数理解析研究所講究録
第 864 巻 1994 年 74-82


# Approximate-Weight-Splitting Algorithm for a Minimum Common Base of a Pair of Matroids

東京工業大学 繁野麻衣子 (Maiko SHIGENO)
東京大学 岩田 覚 (Satoru IWATA)

**Abstract:** This paper deals with the problem of finding a minimum-weight common base of a pair of matroids $M_1$ and $M_2$ both coupled with a weight vector $\boldsymbol{w}$. In each iteration, the approximate-weight-splitting algorithm computes minimum-weight bases $B_1$ of $M_1$ with a weight vector $\boldsymbol{u}$ and $B_2$ of $M_2$ with a weight vector $\boldsymbol{v}$, where $\boldsymbol{u}+\boldsymbol{v}$ is approximately equal to $\boldsymbol{w}$, and then updates the split weight vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ unless $B_1 = B_2$. The algorithm attains an approximately minimum-weight common base $B_1 = B_2$ in pseudo-polynomial iterations. Incorporating cost scaling, we improve the algorithm to run in weakly-polynomial time $\mathrm{O}((f+n)nr\log(rW))$. Here $n$ is the cardinality of the ground set, $r$ is the rank of both matroids, $W$ is a maximum absolute value of the weights and $f$ is the time needed to find a circuit or a cocircuit in a given subset.

**Keywords:** combinatorial optimization, matroid intersection, auction algorithm, cost scaling

## 1 Introduction

Given a pair of matroids $M_1$ and $M_2$ on the same ground set coupled with a weight vector $\boldsymbol{w}$, find a common base of minimum total weight. This problem, which is referred to as the *minimum common base problem*, is equivalent to the weighted matroid intersection problem or the independent assignment problem which has been extensively studied by many researchers. See the papers J. Edmonds [5, 6], E. L. Lawler [11], M. Iri and N. Tomizawa [10], S. Fujishige [8], A. Frank [7], C. Brezovec, G. Cornuéjols and F. Glover [4] and so on. Most of the algorithms studied there are based on the idea of the "auxiliary network" (or border graph) and repeated applications of efficient network algorithms.

In this paper, we propose a different kind of algorithm for the minimum common base problem. The algorithm works by splitting the weights of the elements and adjusting the split weights to get an approximately optimal common base. In each iteration, our algorithm computes minimum-weight bases $B_1$ of $M_1$ with a weight vector $\boldsymbol{u}$ and $B_2$ of $M_2$ with a weight vector $\boldsymbol{v}$, where $\boldsymbol{u}+\boldsymbol{v}$ is approximately equal to $\boldsymbol{w}$, and then updates the split weight vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ unless $B_1 = B_2$. The algorithm attains an approximately optimal common base $B_1 = B_2$ in pseudo-polynomial iterations.

Among previously known algorithms, the present algorithm is similar to that of A. Frank [7] and can be interpreted as an approximate version of his algorithm. Hence it is called an *approximate-weight-splitting* algorithm. On the other hand, it is closely related to the *auction* algorithm proposed by D. P. Bertsekas [1] for the assignment problem, which is a special case of the minimum common base problem (see Remark 1 in Section 3 for the relation).

Although the original auction algorithm has pseudo-polynomial time complexity, it has been improved to run in polynomial time with a cost scaling technique by D. P. Bertsekas and J. Eckstein [3] (see also [2]). Analogously, we will improve our approximate-weight-splitting algorithm for the minimum common base problem to run in polynomial time by a cost scaling technique. The time complexity of our algorithm is $O((f+n)nr\log(rW))$, where $n$ is the cardinality of the ground set, $r$ is the rank of both matroids, $W$ is a maximum absolute value of the weight of an element and $f$ is the time needed to find a circuit or a cocircuit in a given subset.

The outline of this paper is as follows. Section 2 provides preliminaries of matroid theory and defines the minimum common base problem. In Section 3, we propose a naive approximate-weight-splitting algorithm and discuss its correctness and finiteness. Section 4 is devoted to improvements with respect to the time complexity.

## 2 Preliminaries

In this section, we recapitulate several basic concepts of matroid theory and formulate the minimum common base problem.

Let $E$ be a finite set and $\mathcal{B}$ a nonempty family of subsets of $E$. Then $M=(E,\mathcal{B})$ is said to be a *matroid*, if for any $B_1, B_2 \in \mathcal{B}$ and $e_1 \in B_1 \backslash B_2$, there exists $e_2 \in B_2 \backslash B_1$ such that $(B_1 \cup \{e_2\}) \backslash \{e_1\} \in \mathcal{B}$. A member of $\mathcal{B}$ is called a *base* of $M$. All bases of a matroid have the same cardinality, which is the *rank* of the matroid.

A minimal subset of $E$ that is not a subset of any base is called a *circuit* of $M$. For any $B \in \mathcal{B}$ and $e \in E \backslash B$, there exists a unique circuit $C(B|e)$ of $M$ such that $e \in C(B|e) \subseteq B \cup \{e\}$. This circuit $C(B|e)$ is called the *fundamental circuit* of $e$ in the base $B$. A minimal subset of $E$ that has nonempty intersection with every base of $M$ is called a *cocircuit* of $M$. If $B$ is a base of $M$ and $e \in B$, there exists a unique cocircuit $C^*(B|e)$ of $M$ such that $e \in C^*(B|e) \subseteq (E \backslash B) \cup \{e\}$. We call $C^*(B|e)$ the *fundamental cocircuit* of $e$ with respect to the base $B$.

Let $\boldsymbol{w}$ be a rational weight vector indexed by $E$. The weight of a subset $F \subseteq E$ is defined by $\boldsymbol{w}(F) = \sum_{e \in F} \boldsymbol{w}(e)$. Given a matroid $M=(E,\mathcal{B})$ and a weight vector $\boldsymbol{w} \in \mathbf{Q}^E$, we say that $B \in \mathcal{B}$ is a *$\boldsymbol{w}$-minimum base* of $M$ if $\boldsymbol{w}(B) \leq \boldsymbol{w}(B')$ for every $B' \in \mathcal{B}$. It is well known that a $\boldsymbol{w}$-minimum base may be found by the *greedy algorithm*, which is based on the following characterization of $\boldsymbol{w}$-minimum bases.

**Lemma 1.** *Let $M=(E,\mathcal{B})$ be a matroid and $\boldsymbol{w} \in \mathbf{Q}^E$ a weight vector. For a base*

$B$ of $M$, *the following statements are equivalent:*

(1) $B$ *is a* $\boldsymbol{w}$*-minimum base;*

(2) $e \in E \backslash B$ *implies* $\boldsymbol{w}(e) \geq \boldsymbol{w}(e')$ *for all* $e' \in C(B|e)$;

(3) $e \in B$ *implies* $\boldsymbol{w}(e) \leq \boldsymbol{w}(e')$ *for all* $e' \in C^*(B|e)$.

We now define the *minimum common base problem* which we treat in this paper. Let $M_1 = (E, \mathcal{B}_1)$ and $M_2 = (E, \mathcal{B}_2)$ be a pair of matroids on the common ground set $E$ with base families $\mathcal{B}_1$ and $\mathcal{B}_2$. A subset $B \subseteq E$ which belongs to both $\mathcal{B}_1$ and $\mathcal{B}_2$ is said to be a *common base* of the matroids $M_1$ and $M_2$. Given a weight vector $\boldsymbol{w} \in \mathbf{Q}^E$, the *minimum common base problem* is to find a common base $B$ such that $\boldsymbol{w}(B) \leq \boldsymbol{w}(B')$ for every $B' \in \mathcal{B}_1 \cap \mathcal{B}_2$. The following rather obvious fact gives a basic idea of our algorithm.

**Lemma 2.** *Let* $M_1 = (E, \mathcal{B}_1)$ *and* $M_2 = (E, \mathcal{B}_2)$ *be matroids and* $\boldsymbol{w} \in \mathbf{Q}^E$ *a weight vector. If* $\boldsymbol{u}, \boldsymbol{v}$ *are weight vectors satisfying* $\boldsymbol{u} + \boldsymbol{v} = \boldsymbol{w}$ *and* $B \in \mathcal{B}_1 \cap \mathcal{B}_2$ *is a* $\boldsymbol{u}$*-minimum base of* $M_1$ *and a* $\boldsymbol{v}$*-minimum base of* $M_2$ *at the same time, then* $B$ *is a minimum common base with respect to* $\boldsymbol{w}$.

Throughout this paper, we assume that the weight vector $\boldsymbol{w}$ is integral and there exists at least one common base of the matroids $M_1$ and $M_2$. We denote the cardinality of the ground set and the rank of both matroids by $n$ and $r$, respectively.

## 3 The Approximate-Weight-Splitting Algorithm

In this section, we propose an algorithm for the minimum common base problem.

Suppose that the weight vector $\boldsymbol{w}$ is split as $\boldsymbol{w} = \boldsymbol{u} + \boldsymbol{v}$. The outline of our algorithm is described below. Set $\overline{\boldsymbol{u}} = \boldsymbol{u}$ and $\overline{\boldsymbol{v}} = \boldsymbol{v}$. A $\overline{\boldsymbol{u}}$-minimum base $B_1$ of the matroid $M_1$ and a $\overline{\boldsymbol{v}}$-minimum base $B_2$ of the matroid $M_2$ is obtained by the greedy algorithm. If $B_1$ and $B_2$ coincide with each other, $B_1 (= B_2)$ is a desired minimum common base. Otherwise, our algorithm modifies the two weight vectors $\overline{\boldsymbol{u}}$ and $\overline{\boldsymbol{v}}$ in an appropriate manner and maintains a pair of minimum bases with respect to the modified weight vectors. We repeat this process until the two minimum bases coincide. The weight vectors $\overline{\boldsymbol{u}}$ and $\overline{\boldsymbol{v}}$ are modified in one component by a given constant $\varepsilon (> 0)$ so that the intersection of the two minimum bases may be extended. Note that it is not difficult to update the minimum bases of the matroids with respect to the modified weight vectors by finding a fundamental circuit or a fundamental cocircuit.

We denote by $greedy(E, \mathcal{B}, \boldsymbol{w})$ a function which finds a $\boldsymbol{w}$-minimum base of the matroid $M = (E, \mathcal{B})$. Let $C_1^*(B|e)$ be the fundamental cocircuit with respect to $B \in \mathcal{B}_1$ and $e \in B$ in the matroid $M_1$, and $C_2(B|e)$ the fundamental circuit with respect to $B \in \mathcal{B}_2$ and $e \in E \backslash B$ in the matroid $M_2$. An artificial vector $\boldsymbol{p} \in \mathbf{Q}^E$ is introduced for the sake of analysis. The approximate-weight-splitting algorithm for the minimum common base problem is as follows.

**Algorithm** *approximate-weight-splitting*
*input*: a pair of matroids $M_1 = (E, \mathcal{B}_1)$ and $M_2 = (E, \mathcal{B}_2)$ ,
split weight vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbf{Q}^E$ such that $\boldsymbol{u} + \boldsymbol{v} = \boldsymbol{w}$ and a constant $\varepsilon (> 0)$

1: **begin**
2: $\overline{\boldsymbol{u}} := \boldsymbol{u},\ \overline{\boldsymbol{v}} := \boldsymbol{v},\ \boldsymbol{p} := \mathbf{0}$ ;
3: $B_1 \leftarrow \mathit{greedy}(E, \mathcal{B}_1, \overline{\boldsymbol{u}})$ ;
4: $\boldsymbol{p}(e) := \boldsymbol{p}(e) + \varepsilon$ for all $e \in B_1$ ;
5: $B_2 \leftarrow \mathit{greedy}(E, \mathcal{B}_2, \overline{\boldsymbol{v}})$ ;
6: **while** $\exists e^- \in B_1 \backslash B_2$ **do**
7: **begin**
8: $\overline{\boldsymbol{u}}(e^-) := \overline{\boldsymbol{u}}(e^-) + \varepsilon$ ;
9: $e^+ \leftarrow \arg\min\{\overline{\boldsymbol{u}}(e) \mid e \in C_1^*(B_1 | e^-)\}$ ;
10: $B_1 \leftarrow (B_1 \backslash \{e^-\}) \cup \{e^+\}$ ;
11: $\boldsymbol{p}(e^+) := \boldsymbol{p}(e^+) + \varepsilon$ ;
12: **if** $e^+ \notin B_2$ **then**
13: **begin**
14: $\overline{\boldsymbol{v}}(e^+) := \overline{\boldsymbol{v}}(e^+) - \varepsilon$ ;
15: $e' \leftarrow \arg\max\{\overline{\boldsymbol{v}}(e) \mid e \in C_2(B_2 | e^+)\}$ ;
16: $B_2 \leftarrow (B_2 \cup \{e^+\}) \backslash \{e'\}$
17: **end**
18: **end**;
19: output $B_1 (= B_2)$ as a common base $B$
20: **end**.

Observing the approximate-weight-splitting algorithm above, we see the following properties.

**Observation 3.** *The subsets $B_1$ and $B_2$ are a $\overline{\boldsymbol{u}}$-minimum base and a $\overline{\boldsymbol{v}}$-minimum base, respectively, at the end of each iteration as well as right after line* 11.

**Observation 4.** *No element enters $B_2 \backslash B_1$ during any iteration. As a consequence, it always holds that $\boldsymbol{p}(B_2 \backslash B_1) = \mathbf{0}$.*

**Observation 5.** *At the end of each iteration of the approximate-weight-splitting algorithm, the following conditions are satisfied:*

$$\boldsymbol{u}(e) + \boldsymbol{p}(e) = \overline{\boldsymbol{u}}(e) + \varepsilon, \qquad \forall e \in B_1, \tag{3.1}$$

$$\boldsymbol{u}(e) + \boldsymbol{p}(e) = \overline{\boldsymbol{u}}(e), \qquad \forall e \in E \backslash B_1, \tag{3.2}$$

$$\overline{\boldsymbol{v}}(e) - \varepsilon \le \boldsymbol{v}(e) - \boldsymbol{p}(e) \le \overline{\boldsymbol{v}}(e), \quad \forall e \in E. \tag{3.3}$$

Observation 5 implies that the following inequalities hold at the end of each iteration.

$$\boldsymbol{w}(e) - \varepsilon \le \overline{\boldsymbol{u}}(e) + \overline{\boldsymbol{v}}(e) \le \boldsymbol{w}(e) + \varepsilon, \quad \forall e \in E. \tag{3.4}$$

The inequalities (3.4) are also satisfied right after line 11, which is trivial for an element except for $e^+$. Since the element $e^+$ satisfies the condition (3.1) and

$$\overline{\boldsymbol{v}}(e^+) - 2\varepsilon \le \boldsymbol{v}(e^+) - \boldsymbol{p}(e^+) \le \overline{\boldsymbol{v}}(e^+) - \varepsilon$$

right after line 11, the inequalities (3.4) also hold for the element $e^+$.

We now discuss an error bound of a common base obtained by the approximate-weight-splitting algorithm. Let $B_{\mathrm{opt}}$ be a minimum common base with respect to the weight $\boldsymbol{w}(= \boldsymbol{u} + \boldsymbol{v})$.

**Lemma 6.** *Assume that the approximate-weight-splitting algorithm has terminated with a common base $B$. Then*

$$\boldsymbol{w}(B) \leq \boldsymbol{w}(B_{\mathrm{opt}}) + 2\varepsilon r.$$

**Proof.** It follows from the inequalities (3.4) and Observation 3 that

$$\boldsymbol{w}(B) \;\leq\; \overline{\boldsymbol{u}}(B) + \overline{\boldsymbol{v}}(B) + \varepsilon r \leq \overline{\boldsymbol{u}}(B_{\mathrm{opt}}) + \overline{\boldsymbol{v}}(B_{\mathrm{opt}}) + \varepsilon r \leq \boldsymbol{w}(B_{\mathrm{opt}}) + 2\varepsilon r. \quad \blacksquare$$

Since the weight vector $\boldsymbol{w}$ is assumed to be integral, Lemma 6 implies that the approximate-weight-splitting algorithm finds a minimum common base if $\varepsilon < 1/(2r)$.

We show the number of iterations of the approximate-weight-splitting algorithm in the worst case.

**Lemma 7.** *Suppose $B_u$ is a $\boldsymbol{u}$-minimum base of the matroid $M_1$ and $B_v$ is a $\boldsymbol{v}$-minimum base of the matroid $M_2$. Then the number of iterations of the approximate-weight-splitting algorithm is less than $n\lfloor \frac{\boldsymbol{w}(B_{\mathrm{opt}}) - (\boldsymbol{u}(B_u) + \boldsymbol{v}(B_v))}{\varepsilon} \rfloor + 2nr$.*

**Proof.** Since the value of $\boldsymbol{p}(E)$ increases exactly by $\varepsilon$ in every iteration, we will show an upper bound of $\boldsymbol{p}(E)$.

Let us check the value of $\boldsymbol{p}(e^+)$ right after line 11. If $e^+ \in B_2$, then $e^+ \neq e^-$ and $\boldsymbol{p}(e^+) = \varepsilon$. Otherwise,

$$\boldsymbol{p}(e^+) \leq \boldsymbol{p}(B_1 \backslash B_2) = \boldsymbol{p}(B_1) - \boldsymbol{p}(B_2) + \boldsymbol{p}(B_2 \backslash B_1) = \boldsymbol{p}(B_1) - \boldsymbol{p}(B_2).$$

From Observation 3 and inequalities (3.1)–(3.4), we have

$$\begin{aligned}
\boldsymbol{p}(B_1) - \boldsymbol{p}(B_2) &\leq \overline{\boldsymbol{u}}(B_1) + \varepsilon r - \boldsymbol{u}(B_1) + \overline{\boldsymbol{v}}(B_2) - \boldsymbol{v}(B_2) \\
&\leq \overline{\boldsymbol{u}}(B_{\mathrm{opt}}) + \overline{\boldsymbol{v}}(B_{\mathrm{opt}}) - \boldsymbol{u}(B_u) - \boldsymbol{v}(B_v) + \varepsilon r \\
&\leq \boldsymbol{w}(B_{\mathrm{opt}}) - \boldsymbol{u}(B_u) - \boldsymbol{v}(B_v) + 2\varepsilon r.
\end{aligned}$$

Hence the number of iterations in which the same element becomes $e^+$ is at most $\lfloor \{\boldsymbol{w}(B_{\mathrm{opt}}) - (\boldsymbol{u}(B_u) + \boldsymbol{v}(B_v))\}/\varepsilon \rfloor + 2r$, and the total number of iterations is less than $n\lfloor \{\boldsymbol{w}(B_{\mathrm{opt}}) - (\boldsymbol{u}(B_u) + \boldsymbol{v}(B_v))\}/\varepsilon \rfloor + 2nr$. $\blacksquare$

If we put $W = \max_{e \in E} |\boldsymbol{w}(e)|$, then Lemmas 6 and 7 imply the following corollary.

**Corollary 8.** *If* $u = w$, $v = 0$ *and* $\varepsilon = \frac{1}{3r}$, *the approximate-weight-splitting algorithm terminates within* $n(6r^2W + 2r)$ *iterations and finds a minimum common base.*

**Remark 1.** *Let us mention now the relation between our approximate-weight-splitting algorithm and the auction algorithm for the assignment problem originally proposed by D. P. Bertsekas [1]. It has been shown by T. Matsui and K. Shibata [12] that the auction algorithm still works for a minimum common base problem in which one of the matroids is a partition matroid such as the minimum arborescence problem. A minimum common base problem with* $M_1$ *and* $M_2$ *can be reduced to another minimum common base problem with a partition matroid and* $M_1 \oplus M_2^*$, *i.e., the direct sum of* $M_1$ *and the dual of* $M_2$, *which fact leads us to an auction-type algorithm essentially tantamount to the approximate-weight-splitting algorithm.*

## 4 Scaling Approach

As we have seen in Section 3, the error bound and the number of iterations of the approximate-weight-splitting algorithm depend on the value of $\varepsilon$. For sufficiently small $\varepsilon$, the algorithm obtains a minimum common base, however the number of iterations becomes large. The number of iterations also depends on the initial value of the split weight vectors $u$ and $v$. If it starts with split weight vectors close to the desired ones, we might expect that the number of iterations to find a minimum common base will be relatively small.

The observations in the preceding paragraph suggest an approach called *scaling*, which has been extensively used to derive polynomial time algorithms for combinatorial optimization problems. In this case, we employ *cost scaling* which is often called *$\varepsilon$-scaling*. Our scaling algorithm starts with a large value of $\varepsilon$ which is given by $\varepsilon_0$ and successively reduces $\varepsilon$ up to an ultimate value $\mu$. The scaling algorithm performs a number of *scaling phases*. Each scaling phase reduces the value of $\varepsilon$ and calls the approximate-weight-splitting algorithm in which the initial split weight vectors are modified by the artificial vector $p$ obtained at the end of the last scaling phase. The scaling algorithm is given below.

**Algorithm** *scaling*
*input*: a pair of matroids $M_1 = (E, \mathcal{B}_1)$ and $M_2 = (E, \mathcal{B}_2)$, a weight vector $\boldsymbol{w} \in \mathbf{Z}^E$, an initial value $\varepsilon_0 (> 0)$ and an ultimate value $\mu (> 0)$
**begin**
  $\varepsilon := \varepsilon_0, \boldsymbol{u} := \boldsymbol{w}, \boldsymbol{v} := \mathbf{0}, \boldsymbol{p} := \mathbf{0}$ ;
  **while** $\varepsilon \geq \mu$ **do**
  **begin**
    $\varepsilon := \varepsilon/2$ ;
    *approximate-weight-splitting*$(\boldsymbol{u}, \boldsymbol{v}, \varepsilon; \boldsymbol{p}, B)$ ;
    $\boldsymbol{u} := \boldsymbol{u} + \boldsymbol{p}, \boldsymbol{v} := \boldsymbol{v} - \boldsymbol{p}$
  **end**;
  output a common base $B$
**end.**

In the above algorithm, $(\cdot\,;\,\cdot)$ designates that the left arguments are inputs and the right arguments are outputs of the subprocedure.

Let $\boldsymbol{u}^{(k)}, \boldsymbol{v}^{(k)}$ and $\varepsilon^{(k)}$ be inputs of the approximate-weight-splitting algorithm of the $k$-th scaling phase. Obviously, $\boldsymbol{u}^{(k)} + \boldsymbol{v}^{(k)} = \boldsymbol{w}$. Assume that the approximate-weight-splitting algorithm of the $k$-th scaling phase has terminated with a common base $B^{(k)}$. Then we have $\boldsymbol{w}(B^{(k)}) \leq \boldsymbol{w}(B_{\mathrm{opt}}) + 2\varepsilon^{(k)} r$, which may be proved in the same way as the proof of Lemma 6. (Recall that $B_{\mathrm{opt}}$ denotes a minimum common base.) From the assumption that $\boldsymbol{w}$ is integral, the scaling algorithm finds a minimum common base, if we set $\mu = 1/(2r)$.

We now analyze the complexity of the scaling algorithm. Suppose that $\varepsilon_0 = W$. Clearly, the scaling algorithm executes $\lfloor \log_2 (W/\mu) \rfloor + 1$ scaling phases. We compute the number of iterations of the approximate-weight-splitting algorithm of each scaling phase.

**Lemma 9.** *The approximate-weight-splitting algorithm of the $k$-th scaling phase terminates within $6nr$ iterations, provided that $\varepsilon_0 = W$.*

**Proof.** Let $B_u^{(k)}$ and $B_v^{(k)}$ be a $\boldsymbol{u}^{(k)}$-minimum base of $M_1$ and a $\boldsymbol{v}^{(k)}$-minimum base of $M_2$, respectively.

Since $\boldsymbol{u}^{(1)} = \boldsymbol{w}$ and $\boldsymbol{v}^{(1)} = \mathbf{0}$, we have $\boldsymbol{w}(B_{\mathrm{opt}}) - (\boldsymbol{u}^{(1)}(B_u^{(1)}) + \boldsymbol{v}^{(1)}(B_v^{(1)})) \leq 2rW$. From Lemma 7 and $\varepsilon^{(1)} = W/2$, the number of iterations of the approximate-weight-splitting algorithm at the first scaling phase is less than $n \lfloor \frac{2rW}{W/2} \rfloor + 2nr = 6nr$.

Consider the $k$-th scaling phase where $k \geq 2$. We denote a common base obtained at the end of the $k$-th scaling phase by $B^{(k)}$, and modified weight vectors and the artificial vector at the end of the approximate-weight-splitting algorithm of the $k$-th scaling phase by $\overline{\boldsymbol{u}}^{(k)}, \overline{\boldsymbol{v}}^{(k)}$ and $\boldsymbol{p}^{(k)}$. From Observation 3 and the inequalities (3.1)–(3.4), we have

$$\boldsymbol{w}(B_{\mathrm{opt}}) \;\leq\; \boldsymbol{w}(B^{(k-1)})$$

$$\begin{aligned}
& \leq \overline{\boldsymbol{u}}^{(k-1)}(B^{(k-1)}) + \overline{\boldsymbol{v}}^{(k-1)}(B^{(k-1)}) + \varepsilon^{(k-1)}r, \\
\boldsymbol{u}^{(k)}(B_u^{(k)}) + \boldsymbol{v}^{(k)}(B_v^{(k)}) & = \boldsymbol{u}^{(k-1)}(B_u^{(k)}) + \boldsymbol{p}^{(k-1)}(B_u^{(k)}) + \boldsymbol{v}^{(k-1)}(B_v^{(k)}) - \boldsymbol{p}^{(k-1)}(B_v^{(k)}) \\
& \geq \overline{\boldsymbol{u}}^{(k-1)}(B_u^{(k)}) + \overline{\boldsymbol{v}}^{(k-1)}(B_v^{(k)}) - \varepsilon^{(k-1)}r \\
& \geq \overline{\boldsymbol{u}}^{(k-1)}(B^{(k-1)}) + \overline{\boldsymbol{v}}^{(k-1)}(B^{(k-1)}) - \varepsilon^{(k-1)}r.
\end{aligned}$$

Hence it follows from Lemma 7 and $\varepsilon^{(k-1)} = 2\varepsilon^{(k)}$ that the number of iterations of the approximate-weight-splitting algorithm at the $k$-th scaling phase $(k \geq 2)$ is less than $n\lfloor(4\varepsilon^{(k)}r)/(\varepsilon^{(k)})\rfloor + 2nr = 6nr$. ∎

The overall time complexity of the scaling algorithm is as follows. The number of executions of the approximate-weight-splitting algorithm is $\mathrm{O}(\log(W/\mu))$, and each approximate-weight-splitting algorithm has $\mathrm{O}(nr)$ iterations. We assume that for any subset $F \subseteq E$ we can find in $F$, if there is one, a cocircuit of $M_1$ in $f_1^*$ time, and a circuit of $M_2$ in $f_2$ time. We find a $\boldsymbol{u}$-minimum base of $M_1$ in $\mathrm{O}(n \log n + nf_1^*)$ time and a $\boldsymbol{v}$-minimum base of $M_2$ in $\mathrm{O}(n \log n + nf_2)$ time by the greedy algorithm. Moreover, we obtain a minimum element of a fundamental cocircuit at line 9 of the approximate-weight-splitting algorithm in $\mathrm{O}(f_1^* + (n-r))$ time and a minimum element of a fundamental circuit at line 15 in $\mathrm{O}(f_2 + r)$ time. The other steps of the approximate-weight-splitting algorithm are executed in constant time. Summarizing the above and putting $f = \max\{f_1^*, f_2\}$, we obtain the following theorem.

**Theorem 10.** *The scaling algorithm finds a minimum common base in* $\mathrm{O}((f + n)nr\log(rW))$ *time if we set* $\varepsilon_0 = W$ *and* $\mu = \dfrac{1}{2r}$.

## 5 Concluding Remarks

An approximate-weight-splitting algorithm for the minimum common base problem is proposed and the polynomial time complexity of $\mathrm{O}((f+n)nr\log(rW))$ is achieved.

Very recently, S. Fujishige and X. Zhang [9] have proposed a cost scaling algorithm for the independent assignment problem with $\mathrm{O}((f+n)n\sqrt{r}\,\log(rW))$ time. This algorithm is a generalization of the hybrid algorithm by J. B. Orlin and R. K. Ahuja [13] for the ordinary assignment problem. Their cost scaling algorithm performs a number of cost scaling phases and each phase consists of an auction-like algorithm and a successive-shortest-path algorithm. Our scaling version of the approximate-weight-splitting algorithm can also be further improved to achieve the same time complexity by hybridization with the successive-shortest-path algorithm for minimum common base problem.

**Acknowledgements**

The authors are indebted to Professor Tomomi Matsui of University of Tokyo for enjoyable discussions. We are grateful to Professor Satoru Fujishige and Mr. Xiaodong Zhang of University of Tsukuba for the relevant reference [9]. Thanks are also due to Professor Masakazu Kojima and Ms. Yoshiko Ikebe of Tokyo Institute of Technology and Professor David Avis of McGill University for comments on the manuscript.

**REFERENCES**

[1] D. P. Bertsekas, "A distributed algorithm for the assignment problem," Laboratory for Information and Decision Sciences Working Paper, M.I.T. (1979).

[2] D. P. Bertsekas, "Auction algorithms for network flow problems: A tutorial introduction," *Computational Optimization and Applications,* 1 (1992) 7–66.

[3] D. P. Bertsekas and J. Eckstein, "Dual coordinate step methods for linear network flow problems," *Mathematical Programming,* 42 (1988) 203–243.

[4] C. Brezovec, G. Cornuéjols and F. Glover, "Two algorithms for weighted matroid intersection," *Mathematical Programming,* 36 (1986) 39–53.

[5] J. Edmonds, "Submodular functions, matroids and certain polyhedra," *Combinatorial Structures and Their Applications* (R. Guy, H. Hanani, N. Sauer and J. Schönheim, eds.) Gordon and Breach, New York (1970) 69–87.

[6] J. Edmonds, "Matroid intersection," *Annals of Discrete Mathematics,* 4 (1979) 39–49.

[7] A. Frank, "A weighted matroid intersection algorithm," *Journal of Algorithms,* 2 (1981) 328–336.

[8] S. Fujishige, "A primal approach to the independent assignment problem," *Journal of the Operations Research Society of Japan,* 20 (1977) 1–15.

[9] S. Fujishige and X. Zhang, "An efficient cost scaling algorithm for the independent assignment problem," Institute of Socio-Economic Planning Discussion Paper Series No.541, University of Tsukuba (1993).

[10] M. Iri and N. Tomizawa, "An algorithm for finding an optimal "independent assignment"," *Journal of the Operations Research Society of Japan,* 19 (1976) 32–57.

[11] E. L. Lawler, "Matroid intersection algorithms," *Mathematical Programming,* 9 (1975) 31–56.

[12] T. Matsui and K. Shibata, "Auction algorithm for minimum arborescence problems," *Abstracts of the 1991 Spring National Conference of Operations Research Society of Japan* (1991) 254–255.

[13] J. B. Orlin and R. K. Ahuja, "New scaling algorithms for the assignment and minimum mean cycle problems," *Mathematical Programming,* 54 (1992) 41–56.