Edge Coloring with Minimum Reload/Changeover Costs

Didem Gözüpek[1] Mordechai Shalom[2,3]

[1]Department of Computer Engineering, Gebze Technical University, Kocaeli, Turkey

[2]TelHai Academic College, Upper Galilee, 12210, Israel

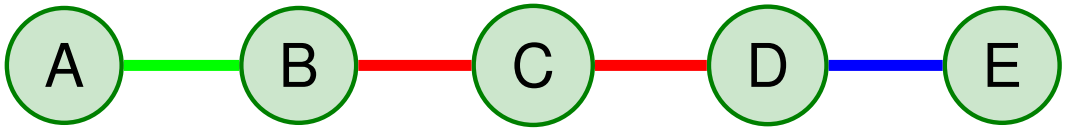
[3]Department of Computer Engineering, Bogazici University, Istanbul, Turkey

Algorithmic Graph Theory on the Adriatic Coast (AGTAC), 2015

Outline

What is traversal cost?

- Traversal cost refers to the cost that occurs when two consecutive edges along a path are of different colors



- Introduced in the seminal paper (under the name of reload cost): Wirth, H.C. and Steffan, J., *Reload cost problems: minimum diameter spanning tree*, Discrete Applied Mathematics, vol.113, pp.73-85, 2001.

Motivation and Applications

- Cost of (un)loading cargo from one carrier to another in intermodal cargo transportation networks
- Cost of losses in transferring energy in energy distribution networks
- Telecommunication networks that incorporate different technologies
- Switching from one frequency to another frequency has a non-negligible cost in ad hoc dynamic spectrum access (cognitive radio) networks

Differences Between Traversal Cost, Reload Cost and Changeover Cost

- Given a graph $G = (V(G), E(G))$, we consider proper edge colorings $\chi : E(G) \to X$ of G where the colors are taken from a set X
- The traversal costs are given by a nonnegative function $tc : X^2 \to \mathbb{R}^+ \cup \{0\}$ satisfying
  i) $tc(i, j) = tc(j, i)$ for every $i, j \in X$, and
  ii) $tc(i, i) = 0$ for every $i \in X$.

Differences Between Traversal Cost, Reload Cost and Changeover Cost

- Given a graph $G = (V(G), E(G))$, we consider proper edge colorings $\chi : E(G) \to X$ of G where the colors are taken from a set X
- The traversal costs are given by a nonnegative function $tc : X^2 \to \mathbb{R}^+ \cup \{0\}$ satisfying
  i) $tc(i, j) = tc(j, i)$ for every $i, j \in X$, and
  ii) $tc(i, i) = 0$ for every $i \in X$.

Differences Between Traversal Cost, Reload Cost and Changeover Cost

- Given a graph $G = (V(G), E(G))$, we consider proper edge colorings $\chi : E(G) \to X$ of G where the colors are taken from a set X
- The traversal costs are given by a nonnegative function $tc : X^2 \to \mathbb{R}^+ \cup \{0\}$ satisfying
  i) $tc(i, j) = tc(j, i)$ for every $i, j \in X$, and
  ii) $tc(i, i) = 0$ for every $i \in X$.

Differences Between Traversal Cost, Reload Cost and Changeover Cost

Given a set of paths,

- traversal cost of a path is the sum of the traversal costs at each vertex along the path
- Total reload cost is the sum of the total traversal costs of all paths
- With changeover cost, the cost of traversing a vertex by using two specific edges is paid only once, regardless of the number of paths traversing it

Differences Between Traversal Cost, Reload Cost and Changeover Cost

Given a set of paths,

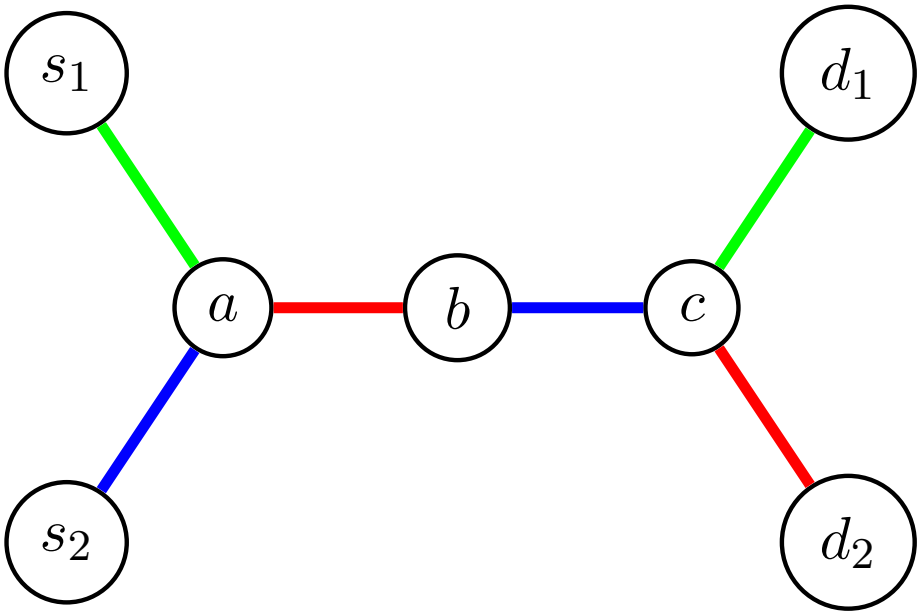
- traversal cost of a path is the sum of the traversal costs at each vertex along the path
- Total reload cost is the sum of the total traversal costs of all paths
- With changeover cost, the cost of traversing a vertex by using two specific edges is paid only once, regardless of the number of paths traversing it

Differences Between Traversal Cost, Reload Cost and Changeover Cost

Given a set of paths,

- traversal cost of a path is the sum of the traversal costs at each vertex along the path
- Total reload cost is the sum of the total traversal costs of all paths
- With changeover cost, the cost of traversing a vertex by using two specific edges is paid only once, regardless of the number of paths traversing it

Differences Between Traversal Cost, Reload Cost and Changeover Cost



Changeover cost=$tc(g,r) + tc(b,r) + tc(r,b) + tc(b,g) + tc(b,r)$

Reload cost=$tc(g,r) + tc(b,r) + 2tc(r,b) + tc(b,g) + tc(b,r)$

Previous work

- Minimum reload cost diameter problem [WS01, G08]
- Minimum reload cost cycle cover problem [GGM14]
- Minimum changeover cost arborescence problem [GGM11, GVSZ14]
- Reload cost path, tour, and flow problems [GLMM09]
- All of these problems focus on edge-colored graphs, where the coloring is given as input
- This work is the first one that focuses on proper edge coloring within the traversal cost concept

Previous work

- Minimum reload cost diameter problem [WS01, G08]
- Minimum reload cost cycle cover problem [GGM14]
- Minimum changeover cost arborescence problem [GGM11, GVSZ14]
- Reload cost path, tour, and flow problems [GLMM09]
- All of these problems focus on edge-colored graphs, where the coloring is given as input
- This work is the first one that focuses on proper edge coloring within the traversal cost concept

Previous work

- Minimum reload cost diameter problem [WS01, G08]
- Minimum reload cost cycle cover problem [GGM14]
- Minimum changeover cost arborescence problem [GGM11, GVSZ14]
- Reload cost path, tour, and flow problems [GLMM09]
- All of these problems focus on edge-colored graphs, where the coloring is given as input
- This work is the first one that focuses on proper edge coloring within the traversal cost concept

Minimum Reload/Changeover Cost Edge Coloring (MinRCEC/MinCCEC) Problems

MinRCEC/MinCCEC $(G, \mathcal{P}, X, tc)$

- **Input:** A set of paths $\mathcal{P}$ constituting a graph $G = \cup\mathcal{P}$, a set X of at least $\Delta(G) + 1$ colors, a traversal cost function $tc : X^2 \rightarrow \mathbb{R}^+ \cup \{0\}$.
- **Output:** A proper edge coloring $\chi : E(G) \rightarrow X$
- **Objective:** Minimize the total changeover/reload cost of all paths.

Minimum Reload Cost Path Tree Edge Coloring (MinRCPTEC) and Minimum Changeover Cost Arborescence Edge Coloring (MinCCAEC) Problems

MinRCPTEC/MinCCAEC (G, r, X, tc)

- **Input:** A graph G, a vertex r of G, a set X of at least $\Delta(G) + 1$ colors, a traversal cost function $tc : X^2 \to \mathbb{R}^+ \cup \{0\}$
- **Output:** A spanning tree T of G and a proper edge coloring $\chi : E(T) \mapsto X$
- **Objective:** Minimize the total changeover/reload cost of the spanning tree rooted at r.

Approximation Algorithms

- Given a minimization problem Π, ALG is a ρ-*approximation algorithm* for Π (with $\rho \geq 1$) if for any instance I of Π, $ALG(I) \leq \rho \cdot OPT(I)$
- Given a real function f, Π is said to be in f-APX-Hard if there is a constant $c > 0$ such that Π is $(c \cdot f(|I|))$-inapproximable where $|I|$ is the size of the instance I
- When f is a constant, this complexity class is called simply APX-Hard
- A polynomial-time approximation scheme (PTAS) is an infinite family of algorithms $\{ALG_\epsilon | \epsilon > 0\}$ such that ALG_ϵ is a $(1+\epsilon)$-approximation algorithm with running time $O(|I|^{f(\epsilon)})$ for some function f.

Approximation Algorithms

- Given a minimization problem Π, ALG is a ρ-*approximation algorithm* for Π (with $\rho \geq 1$) if for any instance I of Π, $ALG(I) \leq \rho \cdot OPT(I)$
- Given a real function f, Π is said to be in f-APX-Hard if there is a constant $c > 0$ such that Π is $(c \cdot f(|I|))$-inapproximable where $|I|$ is the size of the instance I
- When f is a constant, this complexity class is called simply APX-Hard
- A polynomial-time approximation scheme (PTAS) is an infinite family of algorithms $\{ALG_\epsilon | \epsilon > 0\}$ such that ALG_ϵ is a $(1+\epsilon)$-approximation algorithm with running time $O(|I|^{f(\epsilon)})$ for some function f.

Approximation Algorithms

- Given a minimization problem Π, ALG is a *ρ-approximation algorithm* for Π (with $\rho \geq 1$) if for any instance I of Π, $ALG(I) \leq \rho \cdot OPT(I)$
- Given a real function f, Π is said to be in f-APX-Hard if there is a constant $c > 0$ such that Π is $(c \cdot f(|I|))$-inapproximable where $|I|$ is the size of the instance I
- When f is a constant, this complexity class is called simply APX-Hard
- A polynomial-time approximation scheme (PTAS) is an infinite family of algorithms $\{ALG_\epsilon | \epsilon > 0\}$ such that ALG_ϵ is a $(1+\epsilon)$-approximation algorithm with running time $O(|I|^{f(\epsilon)})$ for some function f.

Approximation Algorithms

- Given a minimization problem Π, ALG is a ρ-*approximation algorithm* for Π (with $\rho \geq 1$) if for any instance I of Π, $ALG(I) \leq \rho \cdot OPT(I)$
- Given a real function f, Π is said to be in f-APX-Hard if there is a constant $c > 0$ such that Π is $(c \cdot f(|I|))$-inapproximable where $|I|$ is the size of the instance I
- When f is a constant, this complexity class is called simply APX-Hard
- A polynomial-time approximation scheme (PTAS) is an infinite family of algorithms $\{ALG_\epsilon | \epsilon > 0\}$ such that ALG_ϵ is a $(1+\epsilon)$-approximation algorithm with running time $O(|I|^{f(\epsilon)})$ for some function f.

K-LIGHTEST SUBGRAPH Problem

- Given an edge weighted graph G, the K-LIGHTEST SUBGRAPH problem is to find an induced subgraph of G on k vertices, with minimum total edge weight.
- K-LIGHTEST SUBGRAPH problem is NP-Hard in the strong sense even when the graph is a complete graph and the edge weights are either 1 or 2

Minimum Set Cover Problem

MinimumSetCover

- **Input:** A pair $(U, \mathcal{S})$ where $U = \{u_1, u_2, \ldots, u_n\}$ is a finite ground set of elements, and $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ is a collection of subsets of U.
- **Output:** A subset $\mathcal{C} \subseteq \mathcal{S}$ that covers U, i.e. $\cup\mathcal{C} \stackrel{def}{=} \cup_{S_i \in \mathcal{C}} S_i = U$
- **Objective:** Minimize $|\mathcal{C}|$

The special case where each set has cardinality at most 3 and each element appears in at most 2 sets is called Min3SC2, which is APX-Hard.

Minimum Set Cover Problem

MinimumSetCover

- **Input:** A pair $(U, \mathcal{S})$ where $U = \{u_1, u_2, \ldots, u_n\}$ is a finite ground set of elements, and $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ is a collection of subsets of U.
- **Output:** A subset $\mathcal{C} \subseteq \mathcal{S}$ that covers U, i.e. $\cup\mathcal{C} \stackrel{def}{=} \cup_{S_i \in \mathcal{C}} S_i = U$
- **Objective:** Minimize $|\mathcal{C}|$

The special case where each set has cardinality at most 3 and each element appears in at most 2 sets is called Min3SC2, which is APX-Hard.

Theorem

MINCCEC *and* MINRCEC *are inapproximable within any polynomial-time computable function* $f(|\mathcal{P}|)$.

Proof.

(Sketch) Chromatic index of a graph is either $\Delta(G)$ or $\Delta(G)+1$ and it is NP-Complete to decide between these two values. Construct an instance where:

- The set of paths $\mathcal{P}$ consists of all distinct paths of length 2.
- There are $\Delta(G)+1$ colors, where one color is very expensive and all the rest are cheap (has cost 1)

We have a very large changeover/reload cost value if and only if the graph is $\Delta(G)+1$ edge colorable $\square$

Theorem

MINCCEC *and* MINRCEC *are inapproximable within any polynomial-time computable function* $f(|\mathcal{P}|)$.

Proof.

(Sketch) Chromatic index of a graph is either $\Delta(G)$ or $\Delta(G)+1$ and it is NP-Complete to decide between these two values.

Construct an instance where:

- The set of paths $\mathcal{P}$ consists of all distinct paths of length 2.
- There are $\Delta(G)+1$ colors, where one color is very expensive and all the rest are cheap (has cost 1)

We have a very large changeover/reload cost value if and only if the graph is $\Delta(G)+1$ edge colorable $\square$

Theorem

MINCCEC *and* MINRCEC *are inapproximable within any polynomial-time computable function* $f(|\mathcal{P}|)$.

Proof.

(Sketch) Chromatic index of a graph is either $\Delta(G)$ or $\Delta(G)+1$ and it is NP-Complete to decide between these two values. Construct an instance where:

- The set of paths $\mathcal{P}$ consists of all distinct paths of length 2.
- There are $\Delta(G)+1$ colors, where one color is very expensive and all the rest are cheap (has cost 1)

We have a very large changeover/reload cost value if and only if the graph is $\Delta(G)+1$ edge colorable □

Theorem

MINCCEC *and* MINRCEC *are inapproximable within any polynomial-time computable function* $f(|\mathcal{P}|)$.

Proof.

(Sketch) Chromatic index of a graph is either $\Delta(G)$ or $\Delta(G)+1$ and it is NP-Complete to decide between these two values. Construct an instance where:

- The set of paths $\mathcal{P}$ consists of all distinct paths of length 2.
- There are $\Delta(G)+1$ colors, where one color is very expensive and all the rest are cheap (has cost 1)

We have a very large changeover/reload cost value if and only if the graph is $\Delta(G)+1$ edge colorable □

Theorem

MINCCEC *and* MINRCEC *are* NP-$Hard$ *in the strong sense even when* $tc(i,j) \in \{1,2\}$ *for every distinct pair* i,j *and* G *is a star.*

Proof.

(Sketch) Given an instance of K-LIGHTEST SUBGRAPH problem with a clique K on more than k vertices and the edge weight function w,

build an instance of MINCCEC (or MINRCEC) where

- G is a star on $k+1$ vertices
- We have all $\binom{k}{2}$ paths between every pair of leaves
- $|X| = |K|$ and $tc(i,j) = w(i,j)$

Theorem

MINCCEC *and* MINRCEC *are* NP-$Hard$ *in the strong sense even when* $tc(i,j) \in \{1,2\}$ *for every distinct pair* i,j *and* G *is a star.*

Proof.

(Sketch) Given an instance of K-LIGHTEST SUBGRAPH problem with a clique K on more than k vertices and the edge weight function w,
build an instance of MINCCEC (or MINRCEC) where

- G is a star on $k+1$ vertices
- We have all $\binom{k}{2}$ paths between every pair of leaves
- $|X| = |K|$ and $tc(i,j) = w(i,j)$

Theorem

MINCCEC *and* MINRCEC *are* NP-$Hard$ *in the strong sense even when* $tc(i,j) \in \{1,2\}$ *for every distinct pair* i,j *and* G *is a star.*

Proof.

(Sketch) Given an instance of K-LIGHTEST SUBGRAPH problem with a clique K on more than k vertices and the edge weight function w,
build an instance of MINCCEC (or MINRCEC) where

- G is a star on $k+1$ vertices
- We have all $\begin{pmatrix} k \\ 2 \end{pmatrix}$ paths between every pair of leaves
- $|X| = |K|$ and $tc(i,j) = w(i,j)$

Theorem

MINCCEC *and* MINRCEC *are* NP-$Hard$ *in the strong sense even when* $tc(i,j) \in \{1,2\}$ *for every distinct pair* i,j *and* G *is a star.*

Proof.

(Sketch) Given an instance of K-LIGHTEST SUBGRAPH problem with a clique K on more than k vertices and the edge weight function w,
build an instance of MINCCEC (or MINRCEC) where

- G is a star on $k+1$ vertices
- We have all $\begin{pmatrix} k \\ 2 \end{pmatrix}$ paths between every pair of leaves
- $|X| = |K|$ and $tc(i,j) = w(i,j)$

Theorem

MINCCEC *and* MINRCEC *are* NP-$Hard$ *in the strong sense even when* $tc(i,j) \in \{1,2\}$ *for every distinct pair* i,j *and* G *is a star.*

Proof.

(Sketch) Given an instance of K-LIGHTEST SUBGRAPH problem with a clique K on more than k vertices and the edge weight function w,
build an instance of MINCCEC (or MINRCEC) where

- G is a star on $k+1$ vertices
- We have all $\begin{pmatrix} k \\ 2 \end{pmatrix}$ paths between every pair of leaves
- $|X| = |K|$ and $tc(i,j) = w(i,j)$

Theorem

MINCCAEC *and* MINRCPTEC *are* APX-$Hard$ *in directed graphs even when* $tc(i,j) \in \{1,2\}$ *for every distinct pair* i,j*.*

Proof.

(Sketch) By reduction from the MIN3SC2 problem, which is APX-Hard

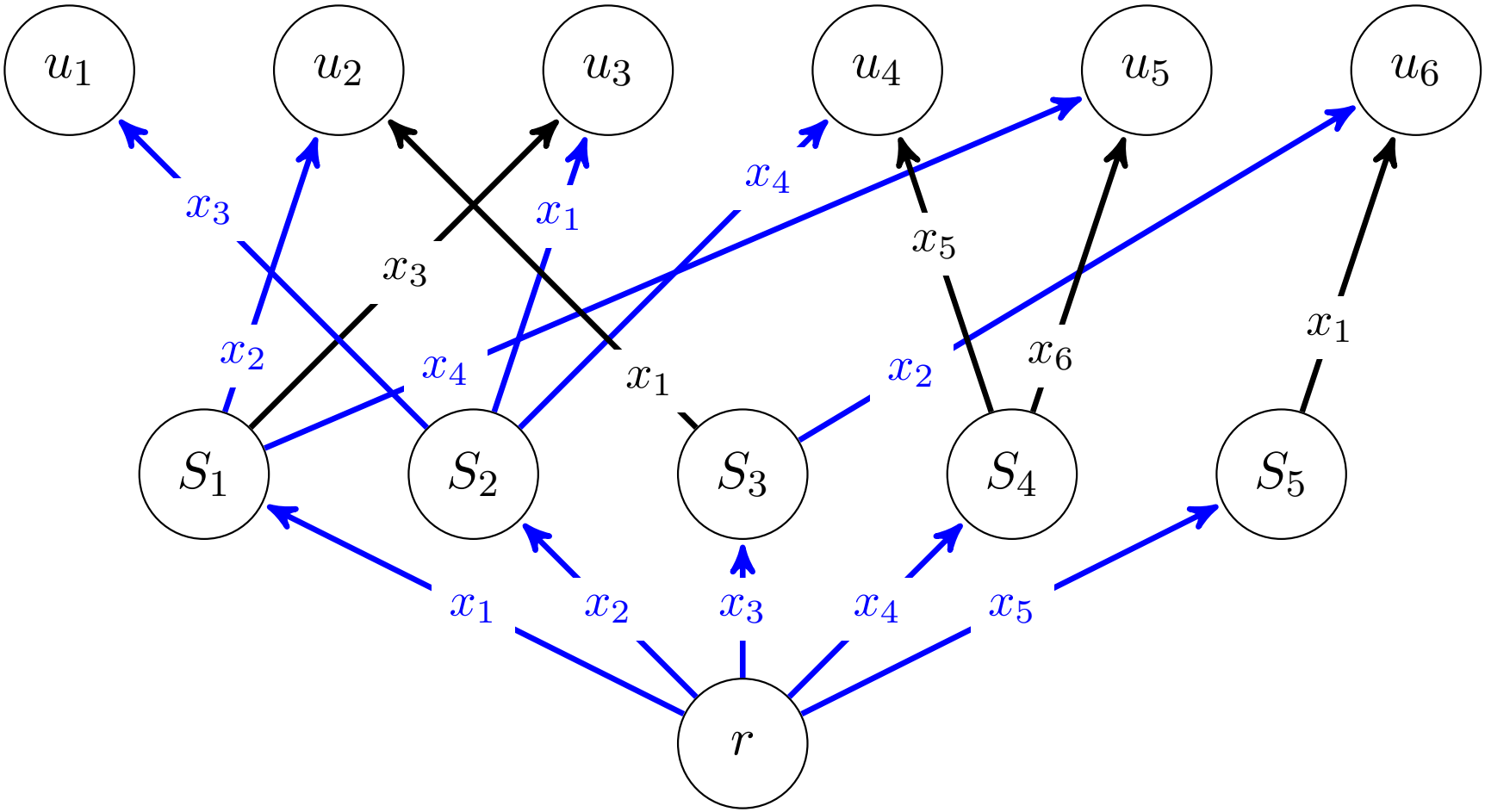
Given an instance $\mathcal{S}$ of MIN3SC2 with n elements and $m \geq 4$ sets and an integer $k \leq m+1$, we construct an instance of MINCCAEC as follows: □

Theorem

MINCCAEC *and* MINRCPTEC *are* APX-$Hard$ *in directed graphs even when* $tc(i,j) \in \{1,2\}$ *for every distinct pair* i,j*.*

Proof.

(Sketch) By reduction from the MIN3SC2 problem, which is APX-Hard
Given an instance $\mathcal{S}$ of MIN3SC2 with n elements and $m \geq 4$ sets and an integer $k \leq m+1$, we construct an instance of MINCCAEC as follows: □



$X = X_c \cup X_e$ where $|X_c| = k$ and $|X_e| = \Delta(G) + 1 - k$.
Here, we have $X_c = \{x_1, x_2, x_3, x_4\}$ and $X_e = \{x_5, x_6\}$

- $tc(x, y) = 1$ if $x, y \in X_c$ and 2 otherwise
- Any feasible solution of $I(\mathcal{S}, k)$ induces a set cover $\mathcal{C}(T)$ that corresponds to the set of parents of the vertices U in T
- We partition $\mathcal{C}(T)$ into two sets:
  - $\mathcal{C}_c(T) = \{S_i \in \mathcal{C}(T) |\ \chi(r, S_i) \in X_c\}$ and
  - $\mathcal{C}_e(T) = \{S_i \in \mathcal{C}(T) |\ \chi(r, S_i) \in X_e\}$
- Observe that $rc_\chi(T, r) = cc_\chi(T, r)$
- $cc_\chi(T, r) \geq n + |\mathcal{C}_e(T)|$ since the arc leading to u_i in T incurs a traversal cost of at least 1 in the parent S_j of u_i, and an additional traversal cost of 1 if S_j is in $\mathcal{C}_e(T)$.
- $|\mathcal{C}_c(T)| \leq k$ since χ is a one-to-one function from the incoming arcs of $\mathcal{C}_c(T)$ into X_c and $|X_c| = k$.

- $tc(x, y) = 1$ if $x, y \in X_c$ and 2 otherwise
- Any feasible solution of $I(\mathcal{S}, k)$ induces a set cover $\mathcal{C}(T)$ that corresponds to the set of parents of the vertices U in T
- We partition $\mathcal{C}(T)$ into two sets:
  - $\mathcal{C}_c(T) = \{S_i \in \mathcal{C}(T) |\ \chi(r, S_i) \in X_c\}$ and
  - $\mathcal{C}_e(T) = \{S_i \in \mathcal{C}(T) |\ \chi(r, S_i) \in X_e\}$
- Observe that $rc_\chi(T, r) = cc_\chi(T, r)$
- $cc_\chi(T, r) \geq n + |\mathcal{C}_e(T)|$ since the arc leading to u_i in T incurs a traversal cost of at least 1 in the parent S_j of u_i, and an additional traversal cost of 1 if S_j is in $\mathcal{C}_e(T)$.
- $|\mathcal{C}_c(T)| \leq k$ since χ is a one-to-one function from the incoming arcs of $\mathcal{C}_c(T)$ into X_c and $|X_c| = k$.

- $tc(x, y) = 1$ if $x, y \in X_c$ and 2 otherwise
- Any feasible solution of $I(\mathcal{S}, k)$ induces a set cover $\mathcal{C}(T)$ that corresponds to the set of parents of the vertices U in T
- We partition $\mathcal{C}(T)$ into two sets:
  - $\mathcal{C}_c(T) = \{S_i \in \mathcal{C}(T) |\ \chi(r, S_i) \in X_c\}$ and
  - $\mathcal{C}_e(T) = \{S_i \in \mathcal{C}(T) |\ \chi(r, S_i) \in X_e\}$
- Observe that $rc_\chi(T, r) = cc_\chi(T, r)$
- $cc_\chi(T, r) \geq n + |\mathcal{C}_e(T)|$ since the arc leading to u_i in T incurs a traversal cost of at least 1 in the parent S_j of u_i, and an additional traversal cost of 1 if S_j is in $\mathcal{C}_e(T)$.
- $|\mathcal{C}_c(T)| \leq k$ since χ is a one-to-one function from the incoming arcs of $\mathcal{C}_c(T)$ into X_c and $|X_c| = k$.

- $tc(x, y) = 1$ if $x, y \in X_c$ and 2 otherwise
- Any feasible solution of $I(\mathcal{S}, k)$ induces a set cover $\mathcal{C}(T)$ that corresponds to the set of parents of the vertices U in T
- We partition $\mathcal{C}(T)$ into two sets:
  - $\mathcal{C}_c(T) = \{S_i \in \mathcal{C}(T) |\ \chi(r, S_i) \in X_c\}$ and
  - $\mathcal{C}_e(T) = \{S_i \in \mathcal{C}(T) |\ \chi(r, S_i) \in X_e\}$
- Observe that $rc_\chi(T, r) = cc_\chi(T, r)$
- $cc_\chi(T, r) \geq n + |\mathcal{C}_e(T)|$ since the arc leading to u_i in T incurs a traversal cost of at least 1 in the parent S_j of u_i, and an additional traversal cost of 1 if S_j is in $\mathcal{C}_e(T)$.
- $|\mathcal{C}_c(T)| \leq k$ since χ is a one-to-one function from the incoming arcs of $\mathcal{C}_c(T)$ into X_c and $|X_c| = k$.

- Assume that Algorithm A is a PTAS for one of the problems MINCCAEC or MINRCPTEC. We claim that the following algorithm, which runs A for every $k \in [m]$ and returns the minimum among all the set covers implied by the solutions is a PTAS for MIN3SC2 (contradiction)

Algorithm 1 PTAS for MIN3SC2

An instance $\mathcal{S}$ of MIN3SC2 and $\epsilon > 0$

- $\epsilon' \leftarrow \epsilon/3$
- $\mathcal{C} = \mathcal{S}$
- **for** k = 1 **to** $|\mathcal{S}|$ **do**
  - $(T, \chi) \leftarrow A_{\epsilon'}(I(\mathcal{S}, k))$
- **if** $|\mathcal{C}(T)| < |\mathcal{C}|$ **then**
  - $\mathcal{C} \leftarrow \mathcal{C}(T)$
- **Return** $\mathcal{C}$

- Assume that Algorithm A is a PTAS for one of the problems MINCCAEC or MINRCPTEC. We claim that the following algorithm, which runs A for every $k \in [m]$ and returns the minimum among all the set covers implied by the solutions is a PTAS for MIN3SC2 (contradiction)

Algorithm 2 PTAS for MIN3SC2

An instance $\mathcal{S}$ of MIN3SC2 and $\epsilon > 0$

- $\epsilon' \leftarrow \epsilon/3$
  $\mathcal{C} = \mathcal{S}$
  for k = 1 **to** $|\mathcal{S}|$ **do**
  $(T, \chi) \leftarrow A_{\epsilon'}(I(\mathcal{S}, k))$
  if $|\mathcal{C}(T)| < |\mathcal{C}|$ **then**
  $\mathcal{C} \leftarrow \mathcal{C}(T)$
  Return $\mathcal{C}$

- Let $\mathcal{C}^*$ be a minimum set cover of $\mathcal{S}$
- Let $(\hat{T}, \hat{\chi})$ be the solution returned by $A_{\epsilon'}$ on input $I(\mathcal{S}, |\mathcal{C}^*|)$.
- Recall that $\mathcal{C}_c(\hat{T}) \leq |\mathcal{C}^*|$. We now bound $\mathcal{C}_e(\hat{T})$
- We know that $A_{\epsilon'}(I(\mathcal{S}, |\mathcal{C}^*|)) \leq (1+\epsilon')OPT(I(\mathcal{S}, |\mathcal{C}^*|))$
- Furthermore, we also have $OPT(I(\mathcal{S}, |\mathcal{C}^*|)) \leq n$ since we can color all arcs that reach each u_i on the tree with colors from X_c (recall that $|X_c| = k \leq n$)
- We also have:

$$A_{\epsilon'}(I(\mathcal{S}, |\mathcal{C}^*|)) = cc_{\hat{\chi}}(\hat{T}, r) \geq n + \left|\mathcal{C}_e(\hat{T})\right|$$

- Let $\mathcal{C}^*$ be a minimum set cover of $\mathcal{S}$
- Let $(\hat{T}, \hat{\chi})$ be the solution returned by $A_{\epsilon'}$ on input $I(\mathcal{S}, |\mathcal{C}^*|)$.
- Recall that $\mathcal{C}_c(\hat{T}) \leq |\mathcal{C}^*|$. We now bound $\mathcal{C}_e(\hat{T})$
- We know that $A_{\epsilon'}(I(\mathcal{S}, |\mathcal{C}^*|)) \leq (1+\epsilon')OPT(I(\mathcal{S}, |\mathcal{C}^*|))$
- Furthermore, we also have $OPT(I(\mathcal{S}, |\mathcal{C}^*|)) \leq n$ since we can color all arcs that reach each u_i on the tree with colors from X_c (recall that $|X_c| = k \leq n$)
- We also have:

$$A_{\epsilon'}(I(\mathcal{S}, |\mathcal{C}^*|)) = cc_{\hat{\chi}}(\hat{T}, r) \geq n + \left|\mathcal{C}_e(\hat{T})\right|$$

- Let $\mathcal{C}^*$ be a minimum set cover of $\mathcal{S}$
- Let $(\hat{T}, \hat{\chi})$ be the solution returned by $A_{\epsilon'}$ on input $I(\mathcal{S}, |\mathcal{C}^*|)$.
- Recall that $\mathcal{C}_c(\hat{T}) \leq |\mathcal{C}^*|$. We now bound $\mathcal{C}_e(\hat{T})$
- We know that $A_{\epsilon'}(I(\mathcal{S}, |\mathcal{C}^*|)) \leq (1+\epsilon')OPT(I(\mathcal{S}, |\mathcal{C}^*|))$
- Furthermore, we also have $OPT(I(\mathcal{S}, |\mathcal{C}^*|)) \leq n$ since we can color all arcs that reach each u_i on the tree with colors from X_c (recall that $|X_c| = k \leq n$)
- We also have:

$$A_{\epsilon'}(I(\mathcal{S}, |\mathcal{C}^*|)) = cc_{\hat{\chi}}(\hat{T}, r) \geq n + \left|\mathcal{C}_e(\hat{T})\right|$$

- Let $\mathcal{C}^*$ be a minimum set cover of $\mathcal{S}$
- Let $(\hat{T}, \hat{\chi})$ be the solution returned by $A_{\epsilon'}$ on input $I(\mathcal{S}, |\mathcal{C}^*|)$.
- Recall that $\mathcal{C}_c(\hat{T}) \leq |\mathcal{C}^*|$. We now bound $\mathcal{C}_e(\hat{T})$
- We know that $A_{\epsilon'}(I(\mathcal{S}, |\mathcal{C}^*|)) \leq (1+\epsilon')OPT(I(\mathcal{S}, |\mathcal{C}^*|))$
- Furthermore, we also have $OPT(I(\mathcal{S}, |\mathcal{C}^*|)) \leq n$ since we can color all arcs that reach each u_i on the tree with colors from X_c (recall that $|X_c| = k \leq n$)
- We also have:

$$A_{\epsilon'}(I(\mathcal{S}, |\mathcal{C}^*|)) = cc_{\hat{\chi}}(\hat{T}, r) \geq n + \left|\mathcal{C}_e(\hat{T})\right|$$

- Let $\mathcal{C}^*$ be a minimum set cover of $\mathcal{S}$
- Let $(\hat{T}, \hat{\chi})$ be the solution returned by $A_{\epsilon'}$ on input $I(\mathcal{S}, |\mathcal{C}^*|)$.
- Recall that $\mathcal{C}_c(\hat{T}) \leq |\mathcal{C}^*|$. We now bound $\mathcal{C}_e(\hat{T})$
- We know that $A_{\epsilon'}(I(\mathcal{S}, |\mathcal{C}^*|)) \leq (1+\epsilon')OPT(I(\mathcal{S}, |\mathcal{C}^*|))$
- Furthermore, we also have $OPT(I(\mathcal{S}, |\mathcal{C}^*|)) \leq n$ since we can color all arcs that reach each u_i on the tree with colors from X_c (recall that $|X_c| = k \leq n$)
- We also have:

$$A_{\epsilon'}(I(\mathcal{S}, |\mathcal{C}^*|)) = cc_{\hat{\chi}}(\hat{T}, r) \geq n + \left|\mathcal{C}_e(\hat{T})\right|$$

- Let $\mathcal{C}^*$ be a minimum set cover of $\mathcal{S}$
- Let $(\hat{T}, \hat{\chi})$ be the solution returned by $A_{\epsilon'}$ on input $I(\mathcal{S}, |\mathcal{C}^*|)$.
- Recall that $\mathcal{C}_c(\hat{T}) \leq |\mathcal{C}^*|$. We now bound $\mathcal{C}_e(\hat{T})$
- We know that $A_{\epsilon'}(I(\mathcal{S}, |\mathcal{C}^*|)) \leq (1+\epsilon')OPT(I(\mathcal{S}, |\mathcal{C}^*|))$
- Furthermore, we also have $OPT(I(\mathcal{S}, |\mathcal{C}^*|)) \leq n$ since we can color all arcs that reach each u_i on the tree with colors from X_c (recall that $|X_c| = k \leq n$)
- We also have:

$$A_{\epsilon'}(I(\mathcal{S}, |\mathcal{C}^*|)) = cc_{\hat{\chi}}(\hat{T}, r) \geq n + \left|\mathcal{C}_e(\hat{T})\right|$$

- Combining all inequalities, we get $n + \left|\mathcal{C}_e(\hat{T})\right| \leq n + n\epsilon'$, implying that:

$$\left|\mathcal{C}_e(\hat{T})\right| \leq n\epsilon' \leq 3\left|\mathcal{C}^*\right|\epsilon' = \epsilon\left|\mathcal{C}^*\right|$$

where the second inequality follows from the fact that every set has at most 3 elements.

- By combining the bounds, we get:

$$|\mathcal{C}| \leq \left|\mathcal{C}(\hat{T})\right| = \left|\mathcal{C}_c(\hat{T})\right| + \left|\mathcal{C}_e(\hat{T})\right| \leq |\mathcal{C}^*| + \epsilon|\mathcal{C}^*| = (1+\epsilon)|\mathcal{C}^*|$$

□

- Combining all inequalities, we get $n + \left|\mathcal{C}_e(\hat{T})\right| \leq n + n\epsilon'$, implying that:

$$\left|\mathcal{C}_e(\hat{T})\right| \leq n\epsilon' \leq 3\left|\mathcal{C}^*\right|\epsilon' = \epsilon\left|\mathcal{C}^*\right|$$

  where the second inequality follows from the fact that every set has at most 3 elements.

- By combining the bounds, we get:

$$|\mathcal{C}| \leq \left|\mathcal{C}(\hat{T})\right| = \left|\mathcal{C}_c(\hat{T})\right| + \left|\mathcal{C}_e(\hat{T})\right| \leq |\mathcal{C}^*| + \epsilon|\mathcal{C}^*| = (1+\epsilon)|\mathcal{C}^*|$$

□

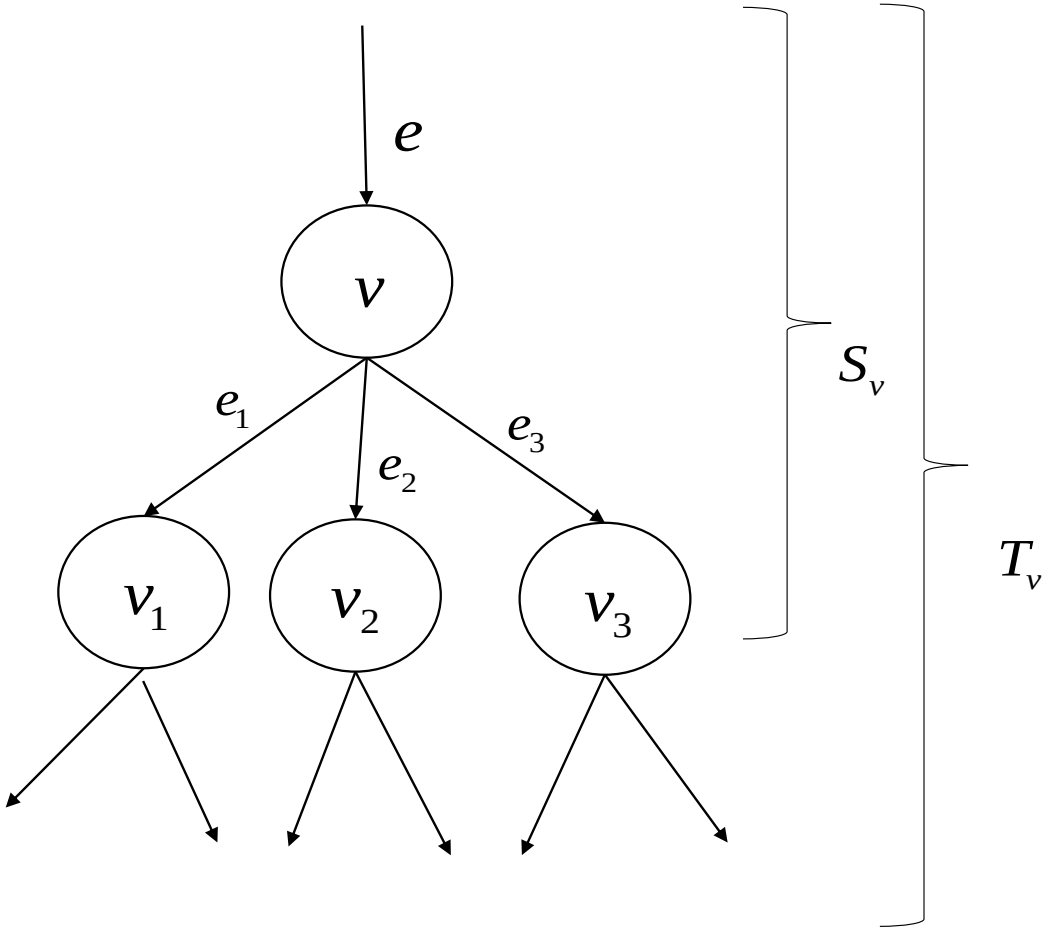
- Combining all inequalities, we get $n + \left|\mathcal{C}_e(\hat{T})\right| \leq n + n\epsilon'$, implying that:

$$\left|\mathcal{C}_e(\hat{T})\right| \leq n\epsilon' \leq 3\left|\mathcal{C}^*\right|\epsilon' = \epsilon\left|\mathcal{C}^*\right|$$

  where the second inequality follows from the fact that every set has at most 3 elements.
- By combining the bounds, we get:

$$|\mathcal{C}| \leq \left|\mathcal{C}(\hat{T})\right| = \left|\mathcal{C}_c(\hat{T})\right| + \left|\mathcal{C}_e(\hat{T})\right| \leq |\mathcal{C}^*| + \epsilon\left|\mathcal{C}^*\right| = (1+\epsilon)\left|\mathcal{C}^*\right|$$

Polynomial-time Solvable Cases



- Every traversal within T_v is either within S_v or within T_{v_i} for some $i \in [k]$. Therefore,

$$cc_\chi(\mathcal{P}, T_v) = cc_\chi(\mathcal{P}, S_v) + \sum_{i=1}^{k} cc_\chi(\mathcal{P}, T_{v_i})$$

- Let $OPT_{cc}(\mathcal{P}, v, x)$ be the minimum changeover cost within T_v, among all colorings χ such that $\chi(in_T(v)) = x$

$$\alpha_{cc}(\chi_v) = cc_{\chi_v}(\mathcal{P}, S_v) + \sum_{i=1}^{k} OPT_{cc}(\mathcal{P}, v_i, \chi_v(e_i))$$

$$OPT_{cc}(\mathcal{P}, v, x) = \min\{\alpha_{cc}(\chi_v) : \chi_v \in \mathcal{F}_{S_v}, \chi_v(e) = x\}$$

- In particular, for $v = r$ we obtain the optimum as

$$cc^*(\mathcal{P}) = \min\{\alpha_{cc}(\chi_r) : \chi_r \in \mathcal{F}_{S_r}\}$$

- Every traversal within T_v is either within S_v or within T_{v_i} for some $i \in [k]$. Therefore,

$$cc_\chi(\mathcal{P}, T_v) = cc_\chi(\mathcal{P}, S_v) + \sum_{i=1}^{k} cc_\chi(\mathcal{P}, T_{v_i})$$

- Let $OPT_{cc}(\mathcal{P}, v, x)$ be the minimum changeover cost within T_v, among all colorings χ such that $\chi(in_T(v)) = x$

$$\alpha_{cc}(\chi_v) = cc_{\chi_v}(\mathcal{P}, S_v) + \sum_{i=1}^{k} OPT_{cc}(\mathcal{P}, v_i, \chi_v(e_i))$$

$$OPT_{cc}(\mathcal{P}, v, x) = \min\{\alpha_{cc}(\chi_v) : \chi_v \in \mathcal{F}_{S_v}, \chi_v(e) = x\}$$

- In particular, for $v = r$ we obtain the optimum as

$$cc^*(\mathcal{P}) = \min\{\alpha_{cc}(\chi_r) : \chi_r \in \mathcal{F}_{S_r}\}$$

- Every traversal within T_v is either within S_v or within T_{v_i} for some $i \in [k]$. Therefore,

$$cc_\chi(\mathcal{P}, T_v) = cc_\chi(\mathcal{P}, S_v) + \sum_{i=1}^{k} cc_\chi(\mathcal{P}, T_{v_i})$$

- Let $OPT_{cc}(\mathcal{P}, v, x)$ be the minimum changeover cost within T_v, among all colorings χ such that $\chi(in_T(v)) = x$

$$\alpha_{cc}(\chi_v) = cc_{\chi_v}(\mathcal{P}, S_v) + \sum_{i=1}^{k} OPT_{cc}(\mathcal{P}, v_i, \chi_v(e_i))$$

$$OPT_{cc}(\mathcal{P}, v, x) = \min\{\alpha_{cc}(\chi_v) : \chi_v \in \mathcal{F}_{S_v}, \chi_v(e) = x\}$$

- In particular, for $v = r$ we obtain the optimum as

$$cc^*(\mathcal{P}) = \min\{\alpha_{cc}(\chi_r) : \chi_r \in \mathcal{F}_{S_r}\}$$

Theorem

MINCCEC *and* MINRCEC *are solvable in polynomial time when* G *is a tree and* $|X|^{\Delta(G)}$ *is polynomial in the input size.*

Corollary

MINCCEC *and* MINRCEC *are solvable in polynomial time whenever*

- G *is a bounded degree tree, or*
- *the number* $|X|$ *of colors is constant,* G *is a tree, and* $\Delta(G)$ *is poly-logarithmic in the size of the input.*

Theorem

MINCCEC *and* MINRCEC *are solvable in polynomial time when* G *is a tree and* $|X|^{\Delta(G)}$ *is polynomial in the input size.*

Corollary

MINCCEC *and* MINRCEC *are solvable in polynomial time whenever*

- G *is a bounded degree tree, or*
- *the number* $|X|$ *of colors is constant,* G *is a tree, and* $\Delta(G)$ *is poly-logarithmic in the size of the input.*

Theorem

MINCCEC *and* MINRCEC *are solvable in polynomial time when* G *is a tree, and a particular vertex* r *is an endpoint of every path* $P \in \mathcal{P}$*.*

Proof (Sketch):

- Since all paths have an endpoint at r, all traversals within S_v contain the edge e.
- Therefore, for $\chi_v(e) = x$,

$$cc_{\chi_v}(\mathcal{P}, S_v) = \sum_{i=1}^{k} tc(\chi_v(e), \chi_v(e_i)) = \sum_{i=1}^{k} tc(x, \chi_v(e_i))$$

- ...

Theorem

MINCCEC *and* MINRCEC *are solvable in polynomial time when* G *is a tree, and a particular vertex* r *is an endpoint of every path* $P \in \mathcal{P}$.

Proof (Sketch):

- Since all paths have an endpoint at r, all traversals within S_v contain the edge e.
- Therefore, for $\chi_v(e) = x$,

$$cc_{\chi_v}(\mathcal{P}, S_v) = \sum_{i=1}^{k} tc(\chi_v(e), \chi_v(e_i)) = \sum_{i=1}^{k} tc(x, \chi_v(e_i))$$

- ...

Theorem

MINCCEC *and* MINRCEC *are solvable in polynomial time when* G *is a tree, and a particular vertex* r *is an endpoint of every path* $P \in \mathcal{P}$.

Proof (Sketch):

- Since all paths have an endpoint at r, all traversals within S_v contain the edge e.
- Therefore, for $\chi_v(e) = x$,

$$cc_{\chi_v}(\mathcal{P}, S_v) = \sum_{i=1}^{k} tc(\chi_v(e), \chi_v(e_i)) = \sum_{i=1}^{k} tc(x, \chi_v(e_i))$$

- ...

Corollary

MINCCAEC *and* MINRCPTEC *are solvable in polynomial time for trees.*

Corollary

MINCCAEC *and* MINRCPTEC *are solvable in polynomial time for graphs* G *where* $|E(G)| - |V(G)|$ *is bounded by some constant.*

Theorem

MINCCAEC *problem is solvable in polynomial time whenever*

a) *the degree of every cut vertex of* G *is bounded by some constant* c_1*, and*

b) *for every block* B *of* G*,* $|E(B) - |V(B)||$ *is bounded by some constant* c_2*.*

Corollary

MINCCAEC *and* MINRCPTEC *are solvable in polynomial time for trees.*

Corollary

MINCCAEC *and* MINRCPTEC *are solvable in polynomial time for graphs* G *where* $|E(G)| - |V(G)|$ *is bounded by some constant.*

Theorem

MINCCAEC *problem is solvable in polynomial time whenever*

a) *the degree of every cut vertex of* G *is bounded by some constant* c_1*, and*

b) *for every block* B *of* G*,* $|E(B) - |V(B)||$ *is bounded by some constant* c_2*.*

Corollary

MINCCAEC *and* MINRCPTEC *are solvable in polynomial time for trees.*

Corollary

MINCCAEC *and* MINRCPTEC *are solvable in polynomial time for graphs* G *where* $|E(G)| - |V(G)|$ *is bounded by some constant.*

Theorem

MINCCAEC *problem is solvable in polynomial time whenever*

a) *the degree of every cut vertex of* G *is bounded by some constant* c_1*, and*

b) *for every block* B *of* G*,* $|E(B) - |V(B)||$ *is bounded by some constant* c_2*.*

- MinCCEC and MinRCEC when a special vertex is an endpoint of every path: Investigating the case with special graph classes such as cactus graphs, bounded treewidth graphs etc.
- Inapproximability of MinCCAEC and MinRCPTEC in undirected graphs
- Parameterized complexity of MinCCEC and MinRCEC (when the maximum number of paths that can use a particular edge is bounded etc.)

- MinCCEC and MinRCEC when a special vertex is an endpoint of every path: Investigating the case with special graph classes such as cactus graphs, bounded treewidth graphs etc.
- Inapproximability of MinCCAEC and MinRCPTEC in undirected graphs
- Parameterized complexity of MinCCEC and MinRCEC (when the maximum number of paths that can use a particular edge is bounded etc.)

- MinCCEC and MinRCEC when a special vertex is an endpoint of every path: Investigating the case with special graph classes such as cactus graphs, bounded treewidth graphs etc.
- Inapproximability of MinCCAEC and MinRCPTEC in undirected graphs
- Parameterized complexity of MinCCEC and MinRCEC (when the maximum number of paths that can use a particular edge is bounded etc.)