A special case of the data arrangement problem on binary trees

Rostislav Staněk*

Eranda Çela**

Joachim Schauer*

*Department of Statistics and Operations Research, University of Graz
**Department of Optimization and Discrete Mathematics, Graz University of Technology
Short overview

▶ problem definition
Short overview

▸ problem definition
▸ upper bound
Short overview

- problem definition
- upper bound (solution algorithm)
Short overview

▶ problem definition
▶ upper bound (solution algorithm)
▶ lower bound
Short overview

- problem definition
- upper bound (solution algorithm)
- lower bound:
  - problem transformation
Short overview

- problem definition
- upper bound (solution algorithm)
- lower bound:
  - problem transformation
- recapitulation, future research and open questions
Problem definition

- Given
  - an undirected graph $G = (V(G), E(G))$, 

\[ \sum_{(i, j) \in E(G)} d(\phi(i), \phi(j)), \]
Problem definition

▶ Given
  ▶ an undirected graph $G = (V(G), E(G))$,
  ▶ an undirected graph $T = (V(T), E(T))$ with $|V(T)| \geq |V(G)|$ and
Problem definition

Given

- an undirected graph $G = (V(G), E(G))$,
- an undirected graph $T = (V(T), E(T))$ with $|V(T)| \geq |V(G)|$ and
- a subset $B \subseteq V(T)$ with $|B| \geq |V(G)|$,
Problem definition

- Given
  - an undirected graph \( G = (V(G), E(G)) \),
  - an undirected graph \( T = (V(T), E(T)) \) with \( |V(T)| \geq |V(G)| \) and
  - a subset \( B \subseteq V(T) \) with \( |B| \geq |V(G)| \),

the **generic graph embedding problem (GEP)** consists of finding an injective embedding of the vertices of \( G \) into the vertices in \( B \) such that some prespecified objective function is minimised.
Problem definition

Given

- an undirected graph $G = (V(G), E(G))$,
- an undirected graph $T = (V(T), E(T))$ with $|V(T)| \geq |V(G)|$ and
- a subset $B \subseteq V(T)$ with $|B| \geq |V(G)|$,

the generic graph embedding problem (GEP) consists of finding an injective embedding of the vertices of $G$ into the vertices in $B$ such that some prespecified objective function is minimised.

A commonly used objective function maps an embedding $\phi: V(G) \rightarrow B$ to

$$\sum_{(i,j) \in E(G)} d(\phi(i), \phi(j)),$$

where $d(x, y)$ denotes the length of the shortest path between $x$ and $y$ in $T$. 
Problem definition

- Given an undirected graph $G = (V(G), E(G))$,
- an undirected graph $T = (V(T), E(T))$ with $|V(T)| \geq |V(G)|$ and
- a subset $B \subseteq V(T)$ with $|B| \geq |V(G)|$,

the **generic graph embedding problem (GEP)** consists of finding an injective embedding of the vertices of $G$ into the vertices in $B$ such that some prespecified objective function is minimised.

- A commonly used objective function maps an embedding $\phi: V(G) \rightarrow B$ to

$$
\sum_{(i,j) \in E(G)} d(\phi(i), \phi(j)),
$$

where $d(x, y)$ denotes the length of the shortest path between $x$ and $y$ in $T$. 
Problem definition

- Different special cases of the GEP have been studied in the literature.
Problem definition

- Different special cases of the GEP have been studied in the literature:
  - The linear arrangement problem (LAP) is probably the most prominent one.
Problem definition

- Different special cases of the GEP have been studied in the literature:
  - The **linear arrangement problem (LAP)** is probably the most prominent one:
    - The problem is solvable in polynomial time for undirected trees [Shiloach 1979\(^1\), Chung 1984\(^2\)].


Problem definition

- Different special cases of the GEP have been studied in the literature:
  - The **linear arrangement problem (LAP)** is probably the most prominent one:
    - The problem is solvable in polynomial time for undirected trees \([\text{Shiloach 1979}^{1}, \text{Chung 1984}^{2}].\)
    - Juvan and Mohar use the eigenvalues in order to obtain a heuristic solution \([\text{Juvan, Mohar 1992}^{3}].\)

---

Problem definition

- Different special cases of the GEP have been studied in the literature:
  - The **linear arrangement problem (LAP)** is probably the most prominent one:
    - The problem is solvable in polynomial time for undirected trees [Shiloach 1979¹, Chung 1984²].
    - Juvan and Mohar use the eigenvalues in order to obtain a heuristic solution [Juvan, Mohar 1992³].
  - In our case $T$ is a $d$-regular tree and $B$ is the set of its leaves.

Problem definition

▶ Different special cases of the GEP have been studied in the literature:
  ▶ The **linear arrangement problem (LAP)** is probably the most prominent one:
    ▶ The problem is solvable in polynomial time for undirected trees [Shiloach 1979\(^1\), Chung 1984\(^2\)].
    ▶ Juvan and Mohar use the eigenvalues in order to obtain a heuristic solution [Juvan, Mohar 1992\(^3\)].
  ▶ In our case \(T\) is a \(d\)-regular tree and \(B\) is the set of its leaves.
  ▶ We will call this problem **data arrangement problem on regular trees (DAPT)** and denote the objective value \(OV(G, d, \phi)\).

Problem definition
Problem definition

Algorithmic Graph Theory on the Adriatic Coast

$OV(G, 3, \phi) = 20$
General properties and our special case

- DAPT is \( \mathcal{NP} \)-hard for every fixed \( d \geq 2 \) [Luczak, Noble 2002\(^4\)].

General properties and our special case

- DAPT is $\mathcal{NP}$-hard for every fixed $d \geq 2$ [Luczak, Noble 2002$^4$].
- Čela and S. introduce some heuristics for this problem [Čela, S. 2013$^5$].


General properties and our special case

- DAPT is \(NP\)-hard for every fixed \(d \geq 2\) [Luzcak, Noble 2002\(^4\)].
- Čela and S. introduce some heuristics for this problem [Čela, S. 2013\(^5\)].
- We deal with the special case where \(G\) and \(T\) are both binary regular trees.


Solution algorithm

OV(G, 2, φ̂) = 6
Solution algorithm

$OV(G, 2, \phi^*) = 6$
Solution algorithm
Solution algorithm

\[ OV(G, 2, \phi^*) = 22 \]
Solution algorithm

```
OV(G, 2, φ*) = 58
```
Solution algorithm

\[ OV(G, 2, \phi^*) = 58 \]
Solution algorithm

\[ \text{OV}(G, 2, \phi^*) = 58 \]
Solution algorithm

$$OV(G, 2, \phi^*) = 56$$
Solution algorithm

Require: binary regular tree $G = (V, E)$ of height $h_G$ labelled according to the canonical order

Ensure: arrangement $\phi^*$

1: $b := 2^{h_G+1};$
2: if $h_G = 0$ then
3: $\phi^*(v_1) := b_1;$
4: else \{ $h_G > 0$ \}
5: solve the problem for the basic subtrees $\hat{G}_1$ and $\hat{G}_2$, place the obtained arrangements on the leaves $b_1, b_2, \ldots, b_{\frac{1}{2}b}$ and $b_{\frac{1}{2}b+1}, b_{\frac{1}{2}b+2}, \ldots, b_b$ and, finally, place the root on the leaf $b_{\frac{1}{2}b};$
6: if $h_G$ is odd and $h_G \geq 3$ then
7: make pair-exchange of the vertices arranged on the leaves $b_{\frac{1}{4}b-1}$ and $b_{\frac{1}{2}b};$
8: end if
9: end if
10: return $\phi^*;$
Solution algorithm

Theorem

Given the binary regular trees $G = (V, E)$ and $T$ with heights $h_G$ and $h = h_G + 1$, let $G$ be the guest graph and $T$ the host graph and let $\phi^*$ be the arrangement obtained from the described algorithm. Then

$$OV(G, 2, \phi^*) = \begin{cases} 
0 & \text{for } h_G = 0 \\
\frac{29}{3} \cdot 2^{h_G} - 4h_G - 9 + \frac{1}{3}(-1)^{h_G} & \text{for } h_G \geq 1
\end{cases}$$

holds.
Lower bound – problem transformation

\[ OV(G, 2, \phi^*) = 56 \]
Lower bound – problem transformation

\[ OV(G, 2, \phi^*) = 56 \]

\[ OV(G, 2, \phi) = 2(1 \cdot 4 + 3 \cdot 3 + 5 \cdot 2 + 5 \cdot 1) = 56 \]
Lower bound – problem transformation

\[ OV(G, 2, \phi^*) = 56 \]

- \[ OV(G, 2, \phi) = 2(1 \cdot 4 + 3 \cdot 3 + 5 \cdot 2 + 5 \cdot 1) = 56 \]
- \[ OV(G, 2, \phi) = 2(a_h(\phi) \cdot h + a_{h-1}(\phi) \cdot (h - 1) + \ldots + a_1(\phi) \cdot 1) \]
Lower bound – problem transformation

\[ OV(G, 2, \phi^*) = 56 \]

\[ OV(G, 2, \phi) = 2 \sum_{i=1}^{h} a_i(\phi) \cdot i \]
Lower bound – problem transformation

$$OV(G, 2, \phi^*) = 56$$

- $$OV(G, 2, \phi) = 2 \sum_{i=1}^{h} a_i(\phi) \cdot i$$
- $$s_i(\phi) := \sum_{j=i}^{h} a_j(\phi)$$ for all $$1 \leq i \leq h$$
Lower bound – problem transformation

\[ OV(G, 2, \phi^*) = 56 \]

- \[ OV(G, 2, \phi) = 2 \sum_{i=1}^{h} a_i(\phi) \cdot i \]
- \[ s_i(\phi) := \sum_{j=i}^{h} a_j(\phi) \text{ for all } 1 \leq i \leq h \]
- \[ a_i(\phi) = \begin{cases} 
  s_i(\phi) - s_{i+1}(\phi) & \text{for } 1 \leq i \leq h - 1 \\
  s_i(\phi) & \text{for } i = h 
\end{cases} \]
Lower bound – problem transformation

\[ OV(G, 2, \phi^*) = 56 \]

- \( OV(G, 2, \phi) = 2 \sum_{i=1}^{h} a_i(\phi) \cdot i \)
- \( s_i(\phi) := \sum_{j=i}^{h} a_j(\phi) \) for all \( 1 \leq i \leq h \)
- \( a_i(\phi) = \begin{cases} s_i(\phi) - s_{i+1}(\phi) & \text{for } 1 \leq i \leq h - 1 \\ s_i(\phi) & \text{for } i = h \end{cases} \)
- \( OV(G, 2, \phi) = 2 \sum_{i=1}^{h} s_i(\phi) \)
Lower bound – problem transformation

\[ OV(G, 2, \phi^*) = 56 \]

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( a_i )</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>( s_i )</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>14</td>
</tr>
</tbody>
</table>
Lower bound – problem transformation

\[ OV(G, 2, \phi^*) = 56 \]

<table>
<thead>
<tr>
<th>i</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_i )</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>( s_i )</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>14</td>
</tr>
</tbody>
</table>

\[ OV(G, 2, \phi) = 2(1 + 4 + 9 + 14) = 56 \]
Lower bound – problem transformation

- Given
  - an undirected graph $G = (V, E)$,
Lower bound – problem transformation

Given

- an undirected graph $G = (V, E)$,
- a constant $k \geq 2$, 

Lower bound – problem transformation

Given

- an undirected graph $G = (V, E)$,
- a constant $k \geq 2$,

the $k$-balanced partitioning problem (kBPP) asks for a partition of the vertex set $V$ into $k$ non-empty vertex sets.
Lower bound – problem transformation

Given

- an undirected graph $G = (V, E)$,
- a constant $k \geq 2$,

the **k-balanced partitioning problem (kBPP)** asks for a partition of the vertex set $V$ into $k$ non-empty vertex sets

- $V_1 \neq \emptyset$, $V_2 \neq \emptyset$, $\ldots$, $V_k \neq \emptyset$, where
- $\bigcup_{i=1}^{k} V_k = V$, $V_i \cap V_j = \emptyset$ for every $i \neq j$ and
- $|V_i| \leq \left\lceil \frac{n}{k} \right\rceil$ for all $1 \leq i \leq k$, 

...
Lower bound – problem transformation

Given

- an undirected graph $G = (V, E)$,
- a constant $k \geq 2$,

the **k-balanced partitioning problem (kBPP)** asks for a partition of the vertex set $V$ into $k$ non-empty vertex sets

- $V_1 \neq \emptyset$, $V_2 \neq \emptyset$, \ldots, $V_k \neq \emptyset$, where
- $\bigcup_{i=1}^{k} V_k = V$, $V_i \cap V_j = \emptyset$ for every $i \neq j$ and
- $|V_i| \leq \left\lceil \frac{n}{k} \right\rceil$ for all $1 \leq i \leq k$,

such that the number of edges connecting these vertex sets

$$c(G, \mathcal{V}) := \left| \{(u, v) \in E | u \in V_i, \ v \in V_j, \ i \neq j\} \right|,$$  \hspace{1cm} (3)

where $\mathcal{V} = \{V_i | 1 \leq i \leq k\}$, is minimised.
Lower bound – problem transformation

It is obvious that $s_i \geq c(G, V)$, where $k = 2h - i + 2$ for all $2 \leq i \leq h$ and that $s_1 = \left| E(G) \right|$. All but one components have the size $|V| + 1k$. One component has the size $|V| + 1k - 1$. 

<table>
<thead>
<tr>
<th>$i$</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>$s_i$</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>14</td>
</tr>
</tbody>
</table>
Lower bound – problem transformation

\[
\begin{array}{c|cccc}
   i & 4 & 3 & 2 & 1 \\
\hline
   a_i & 1 & 3 & 5 & 5 \\
   s_i & 1 & 4 & 9 & 14 \\
\end{array}
\]

\[\text{It is obvious that } s_i \geq c(G, \mathcal{V}), \text{ where } k = 2^{h-i+2} \text{ for all } 2 \leq i \leq h \text{ and that } s_1 = |E(G)|.\]
Lower bound – problem transformation

\[
\begin{array}{|c|c|c|c|}
\hline
i & 4 & 3 & 2 & 1 \\
\hline
a_i & 1 & 3 & 5 & 5 \\
\hline
s_i & 1 & 4 & 9 & 14 \\
\hline
\end{array}
\]

, \quad k = 2^{3-4+2} = 2

It is obvious that \( s_i \geq c(G, \mathcal{V}) \), where \( k = 2^{h-i+2} \) for all \( 2 \leq i \leq h \) and that \( s_1 = |E(G)| \).
Lower bound – problem transformation

\begin{tabular}{c|cccc}
  \(i\) & 4 & 3 & 2 & 1 \\
  \hline
  \(a_i\) & 1 & 3 & 5 & 5 \\
  \(s_i\) & 1 & 4 & 9 & 14 \\
\end{tabular}, \(k = 2^{3-3+2} = 4\)

It is obvious that \(s_i \geq c(G, V)\), where \(k = 2^{h-i+2}\) for all \(2 \leq i \leq h\) and that \(s_1 = |E(G)|\).
Lower bound – problem transformation

\[
\begin{array}{cccc}
 i & 4 & 3 & 2 \\
a_i & 1 & 3 & 5 \\
s_i & 1 & 4 & 9 \\
\end{array}
\]

\[
\begin{array}{cc}
 \text{, } k = 2^{3-2+2} = 8 \\
\end{array}
\]

\[s_i \geq c(G, \mathcal{W}), \text{ where } k = 2^{h-i+2} \text{ for all } 2 \leq i \leq h\]

\[s_1 = |E(G)|.\]
Lower bound – problem transformation

<table>
<thead>
<tr>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>$s_i$</td>
<td>1</td>
<td>9</td>
<td>4</td>
<td>14</td>
</tr>
</tbody>
</table>

It is obvious that $s_i \geq c(G, \mathcal{V})$, where $k = 2^{h-i+2}$ for all $2 \leq i \leq h$ and that $s_1 = |E(G)|$. 
Lower bound – problem transformation

\[ s_i \geq c(G, \mathcal{V}) \], where \( k = 2^{h-i+2} \) for all \( 2 \leq i \leq h \) and that \( s_1 = |E(G)| \).

- All but one components have the size \( |V| + 1 \).

\[
\begin{array}{c|cccc}
  i & 4 & 3 & 2 & 1 \\
  \hline
  a_i & 1 & 3 & 5 & 5 \\
  s_i & 1 & 4 & 9 & 14 \\
\end{array}
\]
Lower bound – problem transformation

\[
\begin{array}{c|cccc}
  i & 4 & 3 & 2 & 1 \\
\hline
  a_i & 1 & 3 & 5 & 5 \\
  s_i & 1 & 4 & 9 & 14 \\
\end{array}
\]

\[\text{It is obvious that } s_i \geq c(G, \mathcal{V}), \text{ where } k = 2^{h-i+2} \text{ for all } 2 \leq i \leq h \text{ and that } s_1 = |E(G)|.\]

\[\text{All but one components have the size } \frac{|V|+1}{k}.\]

\[\text{One component has the size } \frac{|V|+1}{k} - 1.\]
Lower bound – problem transformation

- kBPP is \( \mathcal{NP} \)-hard (we get the \textit{minimum bisection problem} which is \( \mathcal{NP} \)-hard for \( k = 2 \) [Garey, Johnson 2002\(^6\)].

Lower bound – problem transformation

- kBPP is $\mathcal{NP}$-hard (we get the minimum bisection problem which is $\mathcal{NP}$-hard for $k = 2$ [Garey, Johnson 2002$^6$]).
- Andreev and Räcke prove further complexity results for a generalization allowing near-balanced partitions [Andreev, Räcke 2006$^7$].

---


Lower bound – problem transformation

▶ **kBPP** is \(\mathcal{NP}\)-hard (we get the *minimum bisection problem* which is \(\mathcal{NP}\)-hard for \(k = 2\) [Garey, Johnson 2002\(^6\)]).

▶ **Andreev and Räcke** prove further complexity results for a generalization allowing near-balanced partitions [Andreev, Räcke 2006\(^7\)].

▶ **Krauthgamer, Naor and Schwartz** provide an approximation algorithm achieving an approximation of \(O(\sqrt{\log n \log k})\) [Krauthgamer, Naor, Schwartz 2009\(^8\)].

---


Algorithmic Graph Theory on the Adriatic Coast

Lower bound – problem transformation

- kBPP remains APX-hard even if the graph is an unweighted tree with constant maximum degree [Feldmann, Foschini 2013].

\[ c(G, V^*) = \left(3 \cdot 2^h + 1 - 2k' + 1\right) \left(\frac{1}{2} - 1 - 1(1 - 2 - s) 2^{s_l}\right) + 3 \cdot 2^h - s_l + 1 - 2, \]

where \( s = h - k' + 2 \) and \( l = \lfloor h + 1 \rfloor. \)

Lower bound – problem transformation

- $kBPP$ remains $\mathcal{APX}$-hard even if the graph is an unweighted tree with constant maximum degree [Feldmann, Foschini 2013$^9$].

**Theorem (Schauer and S.)**

Let $G = (V, E)$ be a binary regular tree of height $h \geq 1$ and let $k = 2^{k'}$, where $1 \leq k' \leq h$, and $\mathcal{V}^*$ an optimal $k$-balanced partition. Then

$$c(G, \mathcal{V}^*) = \left(3 \cdot 2^{h+1} - 2^{k'+1}\right) \left(\frac{1}{2^s - 1} - \frac{1}{(1 - 2^{-s})2^s l}\right) + 3 \cdot 2^{h-sl+1} - 2,$$

where $s = h - k' + 2$ and $l = \left\lfloor \frac{h+1}{s} \right\rfloor$.

### Lower bound – problem transformation

<table>
<thead>
<tr>
<th>$i$</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>$s_i$</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>14</td>
</tr>
<tr>
<td>$c(G, V^*)$</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>14</td>
</tr>
</tbody>
</table>
Lower bound – problem transformation

<table>
<thead>
<tr>
<th>$i$</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>$s_i$</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>14</td>
</tr>
<tr>
<td>$c(G, V^*)$</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>14</td>
</tr>
</tbody>
</table>

$OV(G, 2, \phi^*) = 56$

$\Rightarrow$ optimality in this case $\checkmark$
Lower bound – problem transformation

<table>
<thead>
<tr>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>$s_i$</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>14</td>
</tr>
<tr>
<td>$c(G, V^*)$</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>14</td>
</tr>
</tbody>
</table>

$\Rightarrow$ optimality in this case ✔

$OV(G, 2, \phi^*) = 56$

<table>
<thead>
<tr>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>$s_i$</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>$c(G, V^*)$</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>20</td>
</tr>
</tbody>
</table>

$\Rightarrow$ optimality in this case ✔
### Lower bound – problem transformation

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>$s_i$</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>14</td>
</tr>
<tr>
<td>$c(G, \mathcal{V}^*)$</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>14</td>
</tr>
</tbody>
</table>

$\Rightarrow$ optimality in this case ✓

$OV(G, 2, \phi^*) = 56$

<table>
<thead>
<tr>
<th></th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>$s_i$</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>$c(G, \mathcal{V}^*)$</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

$\Rightarrow$ optimality in this case ✓

$OV(G, 2, \phi^*) = 130$
## Lower bound – problem transformation

<table>
<thead>
<tr>
<th>$i$</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>12</td>
<td>19</td>
<td>21</td>
</tr>
<tr>
<td>$s_i$</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>22</td>
<td>41</td>
<td>62</td>
</tr>
<tr>
<td>$c(G, \psi^*)$</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>21</td>
<td>41</td>
<td>62</td>
</tr>
</tbody>
</table>
Lower bound – problem transformation

\[
\begin{array}{cccccc}
 i & 6 & 5 & 4 & 3 & 2 & 1 \\
 a_i & 1 & 3 & 6 & 12 & 19 & 21 \\
 s_i & 1 & 4 & 10 & 22 & 41 & 62 \\
 c(G, \mathcal{V}^*) & 1 & 4 & 10 & 21 & 41 & 62 \\
\end{array}
\]

\[278 \leq OV(G, 2, \phi^*) \leq 280\]
Lower bound – problem transformation

\[
\begin{array}{cccccc}
 i & 6 & 5 & 4 & 3 & 2 & 1 \\
 a_i & 1 & 3 & 6 & 12 & 19 & 21 \\
 s_i & 1 & 4 & 10 & 22 & 41 & 62 \\
 c(G, \mathcal{V}^*) & 1 & 4 & 10 & 21 & 41 & 62 \\
\end{array}
\]

\[278 \leq OV(G, 2, \phi^*) \leq 280\]

In fact, the lower bound is tight for all \( \left\lceil \frac{h}{2} \right\rceil + 1 \leq i \leq h \) and for \( i = 1 \) and \( i = 2 \).
Lower bound – problem transformation

<table>
<thead>
<tr>
<th></th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i)</td>
<td>(a_i)</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>12</td>
<td>19</td>
</tr>
<tr>
<td>(s_i)</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>22</td>
<td>41</td>
<td>62</td>
</tr>
<tr>
<td>(c(G, \mathcal{V}^*))</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>21</td>
<td>41</td>
<td>62</td>
</tr>
</tbody>
</table>

\(\Rightarrow\) problem \(X\)

\(278 \leq OV(G, 2, \phi^*) \leq 280\)

In fact, the lower bound is tight for all \(\left\lfloor \frac{h}{2} \right\rfloor + 1 \leq i \leq h\) and for \(i = 1\) and \(i = 2\).

A straightforward analysis yields an approximation ratio 2.
Lower bound – problem transformation

<table>
<thead>
<tr>
<th>i</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>12</td>
<td>19</td>
<td>21</td>
</tr>
<tr>
<td>$s_i$</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>22</td>
<td>41</td>
<td>62</td>
</tr>
<tr>
<td>$c(G, V^*)$</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>21</td>
<td>41</td>
<td>62</td>
</tr>
</tbody>
</table>

$\Rightarrow$ problem $\mathcal{X}$

$278 \leq OV(G, 2, \phi^*) \leq 280$

In fact, the lower bound is tight for all $\left\lfloor \frac{h}{2} \right\rfloor + 1 \leq i \leq h$ and for $i = 1$ and $i = 2$.

A straightforward analysis yields an approximation ratio 2.

The empirical gap between the lower and the upper bound does not exceed 1.1.
Lower bound – problem transformation
Lower bound – problem transformation

\[ c(G, k, \mathcal{V}) = 10 \]
Lower bound – problem transformation

\[ c(G, k, V) = 10 + 12 = 22 \]
Lower bound – problem transformation

\[ c(G, k, \mathcal{V}) = 21 \]
Lower bound – problem transformation

- We cannot reach the lower bound in general (the solution yielded by our algorithm for $h_G = 5$ and $h = 6$ is optimal).
Lower bound – problem transformation

- We cannot reach the lower bound in general (the solution yielded by our algorithm for $h_G = 5$ and $h = 6$ is optimal).
- The presented algorithm does not yield an optimal solution for larger guest graphs (there exists a counterexample e.g. for $h_G = 6$ and $h = 7$).
We cannot reach the lower bound in general (the solution yielded by our algorithm for $h_G = 5$ and $h = 6$ is optimal).

The presented algorithm does not yield an optimal solution for larger guest graphs (there exists a counterexample e.g. for $h_G = 6$ and $h = 7$).

It would be necessary to improve both the algorithm and the lower bound in order to reach the optimum.
Lower bound – problem transformation

- Given
  - an undirected graph \( G = (V, E) \),
Lower bound – problem transformation

- Given
  - an undirected graph \( G = (V, E) \),
  - a constant \( k' \leq \lceil \log_2 n \rceil - 1 \),
Lower bound – problem transformation

- Given
  - an undirected graph \( G = (V, E) \),
  - a constant \( k' \leq \lceil \log_2 n \rceil - 1 \),

a set \( \mathcal{V} = \{ \mathcal{V}(1), \mathcal{V}(2), \ldots, \mathcal{V}(k') \} \), where

\[ \mathcal{V}(j) = \{ V_1^{(j)}, V_2^{(j)}, \ldots, V_{2^j}^{(j)} \} \]

for all \( 1 \leq j \leq k' \), is called a hereditary family of power-two-cuts, iff the following two properties are fulfilled:
Lower bound – problem transformation

Given

- an undirected graph \( G = (V, E) \),
- a constant \( k' \leq \lceil \log_2 n \rceil - 1 \),

a set \( \mathcal{V} = \{ \mathcal{V}^{(1)}, \mathcal{V}^{(2)}, \ldots, \mathcal{V}^{(k')} \} \), where

\( \mathcal{V}^{(j)} = \{ V_1^{(j)}, V_2^{(j)}, \ldots, V_{2^j}^{(j)} \} \) for all \( 1 \leq j \leq k' \), is called a

**hereditary family of power-two-cuts**, iff the following two properties are fulfilled:

- \( \mathcal{V}^{(j)} \) is a \( 2^j \)-balanced partition of \( G \) for all \( 1 \leq j \leq k' \).
Lower bound – problem transformation

- Given
  - an undirected graph \( G = (V, E) \),
  - a constant \( k' \leq \lceil \log_2 n \rceil - 1 \),

a set \( \mathcal{V} = \{\mathcal{V}(1), \mathcal{V}(2), \ldots, \mathcal{V}(k')\} \), where

\[
\mathcal{V}(j) = \{V_1^{(j)}, V_2^{(j)}, \ldots, V_{2^j}^{(j)}\} \text{ for all } 1 \leq j \leq k',
\]

is called a hereditary family of power-two-cuts, iff the following two properties are fulfilled:

- \( \mathcal{V}(j) \) is a \( 2^j \)-balanced partition of \( G \) for all \( 1 \leq j \leq k' \).
- For any \( 1 \leq j \leq k' - 1 \) and any \( 1 \leq i \leq 2^j \), \( V_i^{(j)} \) is given as the union of 2 subsets among \( V_1^{(j+1)}, V_2^{(j+1)}, \ldots, V_{2^{j+1}}^{(j+1)} \).
Lower bound – problem transformation

- Given
  - an undirected graph $G = (V, E)$,
  - a constant $k' \leq \lceil \log_2 n \rceil - 1$,

a set $\mathcal{V} = \{\mathcal{V}(1), \mathcal{V}(2), \ldots, \mathcal{V}(k')\}$, where

$\mathcal{V}(j) = \{V_{1}^{(j)}, V_{2}^{(j)}, \ldots, V_{2^j}^{(j)}\}$ for all $1 \leq j \leq k'$, is called a hereditary family of power-two-cuts, iff the following two properties are fulfilled:

- $\mathcal{V}(j)$ is a $2^j$-balanced partition of $G$ for all $1 \leq j \leq k'$.
- For any $1 \leq j \leq k' - 1$ and any $1 \leq i \leq 2^j$, $V_{i}^{(j)}$ is given as the union of 2 subsets among $V_{1}^{(j+1)}, V_{2}^{(j+1)}, \ldots, V_{2^{j+1}}^{(j+1)}$.

- The $k$-balanced partitioning problem into a hereditary family of power-two-cuts (kBPPH) asks for a hereditary family which minimises the objective value

$\mathcal{V}(1), \mathcal{V}(2), \ldots, \mathcal{V}(k')$. 

$(5)$
Lower bound – problem transformation

- Given
  - an undirected graph $G = (V, E)$,
  - a constant $k' \leq \lceil \log_2 n \rceil - 1$,

  a set $\mathcal{V} = \{\mathcal{V}^{(1)}, \mathcal{V}^{(2)}, \ldots, \mathcal{V}^{(k')}\}$, where

  $\mathcal{V}^{(j)} = \{V^{(j)}_1, V^{(j)}_2, \ldots, V^{(j)}_{2^j}\}$ for all $1 \leq j \leq k'$, is called a hereditary family of power-two-cuts, iff the following two properties are fulfilled:

  - $\mathcal{V}^{(j)}$ is a $2^j$-balanced partition of $G$ for all $1 \leq j \leq k'$.
  - For any $1 \leq j \leq k' - 1$ and any $1 \leq i \leq 2^j$, $V^{(j)}_i$ is given as the union of 2 subsets among $V^{(j+1)}_1, V^{(j+1)}_2, \ldots, V^{(j+1)}_{2^j+1}$.

- The $k$-balanced partitioning problem into a hereditary family of power-two-cuts ($kBPPH$) asks for a hereditary family which minimises the objective value

$$c^H(G, \mathcal{V}) = \sum_{j=1}^{k'} c(G, \mathcal{V}^{(j)}). \quad (5)$$
Lower bound – problem transformation

- $k$BPPH is $\mathcal{NP}$-hard (we get the *minimum bisection problem* which is $\mathcal{NP}$-hard for $k' = 1$ [Garey, Johnson 2002$^{10}$]).

---

Lower bound – problem transformation

- kBPPH is $\mathcal{NP}$-hard (we get the *minimum bisection problem* which is $\mathcal{NP}$-hard for $k' = 1$ [Garey, Johnson 2002\textsuperscript{10}]).
- The question about the computational complexity in our special case is open.

Recapitulation, future research and open questions

- We provide an approximation algorithm for one (very) special case of the DAPT.
Recapitulation, future research and open questions

▶ We provide an approximation algorithm for one (very) special case of the DAPT.
▶ Moreover, we know that neither the algorithm nor the lower bound can reach the optimum in general.
Recapitulation, future research and open questions

- We provide an approximation algorithm for one (very) special case of the DAPT.
- Moreover, we know that neither the algorithm nor the lower bound can reach the optimum in general.
- We would like either to improve the algorithm and the lower bound in order to obtain an exact solution algorithm or to prove that the problem is $\mathcal{NP}$-hard.
Recapitulation, future research and open questions

- We provide an approximation algorithm for one (very) special case of the DAPT.
- Moreover, we know that neither the algorithm nor the lower bound can reach the optimum in general.
- We would like either to improve the algorithm and the lower bound in order to obtain an exact solution algorithm or to prove that the problem is \( NP \)-hard.
- We would like to generalize the algorithm (already done) and the lower bound (to do) for trees of any constant degree \( d \geq 2 \).
Recapitulation, future research and open questions

▶ We provide an approximation algorithm for one (very) special case of the DAPT.

▶ Moreover, we know that neither the algorithm nor the lower bound can reach the optimum in general.

▶ We would like either to improve the algorithm and the lower bound in order to obtain an exact solution algorithm or to prove that the problem is \( \mathcal{NP} \)-hard.

▶ We would like to generalize the algorithm (already done) and the lower bound (to do) for trees of any constant degree \( d \geq 2 \).

Thank you for your attention!