Reconfiguration of Time-Respecting Arborescences

Naoyuki Kamiyama (IMI, Kyushu University)

with
Takehiro Ito, Yuni Iwamasa, Yasuaki Kobayashi, Yusuke Kobayashi,
Shun-ichi Maezawa, and Akira Suzuki

九州大学
KYUSHU UNIVERSITY

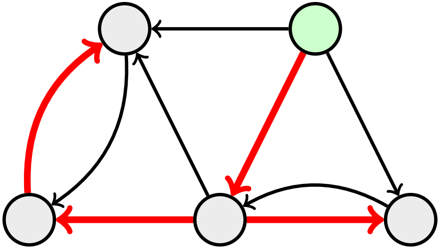
科研費
KAKENHI

Grant Number JP20H05795

Combinatorial Reconfiguration

Arborescence

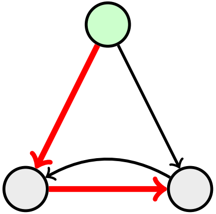
Arborescence = Out-going directed tree in a digraph

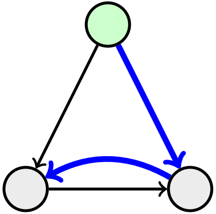


Reconfiguration of Arborescences (Ito et al. TCS 2023)

Reconfiguration of Arborescences (Ito et al. TCS 2023)

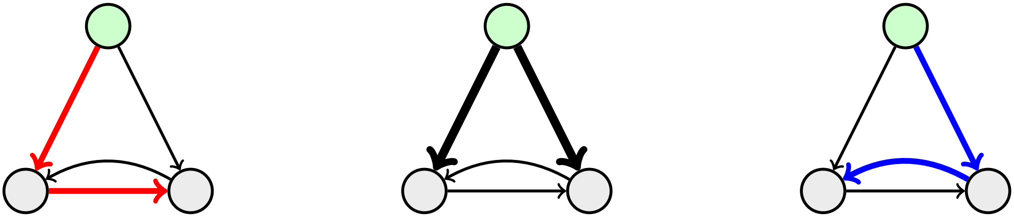
Is there a reconfiguration sequence between given arborescences?





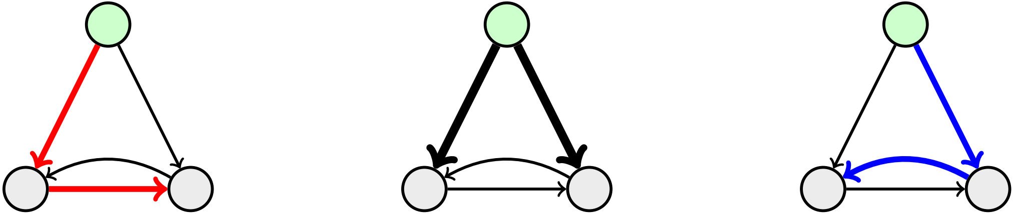
Reconfiguration of Arborescences (Ito et al. TCS 2023)

Is there a reconfiguration sequence between given arborescences?



Reconfiguration of Arborescences (Ito et al. TCS 2023)

Is there a reconfiguration sequence between given arborescences?



- [Ito et al. TCS 2023] Even when the roots are different
  - Checking reachability: **Always Yes**
  - Finding a shortest sequence: **P**

Time-Respecting Arborescence (Kempe et al. JCSS 2002)

Time-Respecting Arborescence (Kempe et al. JCSS 2002)

- A digrph $D = (V, A)$

Time-Respecting Arborescence (Kempe et al. JCSS 2002)

- A digrph $D = (V, A)$
- A **time label** function $\lambda\colon A \to \mathbb{R}_+$

  (End-vertices of arc a can communicate at the time $\lambda(a)$)

Time-Respecting Arborescence (Kempe et al. JCSS 2002)

- A digrph $D = (V, A)$
- A **time label** function $\lambda \colon A \to \mathbb{R}_+$

  (End-vertices of arc a can communicate at the time $\lambda(a)$)
- A dipath $(a_1, a_2, \ldots, a_k)$ in D is **time-respecting**

  $\overset{\text{def}}{\iff} \lambda(a_1) \leq \lambda(a_2) \leq \cdots \leq \lambda(a_k)$

Time-Respecting Arborescence (Kempe et al. JCSS 2002)

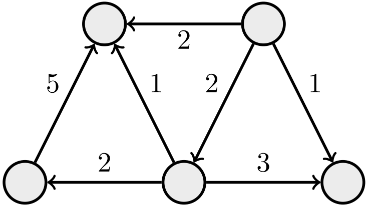
- A digrph $D = (V, A)$
- A **time label** function $\lambda \colon A \to \mathbb{R}_+$

  (End-vertices of arc a can communicate at the time $\lambda(a)$)
- A dipath $(a_1, a_2, \ldots, a_k)$ in D is **time-respecting**

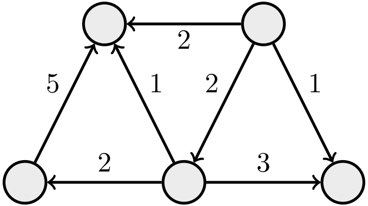
  $\overset{\text{def}}{\iff} \ \lambda(a_1) \leq \lambda(a_2) \leq \cdots \leq \lambda(a_k)$
- An r-arborescence T in D is **time-respecting**

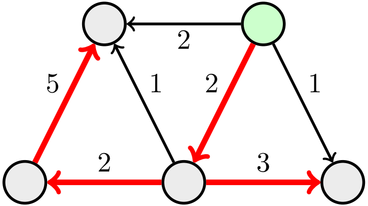
  $\overset{\text{def}}{\iff} \ \forall v \in V :$ the (r, v)-dipath in T is time-respecting

Time-Respecting Arborescence (Kempe et al. JCSS 2002)

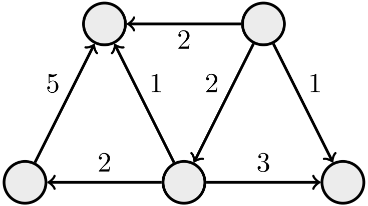


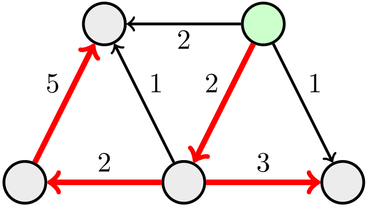
Time-Respecting Arborescence (Kempe et al. JCSS 2002)

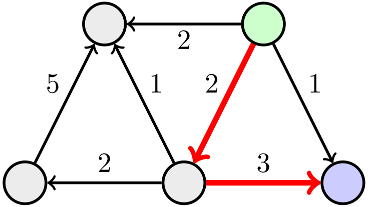




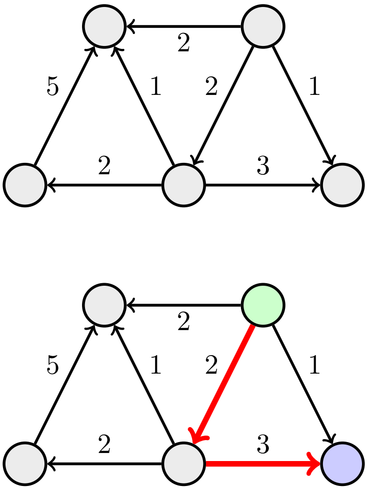
Time-Respecting Arborescence (Kempe et al. JCSS 2002)







Time-Respecting Arborescence (Kempe et al. JCSS 2002)



Reconfiguration of Time-Respecting Arborescences

Reconfiguration of Time-Respecting Arborescences

- Consider time-respecting arboresnces $T_{\mathrm{o}}, T_{\mathrm{t}}$

Reconfiguration of Time-Respecting Arborescences

- Consider time-respecting arboresnces $T_{\mathrm{o}}, T_{\mathrm{t}}$
- A **reconfiguration sequence** $(T^0, T^1, \ldots, T^\ell)$ between $T_{\mathrm{o}}, T_{\mathrm{t}}$

Reconfiguration of Time-Respecting Arborescences

- Consider time-respecting arboresnces $T_{\mathrm{o}}, T_{\mathrm{t}}$
- A **reconfiguration sequence** $(T^0, T^1, \ldots, T^\ell)$ between $T_{\mathrm{o}}, T_{\mathrm{t}}$

  $\overset{\mathrm{def}}{\Longleftrightarrow}$ $T^0 = T_{\mathrm{o}}$ and $T^\ell = T_{\mathrm{t}}$,

  $T^0, \ldots, T^\ell$ are time-respecting arborescences in D, and

  $\forall 0 \leq i < \ell \colon |A(T^{i+1}) \setminus A(T^i)| = |A(T^i) \setminus A(T^{i+1})| = 1$

Reconfiguration of Time-Respecting Arborescences

- Consider time-respecting arboresnces $T_{\mathrm{o}}, T_{\mathrm{t}}$
- A **reconfiguration sequence** $(T^0, T^1, \ldots, T^\ell)$ between $T_{\mathrm{o}}, T_{\mathrm{t}}$

$\overset{\text{def}}{\Longleftrightarrow}$ $T^0 = T_{\mathrm{o}}$ and $T^\ell = T_{\mathrm{t}}$,

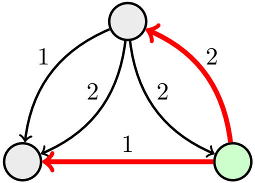
$T^0, \ldots, T^\ell$ are time-respecting arborescences in D, and

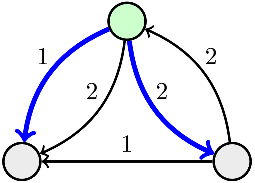
$\forall 0 \leq i < \ell\colon |A(T^{i+1}) \setminus A(T^i)| = |A(T^i) \setminus A(T^{i+1})| = 1$

Our problem:

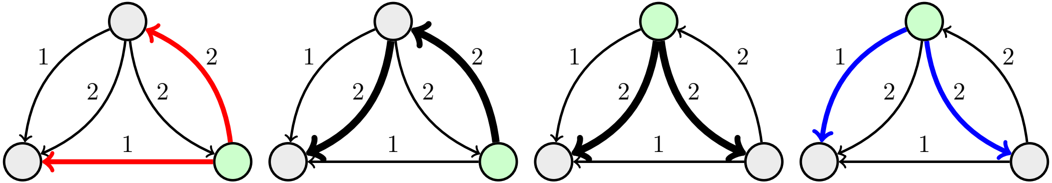
- **Input:** Time-respecting arboresnces T_1, T_2
- **Question:** Is there a reconfiguration sequence between T_1, T_2?

Yes instance:



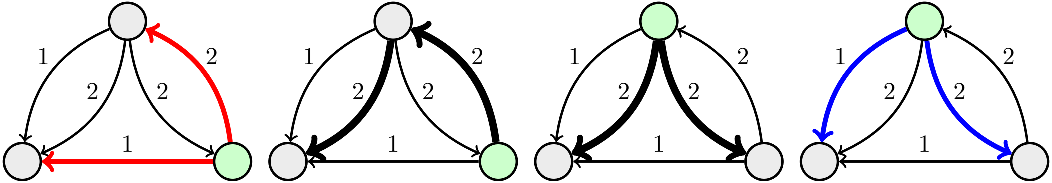


Yes instance:

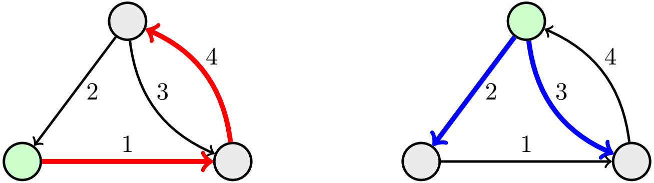


Reconfiguration of Time-Respecting Arborescences

Yes instance:



No instance:



Results

- (without time labels) Identical/Non-identical roots case: Reachability: Always Yes, and Shortest sequence: P

Results

- **Identical roots case:**
  - Checking reachability: **Always Yes**
  - Finding a shortest sequence: **P**

- (without time labels) Identical/Non-identical roots case:

  Reachability: Always Yes, and Shortest sequence: P

Results

- **Identical roots case:**
  - Checking reachability: **Always Yes**
  - Finding a shortest sequence: **P**
- **Non-identical roots case:**
  - Checking reachability: **P**
  - Finding a shortest sequence: **NP-complete** (Decision version)

- (without time labels) Identical/Non-identical roots case:

  Reachability: Always Yes, and Shortest sequence: P

Identical Roots Case

Identical Roots Case

- TR := time-respecting
- $T \rightsquigarrow T'$:= a reconfiguration sequence between T, T'
- T_{o} (resp. T_{t}) := initial (resp. target) TR-arboresncence

Identical Roots Case

- TR := time-respecting
- $T \rightsquigarrow T'$:= a reconfiguration sequence between T, T'
- T_{o} (resp. T_{t}) := initial (resp. target) TR-arboresncence

- **Sketch of Proof:**

Identical Roots Case

- TR := time-respecting
- $T \rightsquigarrow T'$:= a reconfiguration sequence between T, T'
- T_{o} (resp. T_{t}) := initial (resp. target) TR-arboresncence

- **Sketch of Proof:**

  i) There is a **minimal TR arborescence** T^*

Identical Roots Case

- TR := time-respecting
- $T \rightsquigarrow T'$:= a reconfiguration sequence between T, T'
- T_{o} (resp. T_{t}) := initial (resp. target) TR-arboresncence

- **Sketch of Proof:**

  i) There is a **minimal TR arborescence** T^*

  ii) There are $T_{\mathrm{o}} \rightsquigarrow T^*$ and $T^* \rightsquigarrow T_{\mathrm{t}}$

Identical Roots Case

- TR := time-respecting
- $T \rightsquigarrow T'$:= a reconfiguration sequence between T, T'
- $T_{\rm o}$ (resp. $T_{\rm t}$) := initial (resp. target) TR-arboresncence

- **Sketch of Proof:**

  i) There is a **minimal TR arborescence** T^*

  ii) There are $T_{\rm o} \rightsquigarrow T^*$ and $T^* \rightsquigarrow T_{\rm t}$

  iii) Length of $T_{\rm o} \rightsquigarrow T^* \rightsquigarrow T_{\rm t} = |A(T_{\rm o}) \setminus A(T_{\rm t})|$

Minimal TR arborescence

Minimal TR arborescence

- For each $v \in V$, we define $d(v)$ by

$$d(v) := \min\{\lambda(a) \mid a \text{ enters } v \text{, and } \exists \text{ TR dipath from } r \text{ containing } a\}$$

Identical Roots Case

Minimal TR arborescence

- For each $v \in V$, we define $d(v)$ by

$$d(v) := \min\{\lambda(a) \mid a \text{ enters } v, \text{ and } \exists \text{ TR dipath from } r \text{ containing } a\}$$

- A TR r-arborescence is **minimal** $\overset{\text{def}}{\iff} \forall v \in V \setminus \{r\}:$

  the unique arc a of T entering v satisfies $\lambda(a) = d(v)$

Minimal TR arborescence

- For each $v \in V$, we define $d(v)$ by

$$d(v) := \min\{\lambda(a) \mid a \text{ enters } v, \text{ and } \exists \text{ TR dipath from } r \text{ containing } a\}$$

- A TR r-arborescence is **minimal** $\overset{\text{def}}{\iff}$ $\forall v \in V \setminus \{r\}$:

  the unique arc a of T entering v satisfies $\lambda(a) = d(v)$

- We can find a minimal TR arborescence in poly-time

  (Also, $T_{\mathrm{o}} \rightsquigarrow T^* \rightsquigarrow T_{\mathrm{t}}$ can be found in poly-time)

Non-identical Roots Case: Reachability

Non-identical Roots Case: Reachability

- Auxiliary graph $\mathcal{G}$ is define as follows

Non-identical Roots Case: Reachability

- Auxiliary graph $\mathcal{G}$ is define as follows
  - The vertex set of $\mathcal{G}$ is V

Non-identical Roots Case: Reachability

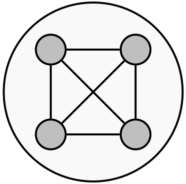
- Auxiliary graph $\mathcal{G}$ is define as follows
  - The vertex set of $\mathcal{G}$ is V
  - $u, v \in V$ is connected in $\mathcal{G}$ $\iff$

    $\exists$ u-arborescence T and v-arboerscence T' such that

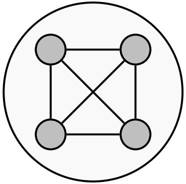
    there is a reconfiguration sequence between T, T'

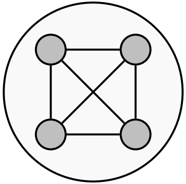
Non-identical Roots Case: Reachability

root $= u$

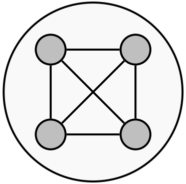


root $= w$



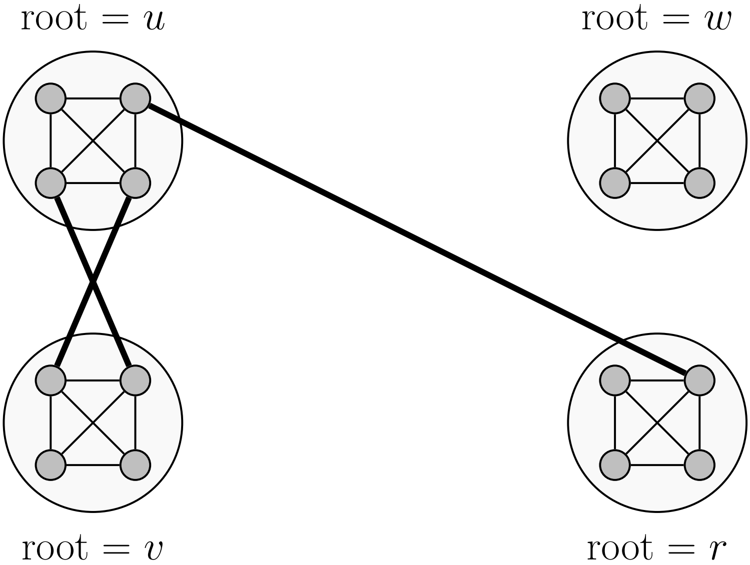


root $= v$

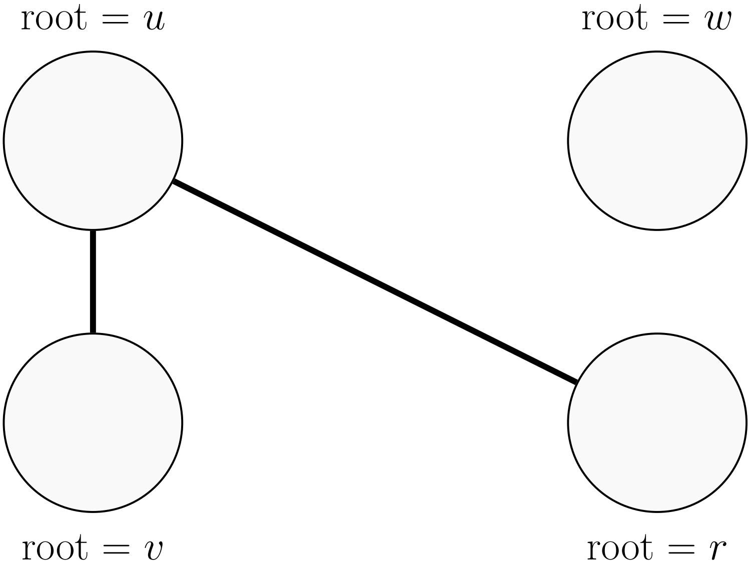


root $= r$

Non-identical Roots Case: Reachability



Non-identical Roots Case: Reachability



Non-identical Roots Case: Reachability

- Auxiliary graph $\mathcal{G}$ is define as follows
  - The vertex set of $\mathcal{G}$ is V
  - $u, v \in V$ is connected in $\mathcal{G}$ $\iff$

    $\exists$ u-arborescence T and v-arboerscence T' such that

    there is a reconfiguration sequence between T, T'

Non-identical Roots Case: Reachability

- Auxiliary graph $\mathcal{G}$ is define as follows
  - The vertex set of $\mathcal{G}$ is V
  - $u, v \in V$ is connected in $\mathcal{G}$ $\iff$ $\exists$ u-arborescence T and v-arboerscence T' such that there is a reconfiguration sequence between T, T'
- We prove $\mathcal{G}$ can be constructed in polynomial time

Non-identical Roots Case: Reachability

- Auxiliary graph $\mathcal{G}$ is define as follows
  - The vertex set of $\mathcal{G}$ is V
  - $u, v \in V$ is connected in $\mathcal{G} \iff$
    $\exists$ u-arborescence T and v-arboerscence T' such that there is a reconfiguration sequence between T, T'
- We prove $\mathcal{G}$ can be constructed in polynomial time
- Suppose T_{o} is rooted at r_{o} and T_{t} is rooted at r_{t}

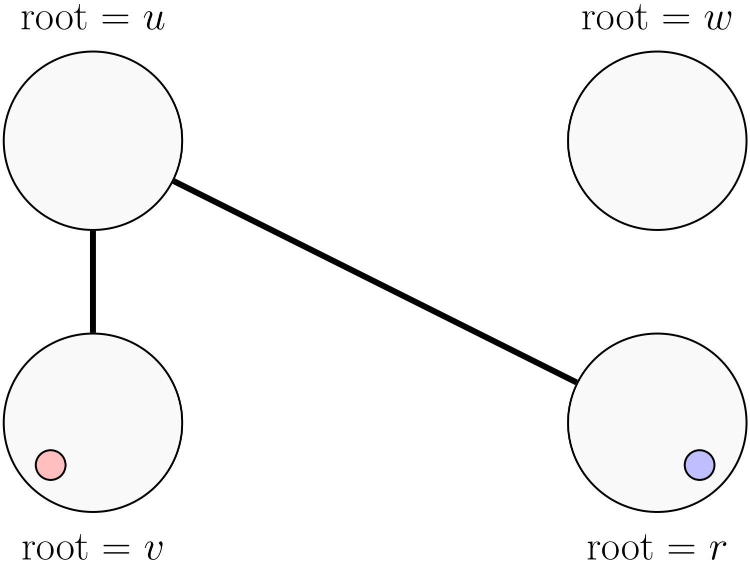
Non-identical Roots Case: Reachability

- Auxiliary graph $\mathcal{G}$ is define as follows
  - The vertex set of $\mathcal{G}$ is V
  - $u, v \in V$ is connected in $\mathcal{G}$ $\iff$

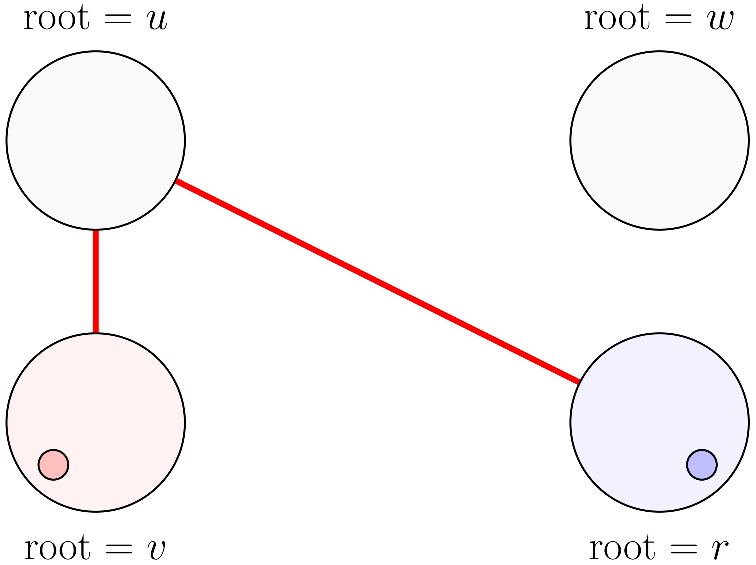
    $\exists$ u-arborescence T and v-arboerscence T' such that

    there is a reconfiguration sequence between T, T'
- We prove $\mathcal{G}$ can be constructed in polynomial time
- Suppose T_{o} is rooted at r_{o} and T_{t} is rooted at r_{t}
- We check the reachability between $r_{\mathrm{o}}, r_{\mathrm{t}}$ in $\mathcal{G}$

Non-identical Roots Case: Reachability



Non-identical Roots Case: Reachability



Conclusion

- We consider reconfiguration of time-respecting arborescences
- **Identical roots case:**
  - Checking reachability: **Always Yes**
  - Finding a shortest sequence: **P**
- **Non-identical roots case:**
  - Checking reachability: **P**
  - Finding a shortest sequence: **NP-complete** (Decision version)

Thank you for your attention!!