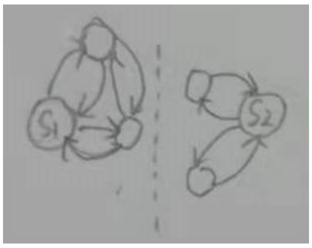
Part 10 Multiway Cut Problem

(**Definition 1**) (**Multiway Cut Problem**, MCP) Given an undirected graph G=(V, E), cost $c_e \geq 0$ associated with each edge $e \in E$ and k **distinguished vertices** $\{s_1, s_2, \cdots, s_k\}$. Our goal is to remove a minimum cost set of edges F such that no pair of distinguished vertices are in the same connected component of (V, E-F).

Notes: We can solve the MCP for k=2 in **polynomial time**. In fact, we can reformulate this problem (with only 2 distinguished vertices $\{s_1, s_2\}$) to a **Minimum Cut Problem**. Concretely, let s_1 and s_2 be the source and sink vertex, respectively and construct a flow network. Note that in MCP, edges don't have the direction, but we can use two directed edges (with 2 different directions) to replace an undirected edge. In this case, MCP is equivalent to finding a **minimum cut** in the converted flow network.

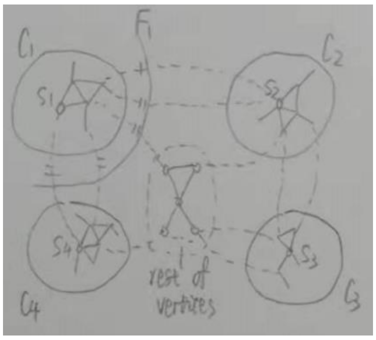


However, for $k \geq 3$, MCP is **NP-Complete**.

(**Example 1**) (**Application** in **Distributed Computing**) Let vertices V represent **'objects'** and c_e represent amount of **communication** between objects. We need to place 'objects' on k different **machine** with **special 'object'** s_i residing on the i-th machine.

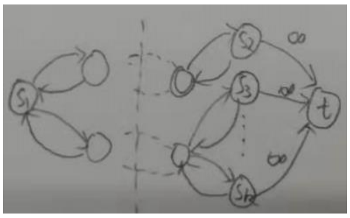
Our goal is to partition the 'objects' into k machines, so as to minimize the communication between machines.

(**Example 2**) Consider a **feasible solution** F of MCP. Let C_i be the connected component containing s_i. Let $F_i = \delta(C_i)$ be the set of edges leaving component C_i. Note that F_i is a **cut** separating s_i from all other **distinguished vertices**. We call F_i an **isolating cut** as it isolate s_i from the other distinguished vertices.



Suppose we compute a **minimum isolating cut** for each distinguish vertices and then take the union of all these isolating cuts. Clearly, this would be a **feasible solution** to the MCP.

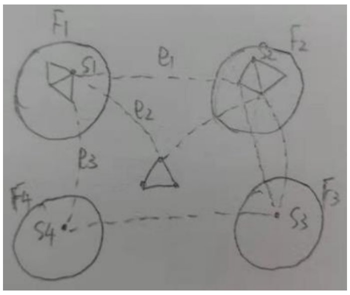
In particular, for vertex s_1, we need to separate it from all other distinguished vertices $\{s_2, s_3, \cdots, s_k\}$. We can construct a flow network with s_1 as the source vertex, while $\{s_2, s_3, \cdots, s_k\}$ are connected to an auxiliary sink vertex t with the capacities $c(s_i, t) = \infty$ ($i \geq 2$). To find F_i is equivalent to finding the **minimum *s-t* cut** of the coverted flow network.



(**Algorithm 1**) (**Approximation Algorithm of MCP**) Compute an **isolating cut** with minimum cost for each **distinguished vertex** and take the **union** of all these isolating cuts.

(**Theorem 1**) The **approximation ratio** of **Algorithm 1** is 2.

Proof of **Theorem 1**. Let F^* be the **optimal solution**. Let F_i^* be the **isolating cut** in the optimal solution for distinguished vertex s_i. In the following example, we have $F_1^*=\{e_1, e_2, e_3\}$.



For the cost of F^*, since each edge in F_i^* contributes at most twice to F^* (e.g., e_1 in the above example), we have

$$\sum_{i=1}^{k} cost(F_i^*) \leq 2cost(F^*) \cdot (1)$$

Let F denote the solution output by **Algorithm 1**. Let F_i be the **minimum isolating cut** for s_i as in **Algorithm 1**. Since in **Algorithm 1**, we have $F = F_1 \cup F_2 \cup \cdots \cup F_k$, so we further have

$$cost(F) \leq \sum_{i=1}^{k} cost(F_i) \cdot (2)$$

In **Algorithm 1**, F_i is the **minimum isolating cut** for s_i, so we have

$$cost(F_i) \leq cost(F_i^*) \cdot (3)$$

By (1), (2), and (3), we have

$$cost(F) \leq \sum_{i=1}^{k} cost(F_i) \leq \sum_{i=1}^{k} cost(F_i^*) \leq 2cost(F^*) \cdot$$

In summary, the **approximation ratio** of **Algorithm 1** is

$$\frac{cost(F)}{cost(F^*)} \leq \frac{2cost(F^*)}{cost(F^*)} = 2 \cdot$$

(**Theorem 2**) We can further improve the analysis of **Theorem 1**, where we only need to find (k-1) isolating cuts (w.r.t. (k-1) distinguish vertices) in **Algorithm 1**. The remaining single distinguished vertex must be isolated from other distinguished vertices.

Especially, we can leave the distinguished vertex with **most expensive isolating cut** and find the isolating cuts of the rest distinguished vertices. Then, the **approximation ratio** of **Algorithm 1** can be improved from 2 to 2(1-1/*k*).

Proof of **Theorem 2**. Without loss of generality, assume that s_k is with the most expensive isolating cut. Namely, we have $cost(F_k) \geq cost(F_i)$ for $1 \leq i \leq (k-1)$. Then, we have

$$\sum_{i=1}^{k} cost(F_i) \leq k \cdot cost(F_k) \Leftrightarrow \frac{1}{k}\sum_{i=1}^{k} cost(F_i) \leq cost(F_k) \cdot$$

Let *F*' be the result given by the improved version of **Algorithm 1**. Then, we have

$$\begin{aligned} cost(F') &= \sum_{i=1}^{k-1} cost(F_i) \\ &= \sum_{i=1}^{k} cost(F_i) - cost(F_k) \\ &\leq (1-\frac{1}{k})\sum_{i=1}^{k} cost(F_i) \end{aligned} \cdot$$

In **Theorem 1**, we have $\sum_{i=1}^{k} cost(F_i) \leq 2cost(F^*)$, thus we have

$$cost(F') \leq (1-\frac{1}{k})\sum_{i=1}^{k} cost(F_i) \leq 2(1-\frac{1}{k})cost(F^*) \cdot$$

In summary, the **approximation ratio** of the revised **Algorithm 1** is

$$\frac{cost(F')}{cost(F^*)} \leq \frac{2(1-1/k)cost(F^*)}{cost(F^*)} = 2(1-\frac{1}{k}) \cdot$$

(**Definition 2**) (**Reformulation of the Multiway Cut Problem**) Let $\delta(C_i)$ be the set of edge leaving the vertex set C_i. Partition the set of vertices *V* into sets C_i, such that (1) $s_i \in C_i$ for $1 \leq i \leq k$ and (2) the cost of $F = \cup\delta(C_i)$ is minimized.

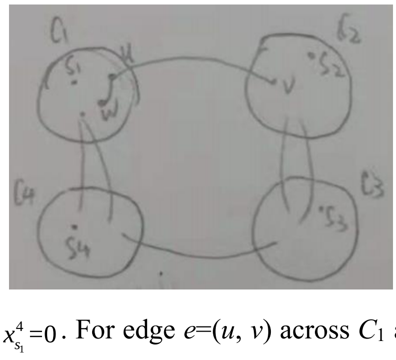
For each vertex $u \in V$, introduce *k* different variables $x_u^i \in \{0,1\}$, in which

$$x_u^i = \begin{cases} 1, & \text{if } u \in C_i \\ 0, & \text{otherwise} \end{cases} .$$

For each edge $e \in E$, introduce *k* variables $z_e^i \in \{0,1\}$, in which

$$z_e^i = \begin{cases} 1, & e \in \delta(C_i) \\ 0, & \text{otherwise} \end{cases} .$$

Consider the following example with *k*=4.



We have $x_{s_1}^1 = 1$ and $x_{s_1}^2 = x_{s_1}^3 = x_{s_1}^4 = 0$. For edge *e*=(*u*, *v*) across C_1 and C_2, we have $z_{(u,v)}^1 = z_{(u,v)}^2 = 1$, w.r.t. $x_u^1 = 1$ and $x_v^2 = 1$. For edge e=(u, w) within C_1, we have $z_{(u,w)}^1 = 0$ w.r.t. $x_u^1 = x_w^1 = 1$.

Since each vertex u can be partitioned only into one vertex set C_i, we have the constraint $\sum_{i=1}^{k} x_u^i = 1$. Since s_i can be partitioned only into set C_i, we have the constraint $x_{s_i}^i = 1$. For each edge e=(u, v), by considering both the cases that (1) end points u and v are in the same set C_i or (2) u and v are in different set C_i and C_i, we have the constraint $z_e^i \geq | x_u^i - x_v^i |$, which can be rewritten as two constraints $z_e^i \geq x_u^i - x_v^i$ and $z_e^i \geq x_v^i - x_u^i$. Note that $z_e^i = 1$ w.r.t. edge e=(u, v) across C_i and C_j <u>contributes twice</u> in $\sum_{i=1}^{k} z_e^i$. Thus, our goal is to minimize the objective function $\frac{1}{2}\sum_{e \in E} c_e \sum_{i=1}^{k} z_e^i$.

In summary, we can reformulate the MCP as the following **ILP**:

$$
\begin{aligned}
&\min \frac{1}{2}\sum_{e \in E} c_e \sum_{i=1}^{k} z_e^i \\
&\text{s.t. } \sum_{i=1}^{k} x_u^i = 1, \forall u \in V \\
&\quad x_{s_i}^i = 1, i \in \{1, \cdots, k\} \\
&\quad z_e^i \geq x_u^i - x_v^i, \forall e = (u,v) \in E \\
&\quad z_e^i \geq x_v^i - x_u^i, \forall e = (u,v) \in E \\
&\quad x_u^i \in \{0,1\}, \forall u \in V
\end{aligned}
.
$$

We can relax the integral constraint $x_u^i \in \{0,1\}$ as $x_u^i \geq 0$, because we already have the constraint that $\sum_{i=1}^{k} x_u^i = 1$, which indicates that <u>we must have $x_u^i \leq 1$, i.e., we only consider $x_u^i \geq 0$ for the original relaxed constraint $0 \leq x_u^i \leq 1$</u>. Thus, we have the following **LP-Relaxation**:

$$
\begin{aligned}
&\min \frac{1}{2}\sum_{e \in E} c_e \sum_{i=1}^{k} z_e^i \\
&\text{s.t. } \sum_{i=1}^{k} x_u^i = 1, \forall u \in V \\
&\quad x_{s_i}^i = 1, i \in \{1, \cdots, k\} \\
&\quad z_e^i \geq x_u^i - x_v^i, \forall e = (u,v) \in E \\
&\quad z_e^i \geq x_v^i - x_u^i, \forall e = (u,v) \in E \\
&\quad x_{s_i}^i = 1, i \in \{1, \cdots, k\} \\
&\quad x_u^i \geq 0, \forall u \in V
\end{aligned}
.
$$

(**Definition 3**) (**Reformulation of LP-Relaxation for MCP**) Think of $\mathbf{x}_u$ as a point in a k-dimensional space with $\mathbf{x}_u = [x_u^1, x_u^2, \cdots, x_u^k]$. Define a <u>$k$-simplex</u> as $\Delta_k = \{\mathbf{x} \in \Re^k : \sum_{i=1}^{k} x^i = 1, x^i \geq 0\}$.

For the constraint $z_e^i \geq | x_u^i - x_v^i |$ w.r.t. the edge e=(u, v), since we aim to <u>minimize the objective function</u>, we can rewrite such a constrain as $z_e^i = \left| x_u^i - x_v^i \right|$. We further have

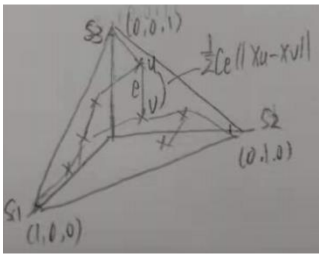
$$\sum_{i=1}^{k} z_e^i = \sum_{i=1}^{k} | x_u^i - x_v^i | = \|\mathbf{x}_u - \mathbf{x}_v\| ,$$

where $\|\mathbf{x}_u - \mathbf{x}_v\|$ denotes the l_1-distance between $\mathbf{x}_u$ and $\mathbf{x}_v$. Thus, the **LP-Relaxation** in **Definition 2** is equivalent to

$$\begin{aligned} &\min \frac{1}{2} \sum_{e=(u,v)\in E} c_e \|\mathbf{x}_u - \mathbf{x}_v\|_1 \\ &\text{s.t. } \mathbf{x}_u \in \Delta_k, \forall u \in V \\ &\quad \mathbf{x}_{s_i} = \mathbf{e}_i, i \in \{1, \cdots, k\} \end{aligned} ,$$

where $\mathbf{e}_i$ is defined as the point with 1 in the i-th coordinate and 0 elsewhere.

For k=3, we have the following **3-simplex**:

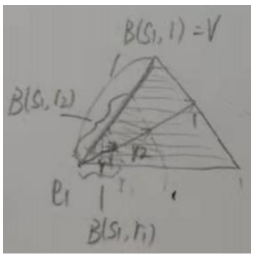


where the distinguished vertices $\{s_1, s_2, s_3\}$ are mapped to the 3 **corners** (1, 0, 0), (0, 1, 0), and (0, 0, 1) while other vertices are mapped to points on the 3-simplex.

Especially, we can **redraw** the **edges** in G on the 3-simplex, where the **cost** of each edge e=(u, v) is **scaled** by half of the l_1-distance between its two end points, i.e., $\|\mathbf{x}_u - \mathbf{x}_v\|_1 / 2$. The goal of the **LP-Relaxation** is to minimize the sum of reweighted cost of all the edges by properly mapping the vertices to the 3-simplex. Intuitively, when $c_e = c_{(u, v)}$ is costly, u and v should be mapped closed to each other on the 3-simplex.

In contrast, for the original **ILP**, we can only map each vertex to 1 of the 3 corners of the 3-simplex, i.e., (1, 0, 0), (0, 1, 0), and (0, 0, 1). The ***l_1*-distance** between two vertices u and v that are mapped to 2 different corners (e.g., (1, 0, 0) and (0, 0, 1)) is (1+1)/2=1.

(**Algorithm 2**) (**Randomized Rounding Algorithm**) Define $B(s_i, r)$ be the set of vertices such that $\|\mathbf{e}_i - \mathbf{x}_u\|/2 \le r$, i.e., with the **ball-distance** at most r. Consider the following example, where we assume r_1<r_2<1:



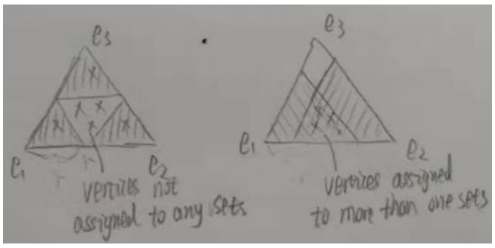
Especially, we have $B(S_i, 1)=V$, i.e., the whose vertex set, since we have $\|\mathbf{e}_i - \mathbf{e}_j\|/2 = 1$.

(1) Solve the **optimal solution** to the **LP-Relaxation**.

(2) Select $r \in (0,1)$ uniformly at random.

(3) Assign vertices within $B(s_i, r)$ to C_i.

Notes: **Algorithm 2** may **not** output a **feasible solution** to MCP, where (1) some vertices are not assigned to any set $\{C_1,\cdots,C_k\}$ (w.r.t. small r value) or (2) some vertices are assigned to more than one sets.



(**Algorithm 3**) (**Revised Randomized Rounding Algorithm**)

(1) Solve the optimal solution to the **LP-Relaxation**.

(2) Select $r \in (0,1)$ uniformly at random.

(3) Generate a **random permutation** Π of $\{1,2,\cdots,k\}$.

(4) <u>Examine the corners in the order given by the permutation Π</u>, i.e., $s_{\Pi(1)}, s_{\Pi(2)}, \cdots, s_{\Pi(k)}$.

(5) For index $\Pi_{(i)}$, assign all vertices <u>not assigned so far</u> in $B(s_{\Pi(i)}, r)$ to set $C_{\Pi(i)}$.

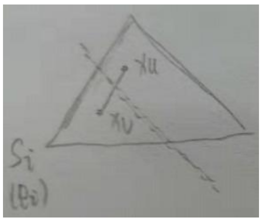
(6) At the end of the order, i.e., $s_{\Pi(k)}$, <u>assign all vertices not assigned so far to $C_{\Pi(k)}$</u>.

(**Lemma 1**) Vertex $u \in B(s_i, r)$ iff $1 - x_u^i \le r$.

Proof of **Lemma 1**. For any vertex $u \in V$, we have $\sum_{i=1}^{k} x_u^i = 1$. For the ball-distance between $\mathbf{x}_u$ and $\mathbf{e}_i$, we have

$$\begin{aligned}\frac{1}{2}\|\mathbf{x}_u - \mathbf{e}_i\| &= \frac{1}{2}\sum_{j=1}^{k} | x_u^j - e_i^j | \\ &= \frac{1}{2}[(1-x_u^i) + (\sum_{j=1}^{k} x_u^j - x_u^i)] \\ &= \frac{1}{2}[(1-x_u^i) + (1-x_u^i)] \\ &= 1 - x_u^i\end{aligned}.$$

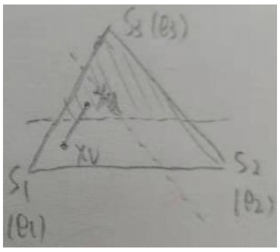
(**Definition 4**) An index i **cuts** an edge (u, v) if exactly one of end points $\mathbf{x}_u$ and $\mathbf{x}_v$ lies in $B(s_i, r)$.



(**Claim 2**) If an edge (u, v) belongs to the **multiway cut**, then some index i must **cut** (u, v).

Poor of **Claim 2**. Suppose that edge (u, v) belongs to the multiway cut, but no index i cut (u, v). Since no index i cut the edge (u, v), it must not contribute the final multiway cut in **Algorithm 3**, which contradicts with our assumption.

Notes: Note that **Claim 2** is the **necessary condition**, but **not** the **sufficient condition** for MCP. Consider the following example, where s_3 cuts the edge (u, v). However, suppose we consider s_1 before s_3, then u and v are assigned to C_1 even though s_3 cuts (u, v).



(**Definition 5**) An index i **settles** an edge (u, v) if i is the **first index** in the random permutation Π such that <u>at least one of the end point u and v belong to $B(s_i, r)$</u>.

(**Claim 3**) For each index i and an edge (u, v), consider the quantity

$$\min\{\|\mathbf{e}_i - \mathbf{x}_u\|, \|\mathbf{e}_i - \mathbf{x}_v\|\}.$$

Let l be the index that minimize the above quantity, i.e., l is the index that <u>minimizes the distance to the closer of the two end points u and v</u>. Index $i \neq l$ cannot settle edge (u, v) if l is ordered before i in the random permutation.

Proof of **Claim 3**. Suppose that $i \neq l$ settles edge (u, v) and l is ordered before i in the random permutation. Since l is the index that minimizes the above quantity and it's ordered before i, l should settle edge (u, v) before checking index i. Thus, i must not settle edge (u, v), which contradicts with our assumption.

(**Claim 4**) For any index $l \in \{1, \cdots, k\}$ and an edge (u, v), we have $|x_u^l - x_v^l| \le \|\mathbf{x}_u - \mathbf{x}_v\|/2$.

Proof of **Claim 4**. Since $\mathbf{x}_u$ and $\mathbf{x}_v$ are on the k-simplex, we have

$$\sum\nolimits_{i=1}^{k} x_u^i = \sum\nolimits_{i=1}^{k} x_v^i = 1.$$

Consider $\sum\nolimits_{i=1}^{k} x_u^i - \sum\nolimits_{i=1}^{k} x_v^i = 0$, which can be rewritten as

$$(x_u^l - x_v^l) + \sum\nolimits_{i \neq l} (x_u^i - x_v^i) = 0 \Leftrightarrow (x_u^l - x_v^l) = -\sum\nolimits_{i \neq l} (x_u^i - x_v^i),$$

Consider the absolute value for both sides:

$$|x_u^l - x_v^l| = |\sum\nolimits_{i \neq l} (x_u^i - x_v^i)|.$$

Since $x_u^j \ge 0$ and $x_v^j \ge 0$ for $j \in \{1, \cdots, k\}$, we have

$$\begin{aligned} |x_u^l - x_v^l| &= |\sum\nolimits_{i \neq l} (x_u^i - x_v^i)| \\ &\le \sum\nolimits_{i \neq l} |x_u^i - x_v^i| \end{aligned}.$$

For both sides, we add $|x_u^l - x_v^l|$ and have

$$2\,|x_u^l - x_v^l| \le \sum_{i=1}^{k} |x_u^i - x_v^i| = \|\mathbf{x}_u - \mathbf{x}_v\| \cdot$$

Namely, we have

$$|x_u^l - x_v^l| \le \frac{1}{2}\|\mathbf{x}_u - \mathbf{x}_v\| \cdot$$

(**Theorem 3**) The **approximation ratio** of **Algorithm 3** is 1.5.

Proof of **Theorem 3**. Let $OPT_{\text{MCP}}(I)$ be the **optimal cost** of **MCP** and $OPT_{\text{LP-R}}(I)$ be the **optimal value** of the **LP-Relaxation**. We have

$$OPT_{\text{MCP}}(I) \ge OPT_{\text{LP-R}}(I) = \frac{1}{2}\sum_{e=(u,v)\in E} c_e \|\mathbf{x}_u - \mathbf{x}_v\| \cdot$$

Let F denote the **multiway cut** given by **Algorithm 3**. The expected cost of F is

$$\begin{aligned} \mathrm{E}[cost(F)] &= \sum_{e=(u,v)\in E} c_e P((u,v)\in F) \\ &= \sum_{e=(u,v)\in E} c_e P(u\in C_i, v\in C_j, i\ne j) \end{aligned} \quad . (1)$$

If an index i **cut** an edge (u, v), we have (1) $u\in B(s_i,r)$ and $v\notin B(s_i,r)$, or (2) $u\notin B(s_i,r)$ and $v\in B(s_i,r)$. By **Claim 1**, for the case (1) and (2), we have

$$1-x_u^i \le r \le 1-x_v^i \text{ and } 1-x_v^i \le r \le 1-x_u^i \,.$$

For the probability that index i cut edge (u, v), we have

$$\begin{aligned} P(\text{index } i \text{ cut } (u,v)) &= P(r\in[\min\{1-x_u^i, 1-x_v^i\}, \max\{1-x_u^i, 1-x_v^i\}]) \\ &= \frac{|(1-x_u^i)-(1-x_v^i)|}{1} \\ &= |x_u^i - x_v^i| \end{aligned} \quad . (2)$$

For the probability that end point u and v are in different sets e.g., C_i and C_j ($i\ne j$), we have

$$\begin{aligned} & P(u\in C_i, v\in C_j, i\ne j) = P(\text{some index } i \text{ cuts } (u,v)) \\ & \le \sum_{i=1}^{k} (\text{index } i \text{ cuts } (u,v)) \\ & \le \sum_{i=1}^{k} |x_u^i - x_v^i| \\ & = \|\mathbf{x}_u - \mathbf{x}_v\| \end{aligned} \quad . (3)$$

By equation (2) and (3), we further have the **expected cost** of F such that

$$\begin{aligned} \mathrm{E}[cost(F)] &= \sum_{e=(u,v)\in E} c_e P(u\in C_i, v\in C_j, i\ne j) \\ &\le \sum_{e=(u,v)\in E} c_e \|\mathbf{x}_u - \mathbf{x}_v\| \end{aligned} \,.$$

Combine with (1), we have

$$OPT_{\text{MCP}}(I) \ge \frac{1}{2}\sum_{e=(u,v)\in E} c_e \|\mathbf{x}_u - \mathbf{x}_v\| \ge \frac{1}{2}\mathrm{E}[cost(F)],$$

namely we have

$$\mathrm{E}[cost(F)] \le 2OPT_{\text{MCP}}(I) \cdot$$

In this case, the **approximation ratio** of **Algorithm 3** is

$$\frac{\mathrm{E}[cost(F)]}{OPT_{\mathrm{MCP}}(I)} \le \frac{2OPT_{\mathrm{MCP}}(I)}{OPT_{\mathrm{MCP}}(I)} = 2 \cdot$$

In fact, we can further improve the analysis by considering the order that the end points of an edges are partitioned.

Let X_i denote the event that index i **settles** edge (u, v). Let Y_i denote the event that index i **cuts** edge (u, v). Edge (u, v) is in the **multiway cut result** only if there's some index i that both **settles** and **cuts** edge (u, v), i.e., $X_i \wedge Y_i$.

For any two indices $i \neq l$, suppose l is with the same definition as in **Claim 3** (l is the index that minimizes the distance to the closer of the two end points u and v, i.e., l is the index that minimize $\min\{\|\mathbf{e}_i - \mathbf{x}_u\|, \|\mathbf{e}_i - \mathbf{x}_v\|\}$). We have

$$P(i \text{ occurs before } l \text{ in } \Pi) = P(l \text{ occurs before } i \text{ in } \Pi) = 1/2 \cdot$$

By **Claim 3**, we further have

$$P(X_i \wedge Y_i \mid l \text{ occurs before } i) = 0 \cdot$$

For the probability that $X_i \wedge Y_i$, we have

$$\begin{aligned}
&P(X_i \wedge Y_i) = P(i \text{ occurs before } l) P(X_i \wedge Y_i \mid i \text{ occurs before } l) \\
&+ P(l \text{ occurs before } i) P(X_i \wedge Y_i \mid l \text{ occurs before } i) \\
&= \frac{1}{2} P(X_i \wedge Y_i \mid i \text{ occurs before } l) \\
&\le \frac{1}{2} P(Y_i)
\end{aligned} \cdot$$

Further, by equation (2), we have

$$P(X_i \wedge Y_i) \le \frac{1}{2} P(Y_i) = \frac{1}{2} | x_u^i - x_v^i | \cdot \quad (4)$$

In contrast, for index l, we have

$$P(X_l \wedge Y_l) \le P(Y_l) = | x_u^l - x_v^l | \cdot \quad (5)$$

By equation (4) and (5), we have

$$\begin{aligned}
P((u,v) \in F) &\le \sum_{i=1}^{k} P(X_i \wedge Y_i) \\
&= \sum_{i=1, i \neq l}^{k} P(X_i \wedge Y_i) + P(X_l \wedge Y_l) \\
&\le \frac{1}{2} \sum_{i=1, i \neq l}^{k} | x_u^i - x_v^i | + | x_u^l - x_v^l | \\
&= \frac{1}{2} \sum_{i=1, i \neq l}^{k} | x_u^i - x_v^i | + \frac{1}{2} | x_u^l - x_v^l | + \frac{1}{2} | x_u^l - x_v^l | \\
&= \frac{1}{2} \sum_{i=1}^{k} | x_u^i - x_v^i | + \frac{1}{2} | x_u^l - x_v^l | \\
&= \frac{1}{2} \|\mathbf{x}_u - \mathbf{x}_v\| + \frac{1}{2} | x_u^l - x_v^l |
\end{aligned} \cdot \quad (6)$$

By **Claim 4** and equation (6), we have

$$\begin{aligned} P((u,v)\in F) &\le \frac{1}{2}\|\mathbf{x}_u-\mathbf{x}_v\|+\frac{1}{2}|x_u^l-x_v^l| \\ &\le \frac{1}{2}\|\mathbf{x}_u-\mathbf{x}_v\|+\frac{1}{4}\|\mathbf{x}_u-\mathbf{x}_v\| \\ &= \frac{3}{4}\|\mathbf{x}_u-\mathbf{x}_v\| \end{aligned} . \quad (8)$$

Thus, the **expected cost** of F is

$$\begin{aligned} \mathrm{E}[cost(F)] &= \sum_{e=(u,v)\in E} c_e P((u,v)\in F) \\ &\le \frac{3}{4}\sum_{e=(u,v)\in E} c_e \|\mathbf{x}_u-\mathbf{x}_v\| \\ &\le \frac{3}{2}\cdot\frac{1}{2}\sum_{e=(u,v)\in E} c_e \|\mathbf{x}_u-\mathbf{x}_v\| \\ &\le \frac{3}{2} OPT_{\mathrm{MCP}}(I) \end{aligned} .$$

In summary, the **approximation ratio** of **Algorithm 3** is

$$\frac{\mathrm{E}[cost(F)]}{OPT_{\mathrm{MCP}}(I)} \le \frac{3}{2}\cdot\frac{OPT_{\mathrm{MCP}}(I)}{OPT_{\mathrm{MCP}}(I)} = 1.5 \cdot$$