y_5 = \{v_6\}$

Solving subgraphs

Input: X, V_+, V_-, E, μ
Output: Y_+, Y_-, Y_0

Two ways of ending a game:

- **Finite** game: ends in a sink
- **Infinite** game: ends in a loop

Solving subgraphs

Input: X, V_+, V_-, E, μ
Output: Y_+, Y_-, Y_0

- **Finite** game: ends in a sink

Function SolveMPG(X, V_+, V_-, E, μ) **is**
  $Y_+ = Y_- = Y_0 := \emptyset$
  $C := \{(v, w) \in E \mid v, w \in X\}$
  if $C = \emptyset$ *and* $X \subset V_+$ **then** $Y_- := X$
  if $C = \emptyset$ *and* $X \subset V_-$ **then** $Y_+ := X$

Figure: pseudo code: finite game

Solving subgraphs

Input: X, V_+, V_-, E, μ
Output: Y_+, Y_-, Y_0

- **Infinite** game: ends in a loop

Solving subgraphs

Input: X, V_+, V_-, E, μ
Output: Y_+, Y_-, Y_0

- **Infinite** game: ends in a loop
  - Optimal positional strategy
  - Characteristic value

Solving subgraphs

Input: X, V_+, V_-, E, μ

Output: Y_+, Y_-, Y_0

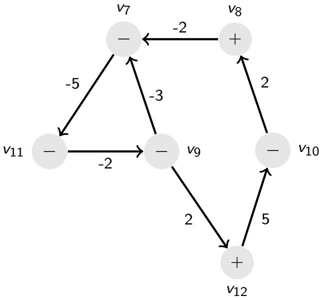


Figure: Subgraph of graph G_4

Solving subgraphs

Input: X, V_+, V_-, E, μ
Output: Y_+, Y_-, Y_0

Characteristic value $\chi(v) < 0 \quad \forall\, v \in \{v_7, ..., v_{12}\}$

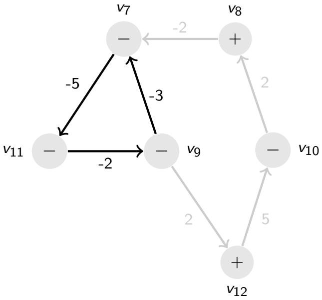


Figure: Subgraph of graph G_4

Solving subgraphs

Input: X, V_+, V_-, E, μ
Output: Y_+, Y_-, Y_0

Characteristic value $\chi(v) = 0 \quad \forall\, v \in \{v_7, ..., v_{12}\}$

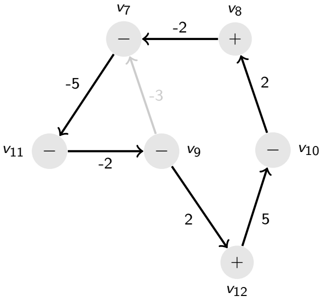


Figure: Subgraph of graph G_4

Solving subgraphs

Input: X, V_+, V_-, E, μ
Output: Y_+, Y_-, Y_0

$Y_- = \{v_7, v_8, v_9, v_{10}, v_{11}, v_{12}\}, Y_+ = \emptyset, Y_0 = \emptyset$

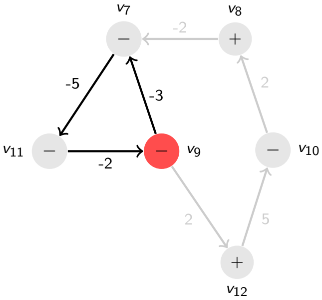


Figure: Subgraph of graph G_4

Reachability

```
function REACH(X, e, V_+, V_-, E)
    for (v, w) ∈ E with v ∉ X and w ∈ X do
        if v ∈ V_e or u ∈ X for all (v, u) ∈ E then
            X := X ∪ {v}
        end if
    end for
    return X
end function
```

Reachability

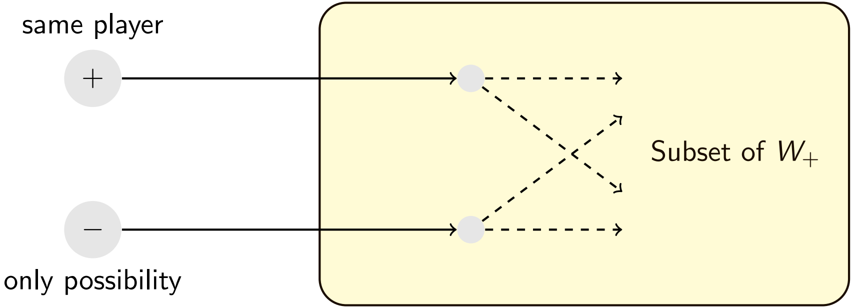


Figure: Finding additional winning vertices of *Max* via reachability

Reachability

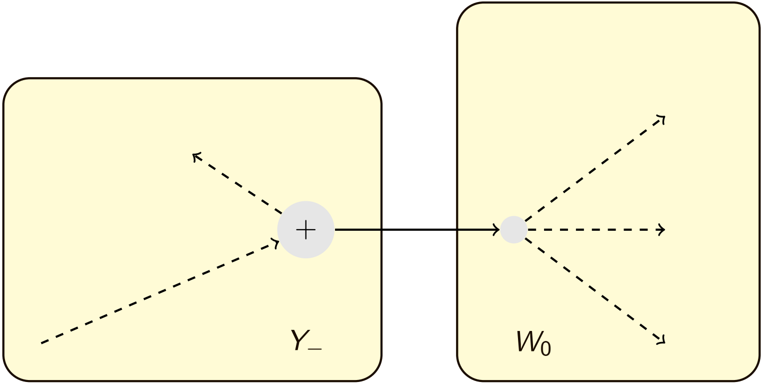


Figure: Additional draw positions for *Max*

Example

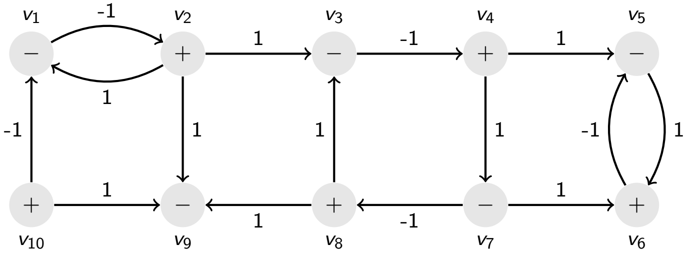


Figure: Graph G_5

Example

- $(s, X_r, ..., X_{r+s-1}) := StrConnComp(X, E)$

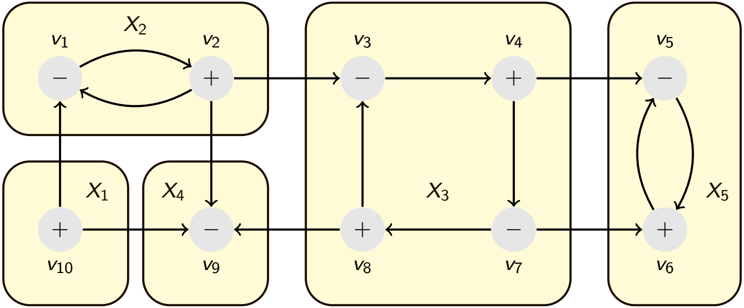


Figure: Strongly connected components

Example

- $(Y_+, Y_-, Y_0) := SolveMPG(X_r, V_+, V_-, E, \mu)$

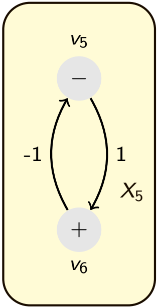


Figure: Subgraph of G_5

$$Y_+ = Y_- = \emptyset,\ \ Y_0 = \{v_5, v_6\}$$

Example

- $W_0 := W_0 \cup Y_0 = \{v_5, v_6\}$

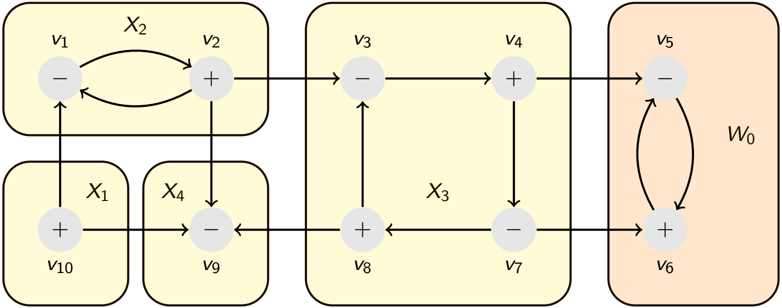


Figure: v_5 and v_6 assigned to W_0

- $(Y_+, Y_-, Y_0) := SolveMPG(X_r, V_+, V_-, E, \mu)$

Example

- $W_+ := Reach(W_+, +, V_+, V_-, E)$

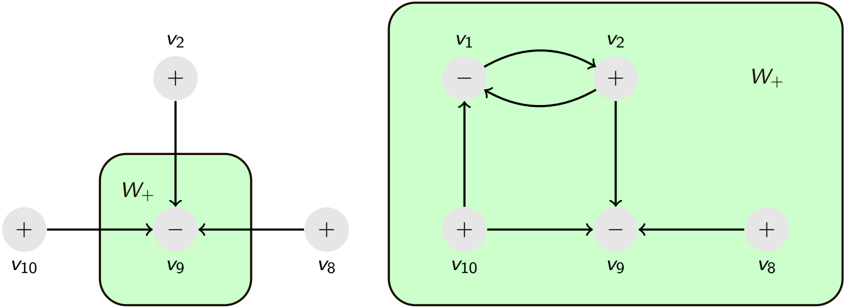


Figure: Reachability game

Example

- $(s, X_r, \ldots, X_{r+s-1}) = (3, X_3, X_4, X_5) := StrConnComp(X, E)$

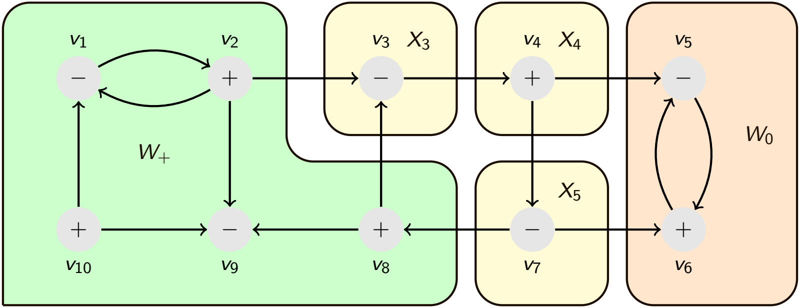


Figure: Third run of *StrConnComp*

Example

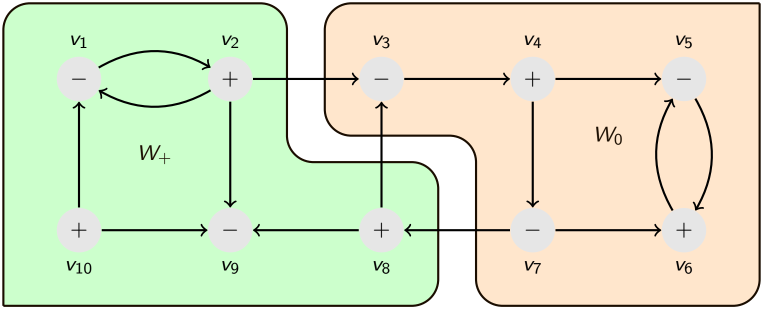


Figure: Resulting winning sets W_+ and W_0

Initialization

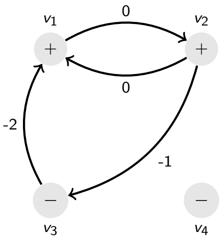


Figure: Graph G_1

$$E = \begin{array}{c c} & \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mu = \begin{array}{c c} & \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} & \begin{pmatrix} -\infty & 0 & -\infty & -\infty \\ 0 & -\infty & -1 & -\infty \\ -2 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \end{pmatrix} \end{array}$$

$$V_+ = \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

$$V_- = \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

Graph generator

- Program to create a random graph[4]
- The number of vertices and *degrees* can be specified or arise randomly
- The graph is stored in a text file
- Option to check and create a signature of a potential

[4] Manish Bhojasia. access on 27.05.22. URL: https://www.sanfoundry.com/.

Experiment 1

- Simultaneously increasing number of nodes and max. degree

#Vertices	max degree	Time[s]
10	10	0.37
50	50	0.51
100	100	2.00
200	200	43.11
500	500	2047.86

Table: Simultaneously increasing vertices and degrees

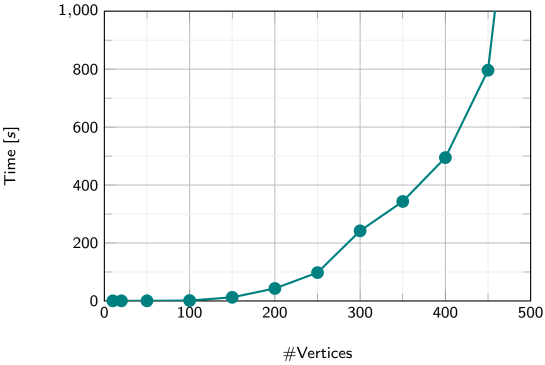


Figure: Simultaneously increasing vertices and degrees

The degree

- Decreasing degree while number of vertices is fixed to 100

max degree	#Components
100	2
50	2
30	3
20	6
10	17
5	46
2	77
1	83
0	100

Table: #Components in relation to degree in a graph with 100 vertices

Experiment 2

- Repeat first experiment with a lower maximal degree of 5

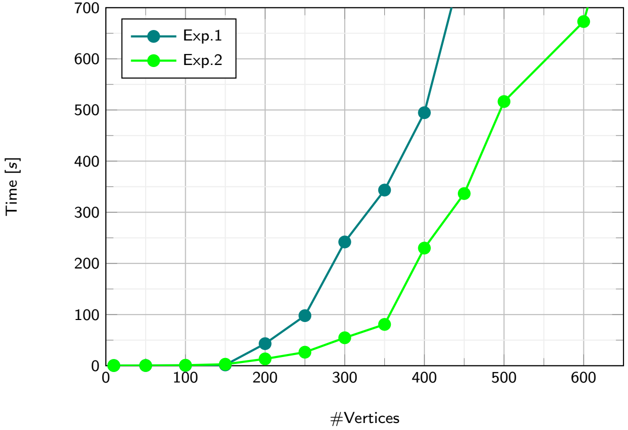


Figure: Comparison arbitrary degree to constant maximal degree 5

Conclusion

- The function *Obsolete* that should normally improve the algorithm claims most of the time
- Already a slightly larger degree leads to a graph with very few components which contradicts the idea of the algorithm
- Observation: the slightly larger degree leads to the result that all nodes belong to the same winning set

- Akian, Mariane et al. "Mean payoff games with signature of a potential". 2021.
- Bhojasia, Manish. access on 27.05.22. URL: https://www.sanfoundry.com/.
- Ehrenfeucht, A. and J. Mycielski. "Positional strategies for mean payoff games". In: *International Journal of Game Theory* 8.2 (1979), pp. 109–113. ISSN: 1432-1270. DOI: 10.1007/BF01768705.
- Zwick, Uri and Mike Paterson. "The complexity of mean payoff games on graphs". In: *Theoretical Computer Science* 158.1 (1996), pp. 343–359. ISSN: 0304-3975. DOI: https://doi.org/10.1016/0304-3975(95)00188-3. URL: https://www.sciencedirect.com/science/article/pii/0304397595001883.