アルゴリズム 62－3
(1998. 5. 20)

半局所改善に基づく3-Set Cover近似アルゴリズム

山下 慶子　浅野 孝夫
中央大学大学院理工学研究科情報工学専攻
E-mail: yyamashi@asano-lab.ise.chuo-u.ac.jp　asano@ise.chuo-u.ac.jp

3-Set Coverに対する最善の近似アルゴリズムはDuh-Fürerによる半局所改善に基づく$\frac{4}{3}$近似アルゴリズムであるが、半局所改善の可能性の判定および適用に関しては未解決の部分が多い。本論文では、半局所改善の可能性の効率的判定法および実行法を示し、3-Set Coverに対する$\frac{4}{3}$近似アルゴリズムの計算時間を明らかにする。

An Approximation Algorithm for 3-Set Cover Based on Semi-Local Improvements

Yoshiko Yamashita　Takao Asano
Department of Information and System Engineering, Chuo University
E-mail: yyamashi@asano-lab.ise.chuo-u.ac.jp　asano@ise.chuo-u.ac.jp

The best known approximation algorithm for 3-Set Cover is the $\frac{4}{3}$-approximation algorithm proposed by Duh and Fürer based on the semi-local improvements. There are, however, several unsolved issues about determining the possibility of semi-local improvements and, if so, about performing them. In this paper, we show an algorithm for determining whether we can do semi-local improvements and performing these improvements and clarify the time complexity of their $\frac{4}{3}$-approximation algorithm based on the semi-local improvements.

1 Introduction

We are given a collection $\mathcal{C}$ of sets with $V = \cup_{S\in\mathcal{C}} S$. A sub-collection $\mathcal{A}$ of $\mathcal{C}$ is called a *set cover*, if $\cup_{S\in\mathcal{A}} S = V$. The set cover problem (*Set Cover*, for short) is the problem of finding a minimum set cover $\mathcal{A}$ of $\mathcal{C}$, i.e., $|\mathcal{A}| \leq |\mathcal{A}'|$ for any set cover $\mathcal{A}'$ of $\mathcal{C}$. *k-Set Cover* is the Set Cover in which each set in $\mathcal{C}$ is of size at most k. For simplicity, a set S with $|S| = i$ is called an *i-set* and we assume that $\mathcal{C}$ is closed under subsets, that is, if S is a set in $\mathcal{C}$ then all subsets of S are also in $\mathcal{C}$. Thus, we always have a disjoint (minimum) set cover.

For $k = 2$, k-Set Cover is (almost) the same as the maximum matching problem and can be solved in polynomial time [3]. For general $k \geq 3$, k-Set Cover is NP-hard and several approximation algorithms have been proposed. A greedy algorithm choosing a set covering as many uncovered elements as possible each time has a performance ratio of $\mathcal{H}_k = 1 + \frac{1}{2} + \cdots + \frac{1}{k}$ for k-Set Cover [6, 7]. For 3-Set Cover, this performance ratio $\frac{11}{6}$ has been improved to $\frac{10}{6}$ [2], $\frac{11}{7}$ [4] and $\frac{7}{5}$ [5]. Recently, Duh and Fürer [1] obtained a $\frac{4}{3}$-approximation algorithm for 3-Set Cover based on semi-local improvements defined as follows: For a set cover $\mathcal{A}$, a *semi-local* (s,t) *improvement* is a step obtaining a better set cover from the current set cover $\mathcal{A}$ by (1) deleting up to t 3-sets, (2) inserting up to s 3-sets, and (3) choosing optimally 2-sets and 1-sets for uncovered elements. They have shown that their semi-local $(2,1)$ optimization algorithm, the algorithm which starts with any solution and continues semi-local $(2,1)$ improvements as long as possible, is a $\frac{4}{3}$-approximation algorithm. They, however, gave no analysis about the time complexity and several ambiguous parts remained to be clarified. For example, the following case may occur. Suppose that the deletion of a 3-set $T \in \mathcal{A}$ does not lead to a semi-local $(0,1)$ improvement, but the deletion of a 3-set $T' \in \mathcal{A}$ leads to a semi-local $(0,1)$ improvement. Let $\mathcal{A}'$ be the new set cover obtained from $\mathcal{A}$ by a semi-local $(0,1)$ improvement using 3-set $T' \in \mathcal{A}$. Then it becomes possible that the deletion of the 3-set $T \in \mathcal{A}'$ will lead to a semi-local $(0,1)$ improvement. Thus, we have to check each time whether the deletion of a 3-set $T \in \mathcal{A}$ leads to a semi-local $(0,1)$ improvement. This is the simplest case and similar cases may happen when performing semi-local $(2,1)$ improvements. In a naive way, we have to check each time, for each possible sets-pair $(\emptyset, \{T\})$, $(\{S\}, \emptyset)$, $(\{S\}, \{T\})$, $(\{S', S''\}, \emptyset)$ or $(\{S', S''\}, \{T\})$ with 3-sets $S, S', S'' \in \mathcal{C} - \mathcal{A}$ and $T \in \mathcal{A}$, whether it is a sets-pair leading to a semi-local $(2,1)$ improvement. After $O(|V|)$ semi-local improvements, we reach a set cover with no semi-local $(2,1)$ improvement. Since $|\mathcal{A}| = O(|V|)$, a naive implementation of Duh and Fürer's algorithm requires $O(|\mathcal{C}|^2|V|^2)$ semi-local improvement tests, until we reach a set cover with no semi-local $(2,1)$ improvement. Since a naive semi-local improvement test requires a maximum matching algorithm with $O(|V|^{2.5})$ time, the entire algorithm requires $O(|\mathcal{C}|^2|V|^{4.5})$ time.

In this paper, we will give an efficient implementation of their algorithm. The time complexity of our implementation is $O(|\mathcal{C}|^2|V|^2)$.

2 Notation

We use the following notational symbols throughout the paper. $\mathcal{C}$ denotes a given instance of 3-Set Cover and $V = \cup_{S \in \mathcal{C}} S$ ($n = |V|$ and $m = |\mathcal{C}|$). $\mathcal{C}_i$ denotes the set of all i-sets in $\mathcal{C}$ and $m_i = |\mathcal{C}_i|$ ($i = 1, 2, 3$). For simplicity, we write $(x_1, ..., x_i)$ for i-set $\{x_1, ..., x_i\}$, for example, (x, y) for 2-set $\{x, y\}$. Similarly, $\mathcal{A}$ denotes a disjoint set cover of $\mathcal{C}$ and $\mathcal{A}_i$ is the set of all i-sets in $\mathcal{A}$ ($a = |\mathcal{A}|$ and $a_i = |\mathcal{A}_i|$). Since we have assumed that $\mathcal{C}$ is closed under subsets, we always have a disjoint set cover $\mathcal{A}$ and $a = a_1 + a_2 + a_3$ and $n = a_1 + 2a_2 + 3a_3$ hold. The *value* of $\mathcal{A}$ is denoted by $val(\mathcal{A}) = (a, a_1)$. We would like to find a disjoint set cover with lexicographically-minimum value. Let $V(\mathcal{A}_i) = \cup_{A \in \mathcal{A}_i} A$ and $n(\mathcal{A}_i) = |V(\mathcal{A}_i)|$ ($n(\mathcal{A}_i) = ia_i$). Corresponding to a disjoint set cover $\mathcal{A}$, we denote by $G(\mathcal{A})$ the (undirected) graph with vertex set V and edge set

$$\{(v_1, v_2) \in \mathcal{C}_2 \mid v_1, v_2 \text{ are not both in } V(\mathcal{A}_3)\}.$$

This is a subgraph of the graph $H(\mathcal{C})$ with vertex set V and edge set $\{(v_1, v_2) \mid (v_1, v_2) \in \mathcal{C}_2\}$. In fact, $G(\mathcal{A})$ can be obtained from $H(\mathcal{C})$ by deleting all edges joining two vertices in $V(\mathcal{A}_3)$ and $\mathcal{A}_2$ is a matching of $G(\mathcal{A})$ (Fig. 1).

Analogously, for a disjoint set cover $\mathcal{A}^{(j)}$ of $\mathcal{C}$, we can define, $\mathcal{A}_i^{(j)}$, $a^{(j)}$, $a_i^{(j)}$, $val(\mathcal{A}^{(j)})$, $V(\mathcal{A}_i^{(j)})$, $n(\mathcal{A}_i^{(j)})$ and $G(\mathcal{A}^{(j)})$. For example, $\mathcal{A}_i^{(j)}$ is the set of all i-sets in $\mathcal{A}^{(j)}$.

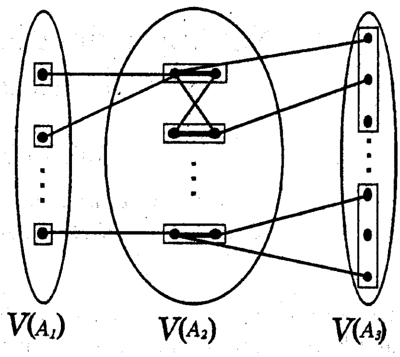

Fig 1: Graph $G(\mathcal{A})$.

3 Outline of Our Algorithm

In this section we will give an outline of our algorithm for 3-Set Cover. For a set cover $\mathcal{A}$, a *semi-local* $[s,t]$ *improvement* is a step obtaining a set cover with better value from the current set cover $\mathcal{A}$ by (1) deleting (exactly) t 3-sets, (2) inserting (exactly) s 3-sets, and (3) choosing optimally 2-sets and 1-sets for uncovered elements. Thus, each semi-local $(2,1)$ improvement is a semi-local $[0,1]$, $[1,0]$, $[1,1]$, $[2,0]$ or $[2,1]$ improvement.

Our algorithm consists of four steps below. There we use only semi-local $[0,1]$, $[1,0]$ and $[2,0]$ improvements.

Step 0. Find a set cover $\mathcal{A}^{(1)}$ by a greedy algorithm.

Step 1. Obtain a set cover $\mathcal{A}^{(2)}$ from $\mathcal{A}^{(1)}$ by using semi-local $[0,1]$ improvements as long as possible.

Step 2. Obtain a set cover $\mathcal{A}^{(3)}$ from $\mathcal{A}^{(2)}$ by using semi-local $[1,0]$ improvements as long as possible.

Step 3. Obtain a set cover $\mathcal{A}^{(4)}$ from $\mathcal{A}^{(3)}$ by using semi-local $[2,0]$ improvements as long as possible.

Then the set cover $\mathcal{A}^{(4)}$ can be shown to satisfy $|\mathcal{A}^{(4)}| \le \frac{4}{3}|\mathcal{B}^*|$ for any optimal set cover $\mathcal{B}^*$, i.e., $a_1^{(4)} + a_2^{(4)} + a_3^{(4)} \le \frac{4}{3}(b_1^* + b_2^* + b_3^*)$ ($b_i^* = |\mathcal{B}_i^*|$ and $a_1^{(4)} + 2a_2^{(4)} + 3a_3^{(4)} = b_1^* + 2b_2^* + 3b_3^* = n$).

4 Implementation

In this section, we give detailed implemetations of Steps 1-3.

4.1 Implementation of Step 1

Set $M^{(1)} := \mathcal{A}_2^{(1)}$. ($M^{(1)}$ is a matching of $G(\mathcal{A}^{(1)})$ and a maximum matching of the subgraph $G^{(1)} := G(\mathcal{A}^{(1)}) - V(\mathcal{A}_3^{(1)})$ at any time in this step. Initially this is true, since $\mathcal{A}^{(1)} = \mathcal{A}_1^{(1)} + \mathcal{A}_2^{(1)} + \mathcal{A}_3^{(1)}$ is obtained by the greedy algorithm in Step 0.) For this matching $M^{(1)}$, we try to find a vertex-disjoint augmenting path in $G(\mathcal{A}^{(1)})$ between a vertex in $V_1^{(1)} := V(\mathcal{A}_1^{(1)})$ and a vertex in $V_3^{(1)} := V(\mathcal{A}_3^{(1)})$. We assume each path is a subset of C_2. There are two cases as follows.

Case A: There is no augmenting path between a vertex in $V_1^{(1)}$ and a vertex in $V_3^{(1)}$.

Case B: Otherwise (such a path exists).

Case B can be divided into two subcases:

(i) There is a 3-set $(u_3, v_3, w_3) \in \mathcal{A}_3^{(1)}$ such that three vertex-disjoint augmenting paths $P(x_1, u_3)$, $P(y_1, v_3)$ and $P(z_1, w_3)$ join $x_1, y_1, z_1 \in V_1^{(1)}$ and u_3, v_3, w_3; and

(ii) no such a 3-set exists.

In Case A, we set $\mathcal{A}^{(2)} := \mathcal{A}^{(1)}$ and finish Step 1. In Case B, we can update the set cover by doing a semi-local $[0,1]$ improvement and repeat this Step 1 using the new set cover. Details are as follows.

In Case B(i), we set

$$\mathcal{A}_1 := \mathcal{A}_1^{(1)} - \{(x_1), (y_1), (z_1)\},$$

$$\mathcal{A}_2 := \mathcal{A}_2^{(1)} \triangle (P(x_1, u_3) + P(y_1, v_3) + P(z_1, w_3)),$$

$$\mathcal{A}_3 := \mathcal{A}_3^{(1)} - \{(u_3, v_3, w_3)\},$$

where $X \triangle Y = (X - Y) \cup (Y - X)$ (Fig. 2). Then, the updated set $\mathcal{A} = \mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3$ is a disjoint set cover and $|\mathcal{A}_1| = |\mathcal{A}_1^{(1)}| - 3$, $|\mathcal{A}_2| = |\mathcal{A}_2^{(1)}| + 3$, $|\mathcal{A}_3| = |\mathcal{A}_3^{(1)}| - 1$. Thus, $val(\mathcal{A}) = (a, a_1) < (a^{(1)}, a_1^{(1)}) = val(\mathcal{A}^{(1)})$ (since $a < a^{(1)}$) and this update is a semi-local $[0,1]$ improvement.

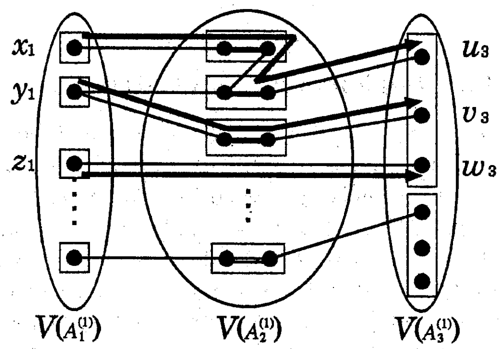

Fig 2: Three augmenting paths.

In Case B(ii), using an augmenting path $P(x_1, u_3)$ joining a vertex x_1 in $V_1^{(1)}$ and a vertex u_3 in a 3-set $(u_3, v_3, w_3) \in \mathcal{A}_3^{(1)}$, we set

$$\mathcal{A}_1 := \mathcal{A}_1^{(1)} - \{(x_1)\},$$

$$\mathcal{A}_2 := (\mathcal{A}_2^{(1)} \triangle P(x_1, u_3)) + \{(v_3, w_3)\},$$

$$\mathcal{A}_3 := \mathcal{A}_3^{(1)} - \{(u_3, v_3, w_3)\}$$

(Fig. 3). Then, the updated set $\mathcal{A} = \mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3$ is a disjoint set cover and $|\mathcal{A}_1| = |\mathcal{A}_1^{(1)}| - 1$, $|\mathcal{A}_2| = |\mathcal{A}_2^{(1)}| + 2$, $|\mathcal{A}_3| = |\mathcal{A}_3^{(1)}| - 1$. Thus, $val(\mathcal{A}) = (a, a_1) < (a^{(1)}, a_1^{(1)}) = val(\mathcal{A}^{(1)})$ (since $a_1 < a_1^{(1)}$ with $a = a^{(1)}$) and this update is a semi-local $[0,1]$ improvement.

In either Case B(i) or Case B(ii), we set $\mathcal{A}^{(1)} := \mathcal{A}$ ($M^{(1)} := \mathcal{A}_2^{(1)}$ is again a maximum matching of the subgraph $G^{(1)} := G(\mathcal{A}^{(1)}) - V(\mathcal{A}_3^{(1)})$, which will be shown later) and repeat this Step 1 until we are in Case A.

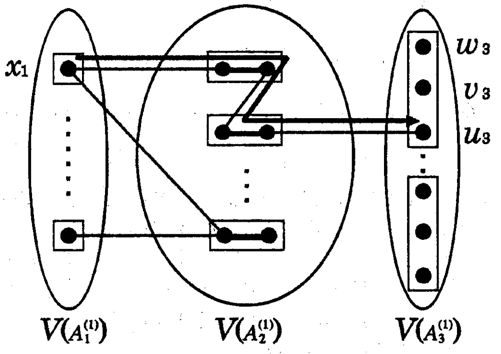

Fig 3: An augmenting path $P(x_1, u_3)$.

4.2 Implementation of Step 2

Let $S_1, S_2, ..., S_p$ be any sequence of the 3-sets in $\mathcal{C}_3 - \mathcal{A}_3^{(2)}$ ($p = |\mathcal{C}_3| - |\mathcal{A}_3^{(2)}| = m_3 - a_3^{(2)}$). We will repeat the following substep for $k := 1$ to p.

Set $G^{(2)} := G(\mathcal{A}^{(2)}) - V(\mathcal{A}_3^{(2)})$, i.e., the graph obtained from $G(\mathcal{A}^{(2)})$ by deleting all vertices in $V(\mathcal{A}_3^{(2)})$. Let $H^{(2)}$ be the graph obtained from $G^{(2)}$ by adding new three vertices $\{u_3', v_3', w_3'\}$ corresponding to $S_k = \{u_3, v_3, w_3\}$ and three edges $\{(u_3, u_3'), (v_3, v_3'), (w_3, w_3')\}$. Set $M^{(2)} := \mathcal{A}_2^{(2)}$. ($M^{(2)}$ is a maximum matching of $G^{(2)}$ at any time in Step 2. Initially, this is true, since $\mathcal{A}^{(2)} = \mathcal{A}_1^{(2)} + \mathcal{A}_2^{(2)} + \mathcal{A}_3^{(2)}$ is obtained by Step 1.) For this matching $M^{(2)}$, we try to find two vertex-disjoint augmenting paths in $H^{(2)}$ one of which joins a vertex in $V_1^{(2)} := V(\mathcal{A}_1^{(2)})$ and a vertex in $\{u_3', v_3', w_3'\}$ and the other of which joins the other two vertices in $\{u_3', v_3', w_3'\}$. There are two cases as follows.

Case A: There are no such augmenting paths.

Case B: Otherwise (such paths exist).

In Case A, we do nothing.

In Case B, we can update the set cover by doing a semi-local [1, 0] improvement. Details are as follows.

We find two such vertex-disjoint augmenting paths $P_1(x_1, u_3')$ and $P_2(v_3', w_3')$ and set

$$\mathcal{A}_1 := \mathcal{A}_1^{(2)} - \{(x_1)\},$$

$$\mathcal{A}_2 := \mathcal{A}_2^{(2)} \triangle (P(x_1, u_3) + P(v_3, w_3)),$$

$$\mathcal{A}_3 := \mathcal{A}_3^{(2)} \cup \{S_k\},$$

where $P(x_1, u_3)$ is the subpath of $P(x_1, u_3')$ joining x_1 and u_3 and $P(v_3, w_3)$ is the subpath of $P(v_3', w_3')$ joining v_3 and w_3 (Fig. 4). Then, the updated set $\mathcal{A} = \mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3$ is a disjoint set cover and $|\mathcal{A}_1| = |\mathcal{A}_1^{(2)}| - 1$, $|\mathcal{A}_2| = |\mathcal{A}_2^{(2)}| - 1$, $|\mathcal{A}_3| = |\mathcal{A}_3^{(2)}| + 1$. Thus, $val(\mathcal{A}) = (a, a_1) < (a^{(2)}, a_1^{(2)}) = val(\mathcal{A}^{(2)})$ (since $a < a^{(2)}$) and this update is a semi-local [1, 0] improvement. Then we set $\mathcal{A}_1^{(2)} := \mathcal{A}_1$, $\mathcal{A}_2^{(2)} := \mathcal{A}_2$ and $\mathcal{A}_3^{(2)} := \mathcal{A}_3$.

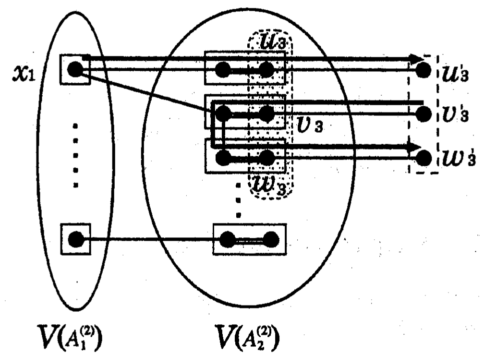

Fig 4: Two augmenting paths.

We set $k := k + 1$ and repeat the above substep until $k = p$. If $k = p + 1$ then we set $\mathcal{A}^{(3)} := \mathcal{A}^{(2)}$ and finish Step 2.

4.3 Implementation of Step 3

Let $S_1, S_2, ..., S_p$ be any sequence of the 3-sets in $\mathcal{C}_3 - \mathcal{A}_3^{(3)}$ ($p = |\mathcal{C}_3| - |\mathcal{A}_3^{(3)}| = m_3 - a_3^{(3)}$). We will repeat the following substep for $k := 1$ to $p - 1$ and $j := k + 1$ to p.

Set $G^{(3)} := G(\mathcal{A}^{(3)}) - V(\mathcal{A}_3^{(3)})$ and $F^{(3)} := G^{(3)} - V(\mathcal{A}_1^{(3)})$. Let $H^{(3)}$ be the graph obtained from $F^{(3)}$ by adding vertices $\{u', v', w', x', y', z'\}$ corresponding to $S_k = \{u, v, w\}$, $S_j = \{x, y, z\}$ and edges $\{(u, u'), (v, v'), (w, w'), (x, x'), (y, y'), (z, z')\}$. Set $M^{(3)} := \mathcal{A}_2^{(3)}$. ($M^{(3)}$ is a maximum matching of $G^{(3)}$, at any time in Step 3. Initially, this is true, since $\mathcal{A}^{(3)} = \mathcal{A}_1^{(3)} + \mathcal{A}_2^{(3)} + \mathcal{A}_3^{(3)}$ is obtained by Step 2.) For this matching $M^{(3)}$, we try to find three vertex-disjoint augmenting paths $P_1(a', b')$, $P_2(c', d')$ and $P_3(e', f')$ with $\{a', b', c', d', e', f'\} = \{u', v', w', x', y', z'\}$ in $H^{(3)}$. There are two cases as follows.

Case A: There are no such augmenting paths.

Case B: Otherwise (such paths exist).

In Case A, we do nothing.

In Case B, we can update the set cover by doing a semi-local $[2,0]$ improvement. Details are as follows.

We find three such vertex-disjoint augmenting paths $P_1(a',b')$, $P_2(c',d')$ and $P_3(e',f')$ and set

$$\mathcal{A}_1 := \mathcal{A}_1^{(3)},$$

$$\mathcal{A}_2 := \mathcal{A}_2^{(3)} \triangle (P_1(a,b) + P_2(c,d) + P_3(e,f)),$$

$$\mathcal{A}_3 := \mathcal{A}_3^{(3)} \cup \{S_k, S_j\},$$

where each $P(g,h)$ is the subpath of $P(g',h')$ joining g and h (Fig. 5). Then, the updated set $\mathcal{A} = \mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3$ is a disjoint set cover and $|\mathcal{A}_1| = |\mathcal{A}_1^{(3)}|$, $|\mathcal{A}_2| = |\mathcal{A}_2^{(3)}| - 3$, $|\mathcal{A}_3| = |\mathcal{A}_3^{(3)}| + 2$. Thus, $val(\mathcal{A}) = (a, a_1) < (a^{(3)}, a_1^{(3)}) = val(\mathcal{A}^{(3)})$ (since $a < a^{(3)}$) and this update is a semi-local $[2,0]$ improvement. Then we set $\mathcal{A}_1^{(3)} := \mathcal{A}_1$ and $\mathcal{A}_2^{(3)} := \mathcal{A}_2$ and $\mathcal{A}_3^{(3)} := \mathcal{A}_3$.

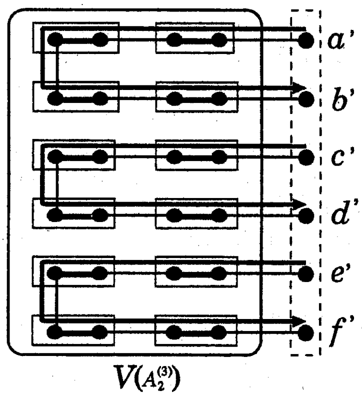

Fig 5: Three augmenting paths.

In either case, we set $j := j + 1$ and repeat the above substep until $j = p$. If $j = p + 1$, then we set $k := k + 1$ and $j := k + 1$ and repeat the above substep until $k = p$. If $k = p + 1$ then we set $\mathcal{A}^{(4)} := \mathcal{A}^{(3)}$ and finish Step 3.

5 Proof of the Correctness

To prove the correctness of the algorithm, we first show the following lemmas about matchings $M^{(1)}$, $M^{(2)}$ and $M^{(3)}$.

Lemma 5.1 *The following (a) and (b) hold.*

(a) At any time in Step 1, $M^{(1)}$ is a maximum matching of $G^{(1)}$ ($M^{(1)} = \mathcal{A}_2^{(1)}$, $G^{(1)} = G(\mathcal{A}^{(1)}) - V(\mathcal{A}_3^{(1)})$).

(b) A 3-set A_3 in $\mathcal{A}_3^{(1)}$ leads to a semi-local $[0,1]$ improvement if and only if there is an augmenting path with respect to $M^{(1)}$ in $G(\mathcal{A}^{(1)})$ joining a vertex in $V(\mathcal{A}_1^{(1)})$ and a vertex in A_3.

Proof. We first consider (a). At the beginning of Step 1, $M^{(1)}$ is a maximum matching of $G^{(1)}$, since we choose $\mathcal{A}_2^{(1)}$ as large as possible by the greedy algorithm in Step 0. Now suppose, for any $k \geq 1$, at the beginning of the k-th iteration, $M^{(1)}$ is a maximum matching of $G^{(1)}$. If Case A occurs in the k-th iteration, then $M^{(1)}$ is a maximum matching of $G^{(1)}$ at the end of the k-th iteration, since no change is made. If Case B occurs, we will show that, at the end of the k-th iteration, $M := \mathcal{A}_2$ is a maximum matching of $G(\mathcal{A}) - V(\mathcal{A}_3)$ (this implies that $M^{(1)}$ is a maximum matching of $G^{(1)}$ at the beginning of the $(k+1)$-th iteration). Note that $G(\mathcal{A}) - V(\mathcal{A}_3)$ has $G^{(1)}$ as an induced subgraph and three more vertices $\{u_3, v_3, w_3\}$. Thus, $M^{(1)}$ is a matching of $G(\mathcal{A}) - V(\mathcal{A}_3)$ and any matching of $G(\mathcal{A}) - V(\mathcal{A}_3)$ has at most three more edges than $M^{(1)}$ since $M^{(1)}$ is a maximum matching of $G^{(1)}$. Thus, M is a maximum matching of $G(\mathcal{A}) - V(\mathcal{A}_3)$ in Case B(i) since M has three more edges than $M^{(1)}$. Case B(ii) implies that $G(\mathcal{A}) - V(\mathcal{A}_3)$ has no matching with cardinality $|M^{(1)}| + 3$ and thus M is a maximum matching of $G(\mathcal{A}) - V(\mathcal{A}_3)$ since $|M| = |M^{(1)}| + 2$.

We next consider (b). If part is trivially true since we can decrease the value of disjoint set cover by deleting A_3 as in Step 1. If A_3 in $\mathcal{A}_3^{(1)}$ leads to a semi-local $[0,1]$ improvement, then we can choose a collection of 2-sets having two more 2-sets than $\mathcal{A}_2^{(1)}$ and this implies that there is such an augmenting path, since A_3 can contain only one disjoint 2-set. □

Lemma 5.2 *At any time in Step 2, $M^{(2)}$ is a maximum matching of $G^{(2)}$ ($G^{(2)} = G(\mathcal{A}^{(2)}) - V(\mathcal{A}_3^{(2)})$, $M^{(2)} = \mathcal{A}_2^{(2)}$).*

Proof. At the beginning of Step 2, $M^{(2)}$ is a maximum matching of $G^{(2)}$ by Lemma 5.1. Now suppose, at the beginning of the k-th iteration, $M^{(2)}$ is a maximum matching of $G^{(2)}$. If Case A occurs in the k-th iteration, then $M^{(2)}$ is a maximum matching of $G^{(2)}$ at the end of the k-th iteration, since no change is made. If Case B occurs, we will show that, at the end of the k-th iteration, $M := \mathcal{A}_2$ is a maximum matching of $G(\mathcal{A}) - V(\mathcal{A}_3)$ (this implies that $M^{(2)}$ is a maximum matching of $G^{(2)}$ at the beginning of the $(k+1)$-th iteration). Note that $G(\mathcal{A})-V(\mathcal{A}_3) = G^{(2)}-\{u_3, v_3, w_3\}$. Thus, a matching M of $G(\mathcal{A}) - V(\mathcal{A}_3)$ is a matching of $G^{(2)}$ and $|M| = |M^{(2)}|-1$. Thus, M is a maximum matching of $G(\mathcal{A}) - V(\mathcal{A}_3)$, since $M' := M \cup \{(v_3, w_3)\}$ is a matching of $G^{(2)}$ with $|M'| = |M^{(2)}|$ and $M^{(2)}$ is a maximum matching of $G^{(2)}$. □

Lemma 5.3 *In Step 2, no 3-set in $\mathcal{A}_3^{(2)}$ leads to a semi-local $[0,1]$ improvement. That is, there is no augmenting path in $G(\mathcal{A}^{(2)})$ joining a vertex in $V(\mathcal{A}_1^{(2)})$ and a vertex in $V(\mathcal{A}_3^{(2)})$.*

Proof. At the beginning of Step 2, the lemma is trivially true, since we have done Step 1. We consider the first iteration and let A_3 be an arbitrary 3-set in $\mathcal{A}_3^{(2)}$. Let $G_{A_3}(\mathcal{A}^{(2)})$ be the graph obtained from $G(\mathcal{A}^{(2)})$ by deleting all vertices $\cup_{A \neq A_3 \in \mathcal{A}_3^{(2)}} A$ and identifying A_3 with a new vertex v_{A_3}. $M^{(2)}$ is a maximum matching of $G_{A_3}(\mathcal{A}^{(2)})$ by Lemma 5.1. If Case A occurs, nothing is changed and A_3 in $\mathcal{A}_3^{(2)}$ does not lead to a semi-local $[0,1]$ improvement. Suppose that Case B occurs. Thus, we can assume, $S_1 = \{u_3, v_3, w_3\}$ and there are two vertex-disjoint alternating paths $P_1(x_1, u_3)$ joining $x_1 \in V(\mathcal{A}_1^{(2)})$ and u_3 and $P_2(v_3, w_3)$ joining v_3 and w_3 in $G(\mathcal{A}^{(2)})$ as in Step 2. If $P_2(v_3, w_3) \neq (v_3, w_3)$ then

$$M := (\mathcal{A}_2^{(2)} \triangle P(v_3, w_3)) \cup \{(v_3, w_3)\},$$

is also maximum matching of $G_{A_3}(\mathcal{A}^{(2)})$ and, for this M, there is no augmenting path in $G_{A_3}(\mathcal{A}^{(2)})$ joining a vertex in $V(\mathcal{A}_1^{(2)})$ and vertex v_{A_3}. Thus, we can consider $M^{(2)} = \mathcal{A}_2^{(2)} = M$ and $P_2(v_3, w_3) = (v_3, w_3)$. This implies $M' := \mathcal{A}_2^{(2)} \triangle P(x_1, u_3)$ is also maximum matching of $G_{A_3}(\mathcal{A}^{(2)})$ and, for this M', there is no augmenting path in $G_{A_3}(\mathcal{A}^{(2)})$ joining a vertex in $V(\mathcal{A}_1 \cup \{u_3\})$ and vertex v_{A_3}.

Let F_{S_1} be the graph obtained from $G_{A_3}(\mathcal{A}^{(2)})$ by identifying S_1 with a new vertex v_{S_1}. Then $M' - \{(v_3, w_3)\} = \mathcal{A}_2$ is a maximum matching of F_{S_1} and there is no augmenting path in F_{S_1} joining a vertex of $\mathcal{A}_1 := \mathcal{A}_1^{(2)} - \{(x_1)\}$, and a vertex v_{A_3} or v_{S_1}.

Thus, by induction, we can obtain the lemma. □

Lemma 5.4 *In Step 2, if S_j is determined in the j-th iteration not to lead to a semi-local $[1,0]$ improvement, then it does not lead to a semi-local $[1,0]$ improvement in k-th iteration for all $k > j$.*

Proof. By induction, we can assume that $j = 1$, $k = 3$ and $S_2 = \{u_3, v_3, w_3\}$ leads to a semi-local $[1,0]$ improvement in the second iteration. We will show that $S_1 = \{x, y, z\}$ does not lead to a semi-local $[1,0]$ improvement in the third iteration. (Of course, S_1 does not lead to a semi-local $[1,0]$ improvement in the second iteration.) Let $\mathcal{A}^{(2)}(i)$ be $\mathcal{A}^{(2)}$ in the i-th iteration and $M^{(2)}(i)$ be $M^{(2)}$ in the i-th iteration, i.e., $M^{(2)}(i) = \mathcal{A}_2^{(2)}(i)$. By the same argument as in the proof of Lemma 5.3, we can assume $(v_3, w_3) \in M^{(2)}(2)$ and $M^{(2)}(3) = (M^{(2)}(2) - \{(v_3, w_3)\}) \triangle P(x_1, u_3)$. Thus, if Case B can be applied for S_1 in the third iteration, we can assume that there are vertex-disjoint augmenting paths $P(x_1, x')$ and $P(y', z')$ in $G_{S_1}^{(2)}(3)$ with respect to $M^{(2)}(3)$, where $G_{S_1}^{(2)}(i)$ is the graph obtained from $G^{(2)}(i)$ by adding vertices $\{x', y', z'\}$ corresponding to $S_1 = \{x, y, z\}$ and three edges $\{(x, x'), (y, y'), (z, z')\}$. But this implies that there are vertex-disjoint augmenting paths $P(x_1, x')$ and $P(y', z')$ in $G_{S_1}^{(2)}(2) = G_{S_1}^{(2)}(1) = H^{(2)}(1)$ with respect to $M^{(2)}(2) = M^{(2)}(1)$. This is a contradiction. Thus, by induction, we can obtain the lemma. □

Lemma 5.5 *The following (a), (b), (c) and (d) hold.*

(a) At any time in Step 3, $M^{(3)}$ is a maximum matching of $G^{(3)}$ ($M^{(3)} = \mathcal{A}_2^{(3)}$, $G^{(3)} = G(\mathcal{A}^{(3)}) - V(\mathcal{A}_3^{(3)})$).

(b) No 3-set in $\mathcal{A}_3^{(3)}$ leads to a semi-local $[0,1]$ improvement.

(c) No 3-set in $C_3 - \mathcal{A}_3^{(3)}$ leads to a semi-local $[1,0]$ improvement.

(d) In Step 3, if a pair of 3-sets (S_k, S_j) is determined in the (k,j)-th iteration not to lead to a semi-local $[2,0]$ improvement, then it does not lead to a semi-local $[2,0]$ improvement at any time after.

Proof. At the beginning of Step 3, $M^{(3)}$ is a maximum matching of $G^{(3)}$ by Lemma 5.2. Now suppose, at the beginning of the k-th iteration, $M^{(3)}$

is a maximum matching of $G^{(3)}$. We also assume the j-th subiteration. If Case A occurs in the j-th subiteration, then $M^{(3)}$ is a maximum matching of $G^{(3)}$ at the end of the j-th subiteration, since no change is made. If Case B occurs, we will show that, at the end of the j-th subiteration, $M := \mathcal{A}_2$ is a maximum matching of $G(\mathcal{A}) - V(\mathcal{A}_3)$ (this implies that $M^{(3)}$ is a maximum matching of $G^{(3)}$ at the beginning of the $(j+1)$-th iteration). Note that $G(\mathcal{A}) - V(\mathcal{A}_3) = G^{(3)} - \{u, v, w, x, y, z\}$ with $S_j = \{x, y, z\}$ and $S_k = \{u, v, w\}$. Thus, a matching M of $G(\mathcal{A}) - V(\mathcal{A}_3)$ is a matching of $G^{(3)}$ and $|M| = |M^{(3)}| - 3$. By symmetry, we can assume $a = u$, $b = x$, $c = v$, $d = w$, $e = y$, $f = z$ and $P_2(c, d) = (v, w)$, $P_3(e, f) = (y, z)$. This is because if $P_2(c, d) = P(v, y)$ and $P_3(e, f) = (w, z)$ then $P(v, y) + P(w, z) + \{(v, w), (y, z)\}$ is an alternating cycle and we can change matching edges and nonmatching edges along this cycle and assume $\{(v, w), (y, z)\}$ are matching edges. Similarly, if $P_2(c, d) = P(v, w) \neq \{(v, w)\}$ and $P_3(e, f) = P(y, z) \neq \{(y, z)\}$ then $P(v, w) \cup \{(v, w)\}$, $P(y, z) \cup \{(y, z)\}$ are both disjoint alternating cycles and we can change matching edges and nonmatching edges along these cycles and assume $\{(v, w), (y, z)\}$ are matching edges of $M^{(3)}$. Furthermore, even if we add (x, u) to the graph $G^{(3)}$, then $M^{(3)}$ remains a maximum matching which can be shown by using (c). Thus, M is a maximum matching of $G(\mathcal{A}) - V(\mathcal{A}_3)$, since $M' := M \cup \{(u, x), (v, w), (y, z)\}$ is a matching of $G^{(3)}$ with $|M'| = |M^{(3)}|$ and $M^{(3)}$ is a maximum matching of $G^{(3)}$. By induction on j (and k), we can obtain (a).

The statements (b)-(d) can be similarly proved. $\square$

Now we are ready to analyze the peformance of the approximation algorithm of Duh and Fürer. They used the *comparison graph* defined as follows [1]. It is a bipartite multi-graph which has an A-vertex for every set chosen by the algorithm and has a B-vertex for every set included in a fixed optimal set cover. The elements of set V are represented by the edges. If the set corresponding to an A-vertex intersects the set corresponding to a B-vertex in k-elements, then there are k edges between the two vertices. We call an A-vertex corresponding to i-set to be an A_i-vertex and a B-vertex corresponding to i-set to be a B_i-vertex. We also use this comparison graph for the set cover $\mathcal{A}^{(4)}$ and a fixed optimal set cover $\mathcal{B}^*$.

To prove that $|\mathcal{A}^{(4)}| \leq \frac{4}{3}|\mathcal{B}^*|$, we use almost the same lemmas as in Duh and Fürer [1]. The first lemma is trivailly true since we consider only disjoint set covers.

Lemma 5.6 $a_1^{(j)} + 2a_2^{(j)} + 3a_3^{(j)} = b_1^* + 2b_2^* + 3b_3^* = n$ *for any* $j = 1, 2, 3, 4$. $\square$

The next lemma plays a critical role and the corresponding lemma given by Duh and Fürer contains a small hole in their proof.

Lemma 5.7 $a_1^4 = a_1^{(3)} \leq b_1^*$.

Proof. There is no path between A_1-vertex to A_3-vertex or A_1 vertex, since we have done Steps 0 and 1. Now consider a connected component containing A_1-vertex. Clearly this connected component contains only one A_1-vertex. If this connected component has no B_3-vertex, then it is a path whose other endvertex is a B_1-vertex. Otherwise (if the connected component contains B_3-vertex), we consider the two cases: (i) it contains B_1-vertex, or (ii) it contains no B_1 vertex. We have to consider only Case (ii). In this case, we can do a semi-local $[1, 0]$ improvement. However, since we have done Step 2, this cannot happen. $\square$

The following lemma can be proved in the same way as Duh and Fürer.

Lemma 5.8 $a_1^{(4)} + a_2^{(4)} \leq b_1^* + b_2^* + b_3^*$. $\square$

By the preceding three lemmas, we have

$$3(a_1^{(4)} + a_2^{(4)} + a_3^{(4)}) = 3b_1^* + 3b_2^* + 4b_3^* \leq 4(b_1^* + b_2^* + b_3^*).$$

Thus, our simplified implementation of Duh and Fürer algorithm is also a $\frac{4}{3}$-approximation algorithm.

6 Time Complexity Analysis

In this section we analyze the time complexity of our implementation. There we have to find two or three vertex-disjoint augmenting paths. We first note here that we have only to find such paths one by one. For example, in Step 1, we look for three vertex-disjoint augmenting paths $P(x_1, u_3)$,

$P(y_1, v_3)$, $P(z_1, w_3)$ joining $x_1, y_1, z_1 \in V_1^{(1)}$ and u_3, v_3, w_3 in $\mathcal{A}_3^{(1)}$. We first find an augmenting path $P(x, u)$ with respect to $M^{(1)}$ joining a vertex $x \in \{x_1, y_1, z_1\}$ and a vertex $u \in \{u_3, v_3, w_3\}$. Then, for $M' := M^{(1)} \triangle P(x, u)$, we find an augmenting path $P(y, v)$ joining a vertex $y \in \{x_1, y_1, z_1\} - \{x\}$ and a vertex $v \in \{u_3, v_3, w_3\} - \{u\}$. Again, for $M'' := M' \triangle P(y, v)$, we find another augmenting path $P(z, w)$ joining a vertex $z \in \{x_1, y_1, z_1\} - \{x, y\}$ and a vertex $w \in \{u_3, v_3, w_3\} - \{u, v\}$ and let $M := M'' \triangle P(z, w)$. Then $M^{(1)} \triangle M$ contains three vertex-disjoint augmenting paths with respect to $M^{(1)}$ joining vertices in $\{x_1, y_1, z_1, u_3, v_3, w_3\}$, since all vertices except these six vertices are matched by both matchings $M^{(1)}$ and M and these six vertices are matched by only M. In Step 1, there is no augmenting path joining vertices in $V_1^{(1)}$, we can assume they are $P(x_1, u_3)$, $P(y_1, v_3)$, $P(z_1, w_3)$ joining $x_1, y_1, z_1 \in V_1^{(1)}$ and u_3, v_3, w_3 in $\mathcal{A}_3^{(1)}$.

Based on this observation, we can also find two vertex-disjoint augmenting paths in Step 2 and three vertex-disjoint augmenting paths in Step 3.

Thus, Step 1 requires $O(\min\{a_1^{(1)}, a_3^{(1)}\})$ iterations and each iteration requires at most three augmenting path findings. Similarly Step 2 requires $O(m_3 - a_3^{(2)})$ iterations and each iteration requires at most two augmenting path findings. Step 3 requires $O((m_3 - a_3^{(3)})^2)$ iterations and each iteration requires at most three augmenting path findings. Thus, if we denote by $A(n)$ the time required to find an augmenting path in a graph with $O(n)$ vertices, the entire time required in our implementation is $O((\min\{a_1^{(1)}, a_3^{(1)}\} + (m_3 - a_3^{(2)}) + (m_3 - a_3^{(3)})^2)A(n))$ time. This is also expressed by $O(m^2 A(n))$.

7 Concluding Remarks

We have presented a simple implementation of the $\frac{4}{3}$-approximation algorithm for 3-Set Cover by Duh and Fürer [1] and analyzed its time complexity. Since our implementation for Step 3 does not seem so efficient, the time complexity for Step 3 will be reduced based on matching theory and data structures. We believe, however, this paper first gives a basis for the research in finding more efficient approximation algorithms for 3-Set Cover. Even if we neglect Step 3, we have Lemmas 5 and 6. Thus we can obtain a $\frac{3}{2}$-approximation algorithm with time complexity $O(mA(n))$ for 3-Set Cover. For general $k \geq 4$, Duh and Fürer also gave an $(\mathcal{H}_k - \frac{1}{2})$-approximation algorithm where semi-local $(2, 1)$ improvements are used for i-sets with $i \leq 3$ on uncovered elements. Thus, our implementation can also be applied to their approximation algorithm for k-Set Cover.

Acknowledgments.

The last author was supported in part by Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan and the Institute of Science and Engineering of Chuo University.

References

[1] R-C. Duh and M. Fürer, Approximation of k-set cover by semi-local optimization, *Proc. 29th ACM Symposium on Theory of Computing*, 1997, pp.256–264.

[2] O. Goldschimidt, D. S. Hochbaum and G. Yu, A modified greedy heuristics for set covering problem with improved worst case bound, *Information Processing Letters*, 48 (1993), pp.305–310.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman and Company, 1979.

[4] M. M. Halldórsson, Approximating discrete collections via local improvements, *Proc. 6th ACM-SIAM Symposium on Discrete Algorithms*, 1996, pp.160–169.

[5] M. M. Halldórsson, Approximating k-set cover and complementary graph coloring, In *Proc. 5th Conference on Integer Programming and Combinatorial Optimization*, 1996, pp.118–131.

[6] D. S. Johnson, Approximation algorithms for combinatorial problems, *Journal of Computer and System Sciences*, 9 (1974), pp.256–278.

[7] L. Lovász, On the ratio of optimal integral and fractional covers, *Discrete Mathematics*, 13 (1975), pp.383–390.