Approximation algorithms for minimum-edge-dilation $k$-center problems

J. Könemann
jochen@cmu.edu

GSIA
Carnegie Mellon University

Joint with Y. Li, O. Parekh and A. Sinha
Roadmap...

1. Problem motivation and definition
2. A combinatorial lower bound
3. Finding a solution for the lowerbound
4. Analysis
5. Extensions and open questions
Motivation: Internet routing

- Think of internet as undirected graph $G = (V, E)$
Motivation: Internet routing

- Think of internet as undirected graph $G = (V, E)$
- Each node $v$ stores routing table: $\{(u, \text{nexthop}_u^v)\}_{u \in V}$
Motivation: Internet routing

- Think of internet as undirected graph $G = (V, E)$
- Each node $v$ stores routing table: $\{(u, \text{nexthop}^v_u)\}_{u \in V}$
- At node $v$, a packet with destination $u$ is forwarded to neighbor $\text{nexthop}^v_u$
Internet routing: An example

Send a packet from node 1 to 7:

(7, 3)
Internet routing: An example

Send a packet from node 1 to 7:

(7, 8)
Internet routing: An example

Send a packet from node 1 to 7:
Internet routing: An example

Send a packet from node 1 to 7:

Problem: Need $O(n)$ space at each node to achieve shortest path routing!
A way out...

Some routing protocols allow subdividing the network...
A way out...

Some routing protocols allow subdividing the network...

Nodes know only their cell and use backbone for intracell communication (e.g. $2 \rightarrow 7$).
**Problem definition**

Have smaller routing tables but routing is not along shortest paths anymore!

**Goal:** Install backbone s.t. that the maximum dilation of any shortest path is as small as possible.
Problem definition

**Definition:** Minimum edge-dilation \( k \)-center (MEDKC)

**Given:** undirected graph \( G = (V, E) \), metric \( l \) on edges, parameter \( k \)

**Find:** \( \Pi \subseteq V, |\Pi| \leq k \) and assignment \( \pi : V \rightarrow \Pi \)

Minimize

\[
\max_{u,v\in V} \frac{d_\pi(u,v)}{d_l(u,v)}
\]
Problem definition

**Definition:** Minimum edge-dilation $k$-center (MEDKC)

**Given:** undirected graph $G = (V, E)$, metric $l$ on edges, parameter $k$

**Find:** $\Pi \subseteq V, |\Pi| \leq k$ and assignment $\pi : V \rightarrow \Pi$

Minimize

$$\max_{u,v \in V} \frac{d_\pi(u,v)}{d_l(u,v)}$$

$$d_l(u, \pi_u) + d_l(\pi_u, \pi_v) + d_l(\pi_v, v)$$
Observe...

$$\max_{u,v \in V} \frac{d_\pi(u,v)}{d_1(u,v)}$$

Two nodes in same cell talk via cell center!
Observe...

\[ \max_{u,v \in V} \frac{d_{\pi}(u,v)}{d_I(u,v)} \]

Two nodes in same cell talk via cell center!

Turns out: Can use **MEDKC** to approximate original routing problem!
Observe...

$max_{u,v \in V} \frac{d_{\pi}(u,v)}{d_{l}(u,v)}$

Two nodes in same cell talk via cell center!

Turns out: Can use **MEDKC** to approximate original routing problem!

Use **MEDKC-B** to refer to problem where close nodes communicate directly and routing tables have size at most $B$. 
Our result

**Theorem 1** There is a polytime algorithm for MEDKC that computes a solution $\Pi$ with stretch at most

$$4\text{opt} + 1$$

where $\text{opt}$ is the stretch of an optimum solution $\Pi^*$. It is NP-hard to compute a $5/4 - \epsilon$-approximation for any $\epsilon > 0$. 
Previous and related work

- \([k\text{-center}]\) Our algorithm use techniques developed in [Dyer, Frieze ’85], [Hochbaum, Shmoys ’85] and [Plesnik ’80]
Previous and related work

- **[\(k\)-center]** Our algorithm use techniques developed in [Dyer, Frieze ’85], [Hochbaum, Shmoys ’85] and [Plesnik ’80]

- **[\(\alpha\)-Spanners]** Given an undirected graph \(G\), find a subgraph \(G'\) of minimum weight/size such that

\[
\max_{u,v \in V} \frac{d_{G'}(u, v)}{d_G(u, v)} \leq \alpha.
\]

See e.g. [Kortsartz, Peleg ’94],[Elkin, Peleg ’01], ...
Previous and related work

- **[k-center]** Our algorithm use techniques developed in [Dyer, Frieze ’85], [Hochbaum, Shmoys ’85] and [Plesnik ’80]
- **[α-Spanners]** Given an undirected graph $G$, find a subgraph $G'$ of minimum weight/size such that
  \[
  \max_{u,v \in V} \frac{d_{G'}(u,v)}{d_G(u,v)} \leq \alpha.
  \]
  See e.g. [Kortsartz, Peleg ’94],[Elkin, Peleg ’01], ...
- **[Compact routing schemes]** Bounded routing table size. What is the best achievable maximum stretch? [Awerbuch et al. ’89+’90], [Cowen ’01], [Eilam et al. ’98], [Peleg, Upfal ’88+’89]
A combinatorial lowerbound

- $\Pi \subseteq V$ is MEDKC solution with stretch $\alpha$
A combinatorial lowerbound

- $\Pi \subseteq V$ is MEDKC solution with stretch $\alpha$
- Must have:

$$\Pi \cap \{ w \in V : d_{l}(u, w) + d_{l}(v, w) \leq \alpha \cdot l(u, v) \} \neq \emptyset$$

for all $uv \in E$. 
A combinatorial lowerbound

• $\Pi \subseteq V$ is MEDKC solution with stretch $\alpha$
• Must have:

$$\Pi \cap \{w \in V : d_l(u, w) + d_l(v, w) \leq \alpha \cdot l(u, v)\} \neq \emptyset$$

for all $uv \in E$.

• [Min-stretch vertex cover] (MSVC-$\alpha$)
  Find min-cardinality set $C \subseteq V$ s.t.

$$S_{uv}^\alpha \cap C \neq \emptyset$$

for all $uv \in E$.
A combinatorial lowerbound

- $\Pi \subseteq V$ is MEDKC solution with stretch $\alpha$

- Must have:

$$\Pi \cap \{w \in V : d_l(u, w) + d_l(v, w) \leq \alpha \cdot l(u, v)\} \neq \emptyset$$

for all $uv \in E$.

- [Min-stretch vertex cover] (MSVC-$\alpha$)
  Find min-cardinality set $C \subseteq V$ s.t.

$$S^\alpha_{uv} \cap C \neq \emptyset$$

for all $uv \in E$

- MEDKC solution $\Pi$, $|\Pi| = k$ and max. stretch $\alpha$

$\implies$ Can find MSVC-$(2\alpha + 1)$ solution of size $k$
A combinatorial lower bound

- $\Pi \subseteq V$ is MEDKC solution with stretch $\alpha$
- Must have:
  \[
  \Pi \cap \{w \in V : d_l(u, w) + d_l(v, w) \leq \alpha \cdot l(u, v)\} \neq \emptyset
  \]
  for all $uv \in E$.
- [Min-stretch vertex cover] (MSVC-$\alpha$)
  Find min-cardinality set $C \subseteq V$ s.t.
  \[
  S_{uv}^{\alpha} \cap C \neq \emptyset
  \]
  for all $uv \in E$
- MEDKC solution $\Pi$, $|\Pi| = k$ and max. stretch $\alpha$
  $\implies$ Can find MSVC-$(2\alpha + 1)$ solution of size $k$
- Use this to find approximate MEDKC solution
Algorithm for MEDKC

Assume: Solution to MEDKC with stretch $\alpha$ exists.

Algorithm:
Algorithm for MEDKC

Assume: Solution to MEDKC with stretch $\alpha$ exists.

Algorithm: (1) $\Pi \leftarrow$ Solution to MSVC-$(2\alpha + 1)$, $|\Pi| \leq k$
Algorithm for MEDKC

Assume: Solution to MEDKC with stretch $\alpha$ exists.

Algorithm: 
1. $\Pi \leftarrow \text{Solution to MSVC-}(2\alpha + 1)$, $|\Pi| \leq k$
2. $\forall v \in V$ let $\pi_v = \min_{u \in \Pi} d_l(v, u)$
Algorithm for MEDKC

Assume: Solution to MEDKC with stretch $\alpha$ exists.

Algorithm: 
(1) $\Pi \leftarrow$ Solution to MSVC-$(2\alpha + 1)$, $|\Pi| \leq k$
(2) $\forall v \in V$ let $\pi_v = \min_{u \in \Pi} d_l(v, u)$

Why is the solution good?
Algorithm for MEDKC

Assume: Solution to MEDKC with stretch $\alpha$ exists.

Algorithm:
1. $\Pi \leftarrow$ Solution to MSVC-$(2\alpha + 1)$, $|\Pi| \leq k$
2. $\forall v \in V$ let $\pi_v = \min_{u \in \Pi} d_l(v, u)$

Why is the solution good?
Algorithm for MEDKC

Assume: Solution to MEDKC with stretch $\alpha$ exists.

Algorithm: (1) $\Pi \leftarrow$ Solution to MSVC-$(2\alpha + 1)$, $|\Pi| \leq k$
(2) $\forall v \in V$ let $\pi_v = \min_{u \in \Pi} d_l(v, u)$

Why is the solution good?

$(2\alpha + 1)d_l(u, v)$
Algorithm for MEDKC

Assume: Solution to MEDKC with stretch $\alpha$ exists.

Algorithm: $(1) \Pi \leftarrow$ Solution to MSVC-$(2\alpha + 1)$, $|\Pi| \leq k$
$(2) \forall v \in V$ let $\pi_v = \min_{u \in \Pi} d_l(v, u)$

Why is the solution good?

$(2\alpha + 1)d_i(u, v) + d_l(u, v)$
Algorithm for MEDKC

Assume: Solution to MEDKC with stretch $\alpha$ exists.

Algorithm: 
1. $\Pi \leftarrow$ Solution to MSVC-$(2\alpha + 1)$, $|\Pi| \leq k$
2. $\forall v \in V$ let $\pi_v = \min_{u \in \Pi} d_l(v, u)$

Why is the solution good?

\[ (2\alpha + 1)d_l(u, v) + d_l(u, v) + (2\alpha + 1)d_l(u, v) = (4\alpha + 3)d_l(u, v) \]
Solving MSVC-$(2\alpha + 1)$

\[ E \leftarrow E, i \leftarrow 1, \Pi \leftarrow \emptyset \]
\[ \textbf{while} \ (E \neq \emptyset) \ \textbf{do} \]

\[ B = 3 \]
Solving MSVC-$(2\alpha + 1)$

\[
\overline{E} \leftarrow E, i \leftarrow 1, \Pi \leftarrow \emptyset
\]

while $(\overline{E} \neq \emptyset)$ do

\[
(u_i, v_i) \leftarrow \text{argmin}_{e \in E} l_e
\]

\[
\Pi \leftarrow \Pi \cup \{v_i\}
\]

$B = 3$
Solving MSVC-\((2\alpha + 1)\)

\[
\overline{E} \leftarrow E, i \leftarrow 1, \Pi \leftarrow \emptyset \\
\text{while } (\overline{E} \neq \emptyset) \text{ do} \\
\quad (u_i, v_i) \leftarrow \arg\min_{e \in \overline{E}} l_e \\
\quad \Pi \leftarrow \Pi \cup \{v_i\} \\
\quad \overline{E} \leftarrow \{e \in \overline{E} : v_i \notin S_{u_i v_i}^{2\alpha + 1}\} \\
\text{od}
\]

\[B = 3\]
Solving MSVC-$(2\alpha + 1)$

\[
\overline{E} \leftarrow E, i \leftarrow 1, \Pi \leftarrow \emptyset \\
\textbf{while} (\overline{E} \neq \emptyset) \textbf{ do} \\
\quad (u_i, v_i) \leftarrow \text{argmin}_{e \in \overline{E}} l_e \\
\quad \Pi \leftarrow \Pi \cup \{v_i\} \\
\quad \overline{E} \leftarrow \{e \in \overline{E} : v_i \notin S_{u_iv_i}^{2\alpha+1}\} \\
\textbf{od}
\]
Solving MSVC\textsuperscript{-}(2\alpha + 1)

\begin{align*}
\overline{E} &\leftarrow E, i \leftarrow 1, \Pi \leftarrow \emptyset \\
\textbf{while} \ (\overline{E} \neq \emptyset) \ \textbf{do} \\
& \quad (u_i, v_i) \leftarrow \arg\min_{e \in \overline{E}} l_e \\
& \quad \Pi \leftarrow \Pi \cup \{v_i\} \\
& \quad \overline{E} \leftarrow \{ e \in \overline{E} : v_i \not\in S_{u_i v_i}^{2\alpha+1} \} \\
\textbf{od}
\end{align*}
Analysis of MSVC-\((2\alpha + 1)\)

**Assume:** Solution \(\Pi\) to MEDKC with stretch \(\alpha\) exists.

**For the sake of contradiction:** Our algorithm ends with \(> k\) nodes.
Analysis of MSVC-$(2\alpha + 1)$

Assume: Solution $\Pi$ to MEDKC with stretch $\alpha$ exists. For the sake of contradiction: Our algorithm ends with $> k$ nodes. $\implies$ must have two nodes $v_i$ and $v_j$ ($i < j$) and $v \in \Pi$ s.t.

$$v \subseteq S_{e_i}^\alpha \cap S_{e_j}^\alpha.$$
Analysis of MSVC-\((2\alpha + 1)\)

**Assume:** Solution \(\Pi\) to MEDKC with stretch \(\alpha\) exists. For the sake of contradiction: Our algorithm ends with \(> k\) nodes. \(\implies\) must have two nodes \(v_i\) and \(v_j\) \((i < j)\) and \(v \in \Pi\) s.t.

\[
v \subseteq S_{e_i}^\alpha \cap S_{e_j}^\alpha.
\]
Analysis of MSVC-$(2\alpha + 1)$

Assume: Solution $\Pi$ to MEDKC with stretch $\alpha$ exists. For the sake of contradiction: Our algorithm ends with $> k$ nodes. $\implies$ must have two nodes $v_i$ and $v_j$ ($i < j$) and $v \in \Pi$ s.t.

$$v \subseteq S^\alpha_{e_i} \cap S^\alpha_{e_j}.$$
Analysis of MSVC-$(2\alpha + 1)$

Assume: Solution $\Pi$ to MEDKC with stretch $\alpha$ exists.
For the sake of contradiction: Our algorithm ends with $> k$ nodes. $\implies$ must have two nodes $v_i$ and $v_j$ ($i < j$) and $v \in \Pi$ s.t.

$$v \subseteq S_{e_i}^\alpha \cap S_{e_j}^\alpha.$$
Analysis of MSVC-$(2\alpha + 1)$

Assume: Solution $\Pi$ to MEDKC with stretch $\alpha$ exists.
For the sake of contradiction: Our algorithm ends with $> k$

...
**Internet routing: Solving MEDKC-B**

- **Problem was:** MEDKC assumes, that each two nodes \( v \) and \( u \) communicate via center.
  
  We want: Close nodes communicate via shortest path.

- **Way out:**
  
  Identify set of terminal nodes \( T \) s.t.
  
  1. \( v;u \) \( T \)
  
  \( u \) and \( v \) communicate via center in OPT
  
  2. \( v \in V \) \( n \) \( T \)
  
  \( v \) is close to a terminal node

  Now solve MEDKC and minimize stretch between terminal nodes.

  Assign each non-terminal to closest center node.
Internet routing: Solving MEDKC-B

- **Problem was:** MEDKC assumes, that each two nodes $v$ and $u$ communicate via center.
  We want: Close nodes communicate via shortest path.

- **Way out:** Identify set of terminal nodes $T$ s.t.
Internet routing: Solving MEDKC-B

- **Problem was:** MEDKC assumes, that each two nodes $v$ and $u$ communicate via center.
  We want: Close nodes communicate via shortest path.
- **Way out:** Identify set of terminal nodes $T$ s.t.
  1. $v, u \in T \implies u$ and $v$ communicate via center in OPT
Internet routing: Solving MEDKC-B

- **Problem was**: MEDKC assumes, that each two nodes $v$ and $u$ communicate via center.
  We want: Close nodes communicate via shortest path.

- **Way out**: Identify set of terminal nodes $T$ s.t.
  1. $v, u \in T \implies u$ and $v$ communicate via center in OPT
  2. $v \in V \setminus T \implies v$ is close to a terminal node

Assign each non-terminal to closest center node.
Problem was: MEDKC assumes, that each two nodes $v$ and $u$ communicate via center. We want: Close nodes communicate via shortest path.

Way out: Identify set of terminal nodes $T$ s.t.
1. $v, u \in T \implies u$ and $v$ communicate via center in OPT
2. $v \in V \setminus T \implies v$ is close to a terminal node

Now solve MEDKC and minimize stretch between terminal nodes.
Internet routing: Solving MEDKC-B

- **Problem was:** MEDKC assumes, that each two nodes $v$ and $u$ communicate via center.
  We want: Close nodes communicate via shortest path.
- **Way out:** Identify set of terminal nodes $T$ s.t.
  1. $v, u \in T \implies u$ and $v$ communicate via center in OPT
  2. $v \in V \setminus T \implies v$ is close to a terminal node
- Now solve MEDKC and minimize stretch between terminal nodes.
- Assign each non-terminal to closest center node.
Internet routing: Solving MEDKC-B

- **Problem was:** MEDKC assumes, that each two nodes \( v \) and \( u \) communicate via center.
  We want: Close nodes communicate via shortest path.

- **Way out:** Identify set of terminal nodes \( T \) s.t.
  1. \( v, u \in T \implies u \) and \( v \) communicate via center in OPT
  2. \( v \in V \setminus T \implies v \) is close to a terminal node

- Now solve MEDKC and minimize stretch between terminal nodes.

- Assign each non-terminal to closest center node.

  This yields a constant factor approximation for MEDKC-B
Conclusion and open questions

Also in the paper: Use facility location techniques to solve the capacitated version.

Further directions:

- **Improve constants!** Maybe using LP techniques?
- General routing with bounded space (i.e. no backbone).
Conclusion and open questions

Also in the paper: Use facility location techniques to solve the capacitated version.

Further directions:

- Improve constants! Maybe using LP techniques?
- General routing with bounded space (i.e. no backbone).

Thanks!