MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Fractional Packing in Ideal Clutters

Yuji MATSUOKA

(Communicated by Satoru IWATA)

METR 2007–04 January 2007

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.i.u-tokyo.ac.jp/edu/course/mi/index_e.shtml

Fractional Packing in Ideal Clutters *

Yuji MATSUOKA

Department of Mathematical Informatics
Graduate School of Information Science and Technology
The University of Tokyo
and
Research Institute for Mathematical Sciences
Kyoto University
yuji_matsuoka@mist.i.u-tokyo.ac.jp

Abstract

This paper presents a generic scheme for fractional packing in ideal clutters. Consider an ideal clutter with a nonnegative capacity function on its vertices. It follows from ideality that for any nonnegative capacity the total multiplicity of an optimal fractional packing is equal to the minimum capacity of a vertex cover. Our scheme finds an optimal packing using at most n edges with positive multiplicities, performing minimization for the clutter at most n times and minimization for its blocker at most n^2 times, where n denotes the cardinality of the vertex set. Applied to the clutter of dijoins (directed cut covers), the scheme provides the first combinatorial polynomial-time algorithm for fractional packing of dijoins.

1 Introduction

Consider a hypergraph $\mathcal{C} = (V, \mathcal{E})$, where V is a finite set and $\mathcal{E}$ is a family of subsets of V. An element of V is called a *vertex* of $\mathcal{C}$ and an set in $\mathcal{E}$ an *edge* of $\mathcal{C}$. Let n be the cardinality of the vertex set V. A hypergraph $\mathcal{C} = (V, \mathcal{E})$ is called a *clutter* if no two sets in $\mathcal{E}$ are contained in each other. For a clutter $\mathcal{C} = (V, \mathcal{E})$, the *blocker* of $\mathcal{C}$ is defined to be the clutter $b(\mathcal{C}) = (V, \mathcal{B})$ whose edge set $\mathcal{B}$ is the collection of all inclusionwise minimal members of $\{B \subseteq V \mid |B \cap E| \geq 1, \forall E \in \mathcal{E}\}$. The term 'clutter' was introduced by Edmonds and Fulkerson [3]. They noticed the important duality relation that $b(b(\mathcal{C})) = \mathcal{C}$ holds for any clutter $\mathcal{C}$.

*Preliminary version of this paper appeared in Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2007, pp. 1181–1186.

For a clutter $\mathcal{C} = (V, \mathcal{E})$ and a nonnegative capacity function w on V, we use $\tau_{\mathcal{C}}(w)$ to denote the minimum capacity of a vertex cover:

$$\tau_{\mathcal{C}}(w) = \min\{w(B) \mid B \in \mathcal{B}\}.$$

We can characterize $\tau_{\mathcal{C}}(w)$ by the following integer linear program:

$$\tau_{\mathcal{C}}(w) = \min\{z^\top w \mid z \in \mathbb{Z}_+^n,\ z^\top \chi_E \geq 1 \text{ for all } E \in \mathcal{E}\},$$

where χ_E denotes the characteristic vector of $E \subseteq V$ and $\mathbb{Z}_+$ denotes the set of nonnegative integers. We use $\tau^*_{\mathcal{C}}(w)$ to denote the optimal value of the relaxed linear programming. Then we have

$$\tau^*_{\mathcal{C}}(w) = \min\{z^\top w \mid z \geq \mathbf{0},\ z^\top \chi_E \geq 1 \text{ for all } E \in \mathcal{E}\}, \qquad (1)$$

where $\mathbf{0}$ denotes a vector whose components are all 0. A clutter $\mathcal{C}$ is called *ideal* if $\tau_{\mathcal{C}}(w) = \tau^*_{\mathcal{C}}(w)$ holds for any nonnegative vector w. This is equivalent to the property that the relaxed linear program (1) has an integral optimal solution for any nonnegative function w, or the integrality of the polyhedron

$$P = \{z \geq \mathbf{0} \mid z^\top \chi_E \geq 1 \text{ for all } E \in \mathcal{E}\}. \qquad (2)$$

The notion of ideality is also known as the *width-length property* [12] or the $\mathbb{Q}_+$-*max-flow min-cut property* [15] (shortly the $\mathbb{Q}_+$-*MFMC property*). Fulkerson [5, 6] showed that a clutter is ideal if and only if its blocker is ideal.

For a clutter $\mathcal{C} = (V, \mathcal{E})$ and a nonnegative capacity function w on V, a nonnegative vector y indexed by $E \in \mathcal{E}$ is called a *packing* if it satisfies the capacity constraint:

$$\sum_{E \in \mathcal{E}} \{y_E \mid v \in E\} \leq w(v),$$

for any vertex $v \in V$. The component y_E for each edge $E \in \mathcal{E}$ is called a *multiplicity* for E. A packing y is said to be *integral* when y_E is integral for each $E \in \mathcal{E}$. A packing that is not necessarily integral is often refererd to as a *fractional packing*. The *total multiplicity* of a packing y is defined by the sum of all multiplicities: $\sum\{y_E \mid E \in \mathcal{E}\}$. A *maximum packing* is a packing with the maximum total multiplicity. The problem of finding a maximum fractional packing can be described as

$$\max\{\sum_{E \in \mathcal{E}} y_E \mid \sum_{E \in \mathcal{E}} y_E \chi_E \leq w,\ y_E \geq 0 \text{ for all } E \in \mathcal{E}\}, \qquad (3)$$

which is the dual linear program of (1). Therefore, for an ideal clutter $\mathcal{C}$ and a nonnegative capacity function w, it follows from the linear programming duality theorem that the total multiplicity of a maximum fractional packing is equal to $\tau_{\mathcal{C}}(w)$ $(= \tau^*_{\mathcal{C}}(w))$. Moreover, the ideality can be characterized by the property that for any nonnegative capacity function the total multiplicity of an optimal fractional packing is equal to $\tau_{\mathcal{C}}(w)$, the minimum capacity of a vertex cover.

We say that a clutter $\mathcal{C}$ has *MFMC property* if for any nonnegative integral capacity function the total multiplicity of an optimal integral packing is equal to $\tau_{\mathcal{C}}(w)$. This is equivalent to the property that the dual linear program (3) has an integral optimal solution for any nonngative integral vector w, or the totally dual integrality of the linear system that defines P in (2). A clutter with MFMC property is also said to have the $\mathbb{Z}_+$*-max-flow min-cut property* [15], shortly the $\mathbb{Z}_+$*-MFMC property.*

In this paper, we consider the fractional packing problem for ideal clutters, which includes the maximum flow problem and the problem of finding fractional packing of r-arborescences as special cases. Since our problem can be formalized as the linear program (3), we can find an optinal solution in polynomial time with the aid of the ellipsoid method [9], which is not efficient in practice. For this problem, we present a combinatorial scheme, where we assume that two oracles are given: one returns a minimum cost edge of the clutter for a given nonnegative cost function on vertices, and the other tells a minimum capacity edge of the blocker. Our scheme finds an optimal packing using at most n edges with positive multiplicities, performing minimization for the clutter at most n times and minimization for its blocker at most n^2 times, where n denotes the cardinality of the vertex set. The scheme can be regarded as generalization of an algorithm for fractional packing of r-arborescences proposed by Gabow and Manu [8], and an algorithm for fractional packing of T-joins given by Barahona [1].

Applying the proposed scheme to the clutter of dijoins, we give the first combinatorial polynomial–time algorithm for fractional packing of dijoins. Consider a directed graph $G = (N, A)$ with a nonnegative capacity function w on arcs. A set of arcs $K \subseteq A$ is called a *dicut (directed cut)* if $K = \delta^-(U)$ for some U with $\emptyset \neq U \subsetneq N$ and $\delta^+(U) = \emptyset$, where $\delta^-(U)$ $(\delta^+(U))$ denotes arcs entering (leaving) U. A set of arcs $J \subseteq A$ is called a *dijoin (directed cut cover)* if it is an inclusionwise minimal arc set which intersects every dicut.

The theorem of Lucchesi–Younger [13] states that the clutter of dicuts has MFMC property, implying that the clutter of dijoins is ideal. Hence the capacity of a minimum dicut is equal to the total multiplicity of a maximum fractional packing of dijoins, while Schrijver [14] showed a counterexample where the capacity of a minimum dicut is strictly larger than the total multiplicity of a maximum integral packing of dijoins. By applying the algorithm for minimum cuts proposed by Hao and Orlin [10], one can find a minimum dicut in $\mathrm{O}(|A||N| \log(|N|^2/|A|))$ time. Remark that this algorithm does not yield a fractional packing of dijoins simultaneously. Efficient algorithms have been developed for minimum dijoins. The first one is due to Frank [4], and the current best complexity is $O(|A||N|^2)$ [7, 11, 17].

By applying our scheme to the clutter of dijoins, we obtain the first polynomial-time algorithm for fractional packing of dijoins, which runs in $O(|A|^3|N|\log(|N|^2/|A|))$ time. This complexity can be further improved to $O(|A|^2|N|^2 + |A||N|^3\log(|N|^2/|A|))$.

The paper is organized as follows. Section 2 provides a short review on the theory of blocking polyhedra. Section 3 presents our scheme to find fractional packing in ideal clutters. Section 4 gives an application of the proposed scheme to the clutter of dijoins, and yields a combinatorial polynomial–time algorithm for fractional packing of dijoins.

2 Preliminaries

This section provides a short review on the theory of blocking polyhedra. See [16] for more details.

For a closed convex set $P \subseteq \mathbb{R}^n_+$, we say P is of *blocking type* if $P = P + \mathbb{R}^n_+$. With a polyhedron $P \subseteq \mathbb{R}^n_+$, we associate its *blocking polyhedron*

$$B(P) = \{z \in \mathbb{R}^n_+ \,|\, z^\top x \geq 1 \text{ for all } x \in P\}.$$

Fulkerson [5, 6] showed the following important theorem.

Theorem 2.1. *Let $P \subseteq \mathbb{R}^n_+$ be a polyhedron of blocking type. Then $B(P)$ is again a polyhedron of blocking type, and $B(B(P)) = P$. Moreover, $P = \mathrm{conv}\{c_1, \ldots, c_n\} + \mathbb{R}^n_+$ if and only if $B(P) = \{z \geq \mathbf{0} \,|\, z^\top c_i \geq 1$ for $i = 1, \ldots, n\}$.*

Fulkerson [5, 6] also gave a characterization of ideal clutters in terms of blocking polyhedra. A clutter $\mathcal{C} = (V, \mathcal{E})$, as well as its blocker $b(\mathcal{C}) = (V, \mathcal{B})$, is ideal if and only if the polyhedra

$$P = \{z \geq \mathbf{0} \,|\, z^\top \chi_E \geq 1 \text{ for all } E \in \mathcal{E}\}$$

and

$$Q = \{x \geq \mathbf{0} \,|\, \chi_B^\top x \geq 1 \text{ for all } B \in \mathcal{B}\}$$

form a pair of blocking polyhedra. Therefore, for an ideal clutter $\mathcal{C}$, we have

$$Q = B(P) = \mathrm{conv}^+\{\chi_E \,|\, E \in \mathcal{E}\},$$

where $\mathrm{conv}^+\{\chi_E \,|\, E \in \mathcal{E}\}$ denotes $\mathrm{conv}\{\chi_E \,|\, E \in \mathcal{E}\} + \mathbb{R}^V_+$. Note that vectors χ_E are exactly the extreme points of Q.

Lehman [12] gave another characterization of ideal clutters by introducing the width-length inequality. A clutter $\mathcal{C} = (V, \mathcal{E})$, as well as its blocker $b(\mathcal{C}) = (V, \mathcal{B})$, is ideal if and only if

$$\min\{w(E) \,|\, E \in \mathcal{E}\} \times \min\{l(B) \,|\, B \in \mathcal{B}\} \leq l^\top w \quad (4)$$

holds for any $w \in \mathbb{R}^V_+$ and any $l \in \mathbb{R}^V_+$.

We now introduce two oracles which are used as subroutines in our fractional packing scheme. We use **Oracle MINC** (MINimization for the Clutter) to denote an oracle which returns a minimum cost edge in $\mathcal{E}$ for a given nonnegative cost vector l, and **Oracle MINB** (MINimization for the Blocker) to denote an oracle which gives a minimum capacity edge in $\mathcal{B}$ for a given nonnegative capacity vector w. Remark that **Oracle MINC** corresponds to an optimization oracle for the polyhedron Q, because for a given nonnegative cost vector l, it outputs a minimum cost edge $E \in \mathcal{E}$ whose characteristic vector χ_E is a minimizer of the minimization problem $\min\{l^\top x \mid x \in Q\}$. On the other hand, **Oracle MINB** serves as a separation oracle for the polyhedron Q. Suppose that it returns a minimum capacity edge $D \in \mathcal{B}$ for a given nonnegative capacity vector w. If its optimal value $w(D)$ is greater than or equal to 1, then w is in Q, because we have $\chi_B^\top w = w(B) \geq w(D) \geq 1$ for any $B \in \mathcal{B}$. Otherwise, $\{x \mid \chi_D^\top x \geq 1\}$ is the hyperplane which separates w and Q, because we have $\chi_D^\top x \geq 1$ for any $x \in Q$ while $\chi_D^\top w = w(D) < 1$ holds.

From the above observation, it is easy to see that the following lemma holds for an ideal clutter $\mathcal{C}$.

Lemma 2.2. *For any nonnegative vector w with $\tau_{\mathcal{C}}(w) > 0$, we have*

$$\frac{w}{\tau_{\mathcal{C}}(w)} \in Q,$$

where $\tau_{\mathcal{C}}(w) = \min\{w(B) \mid B \in \mathcal{B}\}$.

Note that $\frac{w}{\tau_{\mathcal{C}}(w)}$ is, in fact, in some facet of Q, because we have $\chi_D^\top w = \tau_{\mathcal{C}}(w)$ for a minimum capacity edge $D \in \mathcal{B}$. In the remainder we use $\tau(w)$, instead of $\tau_{\mathcal{C}}(w)$, for convenience.

3 A Fractional Packing Scheme

In this section we present a generic scheme for fractional packing in ideal clutters.

Consider an ideal clutter $\mathcal{C} = (V, \mathcal{E})$ and a nonnegative capacity function w on the vertex set V. Let $b(\mathcal{C}) = (V, \mathcal{B})$ be the blocker of the clutter $\mathcal{C}$. Since $\mathcal{C}$ is ideal, the minimum capacity $\tau(w)$ of an edge in the blocker is equal to the total multiplicity of a maximum fractional packing, that is,

$$\begin{aligned}\tau(w) &= \min\{w(B) \mid B \in \mathcal{B}\} \\ &= \max\{\sum_{E \in \mathcal{E}} y_E \mid \sum_{E \in \mathcal{E}} y_E \chi_E \leq w,\, y_E \geq 0 \text{ for all } E \in \mathcal{E}\}.\end{aligned}$$

For an edge $E \in \mathcal{E}$, we use $\beta(E)$ to denote $\min\{w(v) \mid v \in E\}$. We define the *packing capacity* $\alpha(E)$ for an edge $E \in \mathcal{E}$ as follows:

$$\alpha(E) = \max\{0 \leq \alpha \leq \beta(E) \mid \tau(w - \alpha\chi_E) = \tau(w) - \alpha\}.$$

Remark that this value may be zero for some edge.

This definition suggests the following greedy algorithm. Find an edge $E \in \mathcal{E}$ and compute $\alpha(E)$. We then solve the fractional packing problem recursively for the new weight function $w - \alpha(E)\chi_E$. By the definition of $\alpha(E)$, the total multiplicity of an optimal packing to $w - \alpha(E)\chi_E$ should be $\tau(w) - \alpha(E)$ Then assign $\alpha(E)$ to y_E. The resulting packing attains the total multiplicity $\tau(w)$.

We now discuss complexity of the algorithm. We can find an edge $E \in \mathcal{E}$ by using **Oracle MINC** once for some cost function l. We can compute $\alpha(E)$ by repeating **Oracle MINB** at most n times, as we will show in Section 3.1. However, the number of iterations is not clear at this stage. In order to bound the number of iterations, we introduce guidance on the choice of the edge E, which we will describe in Section 3.3.

3.1 Computation of the Packing Capacity

In this subsection, we discuss how to compute $\alpha(E)$ for a given edge $E \in \mathcal{E}$.

Suppose we are given an edge $E \in \mathcal{E}$ with $\beta(E) > 0$ and a capacity function w with $\tau(w) > 0$. Remark that $\tau(w - \alpha\chi_E)$ is a piecewise-linear concave function in the parameter α, since it is the minimum of a finite number of affine functions. We start by setting $\alpha = \beta(E)$. Given a value of α, we find a minimum capacity edge B in $\mathcal{B}$ with respect to $w - \alpha\chi_E$. This can be done by using **Oracle MINB** for $w - \alpha\chi_E$. If the optimal value $\tau(w - \alpha\chi_E)$ is equal to $\tau(w) - \alpha$, then $\alpha(E)$ is equal to the current value of α. Otherwise, we apply the following lemma.

Lemma 3.1. *If $\tau(w - \alpha\chi_E) < \tau(w) - \alpha$ for an edge $E \in \mathcal{E}$ and a positive value α, then $|B \cap E| > 1$ holds for any minimum capacity edge $B \in \mathcal{B}$ with respect to $w - \alpha\chi_E$.*

Proof. Suppose that $|B \cap E| = 1$ holds for some minimum capacity edge B with respect to $w - \alpha\chi_E$. Then we have $\tau(w - \alpha\chi_E) = \chi_B^\top(w - \alpha\chi_E) = w(B) - \alpha$. This implies that $\tau(w) \leq w(B) = \tau(w - \alpha\chi_E) + \alpha$, which contradicts the assumption. □

Remark that we have $\tau(w) - \alpha(E) = \tau(w - \alpha(E)\chi_E) \leq (w - \alpha(E)\chi_E)(B) = w(B) - \alpha(E)|B \cap E|$. Since $|B \cap E| > 1$ by Lemma 3.1, this implies that $\alpha(E)$ is at most $\dfrac{w(B) - \tau(w)}{|B \cap E| - 1}$. We replace α by this value, and repeat this procedure. The procedure is summarized as follows:

Computation of $\alpha(E)$

Input: An edge $E \in \mathcal{E}$ with $\beta(E) > 0$ and a capacity function w with $\tau(w) > 0$.

Output: A packing capacity $\alpha(E)$ and, if $\alpha(E) < \beta(E)$, a minimum capacity edge $D \in \mathcal{B}$ with respect to $w - \alpha(E)\chi_E$ satisfying $|D \cap E| > 1$.

Step 0: Compute $\beta(E)$. Set $\alpha \leftarrow \beta(E)$ and $D \leftarrow \emptyset$.

Step 1: Use **Oracle MINB** for $w - \alpha\chi_E$ to obtain a minimizer $B \in \mathcal{B}$.

Step 2: If $(w - \alpha\chi_E)(B) < \tau(w) - \alpha$, then replace D by B and α by $\dfrac{w(B) - \tau(w)}{|B \cap E| - 1}$. Go back to Step 1.

Step 3: Return α. If $\alpha < \beta(E)$, then return D.

Since at each iteration the value $|B \cap E|$ decreases by at least one, the procedure requires at most n $(= |V|)$ iterations. Thus we have the following lemma:

Lemma 3.2. *If $\alpha(E) = \beta(E)$, the procedure requires one computation of* **Oracle MINB**. *Otherwise, it requires at most n computations of* **Oracle MINB**.

3.2 Polyhedral Characterization of the Packing Capacity

In this subsection, we introduce the blocking polyhedron Q and characterize the case where the packing capacity $\alpha(E)$ is equal to $\tau(w)$.

We use $\tilde{w}(\lambda)$ to denote the externally dividing point of $\frac{w}{\tau(w)}$ and χ_E defined by

$$\tilde{w}(\lambda) = \frac{\frac{w}{\tau(w)} - \lambda\chi_E}{1 - \lambda}$$

for each λ with $0 \leq \lambda < 1$. Then the packing capacity $\alpha(E)$ is characterized as follows.

Lemma 3.3. *For any edge $E \in \mathcal{E}$ with $\beta(E) > 0$ and any capacity function w with $\tau(w) > 0$, we have*

$$\frac{\alpha(E)}{\tau(w)} = \sup\{\lambda \mid 0 \leq \lambda < 1,\ \tilde{w}(\lambda) \in Q\}.$$

Proof. This follows from the following equations:

$$\begin{aligned}\alpha(E) &= \max\{\alpha \,|\, 0 \le \alpha \le \beta(E),\ \tau(w - \alpha\chi_E) = \tau(w) - \alpha\} \\ &= \max\{\alpha \,|\, 0 \le \alpha \le \tau(w),\ w - \alpha\chi_E \ge \mathbf{0},\ \tau(w - \alpha\chi_E) = \tau(w) - \alpha\} \\ &= \max\{\alpha \,|\, 0 \le \alpha \le \tau(w),\ w - \alpha\chi_E \ge \mathbf{0},\ \chi_B^\top(w - \alpha\chi_E) \ge \tau(w) - \alpha \\ &\qquad \text{for all } B \in \mathcal{B}\} \\ &= \sup\{\alpha \,|\, 0 \le \alpha < \tau(w),\ \frac{w - \alpha\chi_E}{\tau(w) - \alpha} \in Q\} \\ &= \tau(w) \times \sup\{\lambda \,|\, 0 \le \lambda < 1,\ \tilde{w}(\lambda) \in Q\},\end{aligned}$$

where in the last equality we let $\lambda = \frac{\alpha}{\tau(w)}$. □

Consider the case where $\alpha(E) = \tau(w)$ holds. In this case, we can finish the algorithm by assigning $\alpha(E)$ to y_E. The following corollary gives the condition when this case occurs.

Corollary 3.4. *For any edge $E \in \mathcal{E}$ with $\beta(E) > 0$ and any capacity function w with $\tau(w) > 0$, we have $\alpha(E) = \tau(w)$ if and only if $\chi_E \le \frac{w}{\tau(w)}$ holds.*

Proof. Suppose that we have $\alpha(E) = \tau(w)$, then we have $\frac{w}{\tau(w)} - \lambda\chi_E \ge 0$ for any λ in $0 \le \lambda < 1$ by using Lemma 3.3. This implies $\chi_E \le \frac{w}{\tau(w)}$.

Conversely, if $\chi_E \le \frac{w}{\tau(w)}$, then $\chi_E \le \frac{w}{\tau(w)} \le \tilde{w}(\lambda)$ holds for any λ in $0 \le \lambda < 1$, which implies $\alpha(E) = \tau(w)$. □

3.3 A Packing Algorithm

This subsection provides guidance on the choice of edges, and presents our scheme for fractional packing.

For a vertex set $W \subseteq V$ and an edge set $\mathcal{D} \subseteq \mathcal{B}$, we define a face $F(W, \mathcal{D})$ of Q as

$$F(W, \mathcal{D}) = \{x \in Q \,|\, x(v) = 0, \text{ for all } v \in W \text{ and } \chi_D^\top x = 1, \text{ for all } D \in \mathcal{D}\}.$$

Remark that $F(W, \mathcal{D})$ may be empty for some $W \subseteq V$ and $\mathcal{D} \subseteq \mathcal{B}$. In the algorithm we keep $F(W, \mathcal{D})$ so that $\frac{w}{\tau(w)} \in F(W, \mathcal{D})$, and choose a vector χ_E from $F(W, \mathcal{D})$.

We now describe our algorithm. We start the algorithm by initializing variables: $\mathcal{D} \leftarrow \emptyset$ and $W \leftarrow \{v \in V \,|\, w(v) = 0\}$. First, find a minimum cost edge $E \in \mathcal{E}$ for the cost function $l = \chi_W + \sum_{D \in \mathcal{D}} \chi_D$. This can be done by using **Oracle MINC** for l. Next, we compute $\alpha(E)$ for E and assign $\alpha(E)$ to the multiplicity y_E. If $\alpha(E) < \beta(E)$, then the procedure of computing $\alpha(E)$ returns a minimum capacity edge $D \in \mathcal{B}$ with respect to $w - \alpha(E)\chi_E$ satisfying $|D \cap E| > 1$. Finally, replace w by $w - \alpha(E)\chi_E$ and update $F(W, \mathcal{D})$. If $\alpha(E) = \beta(E)$, then replace W by $\{v \in V \mid w(v) = 0\}$. Otherwise, add D to $\mathcal{D}$. Repeat this procedure until $\tau(w)$ is equal to zero.

The algorithm is summarized as follows.

Fractional Packing in Ideal Clutters

Input: A nonnegative capacity function w.
Output: A maximum fractional packing y.
Assumption: Oracle MINC and **Oracle MINB** are given.

Step 0: (Initialization)
Set $\mathcal{D} \leftarrow \emptyset$ and $W \leftarrow \{v \in V \mid w(v) = 0\}$.
Step 1: (Finding a vertex χ_E in a face $F(W, \mathcal{D})$)
Use **Oracle MINC** for $l = \chi_W + \sum_{D \in \mathcal{D}} \chi_D$ to obtain a minimum cost edge $E \in \mathcal{E}$.
Step 2: (Computation of $\alpha(E)$)
Compute $\alpha(E)$ and assign $\alpha(E)$ to y_E.
If $\alpha(E) < \beta(E)$, we obtain a minimum capacity edge $D \in \mathcal{D}$ with respect to $w - \alpha(E)\chi_E$ satisfying $|D \cap E| > 1$.
Step 3: (Updating w and $F(W, \mathcal{D})$)
Replace w by $w - \alpha(E)\chi_E$. If $\alpha(E) = \beta(E)$, then replace W by $\{v \in V \mid w(v) = 0\}$. Otherwise, add D to $\mathcal{D}$.
Step 4: If $\tau(w) = 0$, then return y. Otherwise, go back to Step 1.

To prove the validity of the algorithm, we show the following two lemmas:

Lemma 3.5. *At any point of the algorithm, we have $w(D) = \tau(w)$ for any $D \in \mathcal{D}$.*

Proof. Suppose that before Step 3 we have $w(D) = \tau(w)$ for any $D \in \mathcal{D}$.

From the definition of τ, we have for any $B \in \mathcal{B}$,

$$\begin{aligned}\tau(w) - \alpha(E) = \tau(w - \alpha(E)\chi_E) &\leq (w - \alpha(E)\chi_E)(B) \\ &= w(B) - \alpha(E)|B \cap E| \leq w(B) - \alpha(E),\end{aligned}$$

where the last inequality follows from $|B \cap E| \geq 1$. Then for any $D \in \mathcal{D}$, it holds that $\tau(w - \alpha(E)\chi_E) = (w - \alpha(E)\chi_E)(D)$, because we have $w(D) = \tau(w)$.

A new edge $D \in \mathcal{B}$ added to $\mathcal{D}$ in Step 3 is a minimum capacity edge for the replaced w, which implies $w(D) = \tau(w)$.

Thus $w(D) = \tau(w)$ holds for any $D \in \mathcal{D}$ after execution of Step 3. $\square$

From this lemma we have $\frac{w}{\tau(w)} \in F(W, \mathcal{D})$ throughout the algorithm, which implies that $F(W, \mathcal{D})$ is not empty.

Lemma 3.6. *For any $W \subseteq V$ and any $\mathcal{D} \subseteq \mathcal{B}$, let $E \in \mathcal{E}$ be a minimum cost edge with respect to $l = \chi_W + \sum_{D \in \mathcal{D}} \chi_D$. If $F(W, \mathcal{D})$ is not empty, then χ_E is in the face $F(W, \mathcal{D})$.*

Proof. Since $|D \cap E| \geq 1$ for all $D \in \mathcal{D} \subseteq \mathcal{B}$, we have

$$l(E) = |W \cap E| + \sum_{D \in \mathcal{D}} |D \cap E| \geq \sum_{D \in \mathcal{D}} |D \cap E| \geq |\mathcal{D}|.$$

On the other hand, by using Lehman's width-length inequality (4) for a minimum cost edge $E \in \mathcal{E}$, we have

$$\tau(w) \times l(E) \leq l^\top w = w(W) + \sum_{D \in \mathcal{D}} w(D) = \sum_{D \in \mathcal{D}} w(D) = \tau(w) \times |\mathcal{D}|,$$

where the last equality follows from Lemma 3.5. Since we have $\tau(w) > 0$ in the algorithm, these inequalities implies that $l(E) = |\mathcal{D}|$. Therefore, $W \cap E = \emptyset$ and $|D \cap E| = 1$ for all $D \in \mathcal{D}$, which imply $\chi_E \in F(W, \mathcal{D})$. $\square$

Remark that $\chi_E \in F(W, \mathcal{D})$ implies that $\beta(E) > 0$ holds, because $\chi_E(v) = 0$ for all $v \in W$ and $w(v) > 0$ for all $v \in E$. This lemma also follows from the observation that $-l$ is in the relative interior of the normal cone of the face $F(W, \mathcal{D})$. Hence in the scheme we may choose l to be $\sum_{v \in W} \gamma_v \chi_v + \sum_{D \in \mathcal{D}} \gamma_D \chi_D$ for arbitrary positive values γ_v and γ_D. Note that χ_E is, in fact, an extreme point of $F(W, \mathcal{D})$, but we do not use this property in our algorithm.

For complexity of the algorithm, we have the following lemma.

Lemma 3.7. *The dimension of $F(W, \mathcal{D})$ decreases by at least one in each iteration.*

Proof. By Lemma 3.6, we have $\chi_E \in F(W, \mathcal{D})$ before updating $F(W, \mathcal{D})$. If $\alpha(E) = \beta(E)$, there exists at least one vertex $v \in V$ with $w(v) > 0$ such that packing of E makes $w(v)$ zero. Hence we have $\chi_E \notin F(W, \mathcal{D})$ after enlarging W. Otherwise ($\alpha(E) < \beta(E)$), we have a minimum capacity edge D in $\mathcal{B}$ satisfying $|D \cap E| > 1$. After adding D to $\mathcal{D}$, we have $\chi_E \notin F(W, \mathcal{D})$. $\square$

Consider the case where the dimension of $F(W, \mathcal{D})$ is equal to 0. Then we can describe $F(W, \mathcal{D})$ as $\{\chi_E\}$ for some edge $E \in \mathcal{E}$. Therefore, in Step 1 we obtain E as the unique minimizer of **Oracle MINC**, and in Step 2 we have $\alpha(E) = \tau(w)$ from Corollary 3.4, which leads to the termination of the algorithm.

Lemma 3.7 implies that the number of iterations is at most n, which together with Lemma 3.2 gives the following theorem on the running time of our algorithm.

Theorem 3.8. *For an ideal clutter $\mathcal{C}$, our scheme finds an optimal fractional packing of edges in $\mathcal{E}$, performing at most n computations of* **Oracle MINC** *and at most n^2 computations of* **Oracle MINB**.

As a consequence, we have the following bound on the number of edges with positive multiplicities.

Corollary 3.9. *For an ideal clutter $\mathcal{C} = (V, \mathcal{E})$ with $|V| = n$, there exists an optimal fractional packing of at most n edges with positive multiplicities.*

4 Application — Fractional Packing of Dijoins

Consider a directed graph $G = (N, A)$ with a nonnegative capacity function w on arcs.

The clutter of dijoins is ideal, because the theorem of Lucchesi–Younger [13] states that the clutter of dicuts has MFMC property. We can compute a minimum capacity dicut in time $\mathrm{O}(|A||N| \log(|N|^2/|A|))$. On the other hand, we can find a minimum cost dijoin in time $\mathrm{O}(|A||N|^2)$. Remark that most algorithms for minimum cost dijoins do not necessarily return minimal arc sets as optimal solutions, and we can compute a minimal arc set without violating optimality by removing some arcs. This can be done in time $\mathrm{O}(|A||N|)$.

Therefore, by applying our scheme to the clutter of dijoins, we have a combinatorial polynomial-time algorithm for fractional packing of dijoins, which runs in time $\mathrm{O}(|A|^3|N| \log(|N|^2/|A|))$. The obtained fractional packing uses at most $|A|$ distinct dijoins.

We now improve the complexity of the algorithm by giving better bound on the number of oracle calls to find minimum capacity dicuts. Consider the case where $\alpha(E) = \beta(E)$ holds. This case occurs at most $|N|$ times, and each computation of $\alpha(E)$ requires one call. Consider the other case. We can compute $\alpha(E)$ by at most $|N|$ calls, because we have $|E| \leq |N| - 1$ for any dijoin E. Furthermore, this case occurs at most $4|N| - 2$ times, because we can represent minimum capacity dicuts in $\mathcal{D}$ as a cross-free family and this implies $|\mathcal{D}| \leq 4|N| - 2$.

Thus, the number of oracle calls to compute minimum capacity dicuts is $\mathrm{O}(|N|^2)$. On the other hand, the number of oracle calls to compute minimum cost dijoins is $\mathrm{O}(|A|)$. The total running time is as follows.

Theorem 4.1. *Our algorithm applied to the clutter of dijoins runs in* $\mathrm{O}(|A|^2|N|^2 + |A||N|^3 \log(|N|^2/|A|))$ *time.*

5 Conclusions

This paper has presented a generic scheme for fractional packing in ideal clutters. Our scheme finds an optimal packing of at most n edges with positive multiplicities, performing minimization for the clutter at most n times and minimization for the blocker at most n^2 times, where n denotes the cardinality of the vertex set. Applied to the clutter of dijoins (directed cut covers), the scheme yields the first combinatorial polynomial-time algorithm for fractional packing of dijoins.

Acknowledgement

I am very grateful to Satoru Iwata for valuable discussions and useful comments throughout my research.

References

[1] F. Barahona: Fractional packing of T-joins. *SIAM Journal on Discrete Mathematics*, 17, 2004, pp. 661–669.

[2] G. Cornuéjols: *Combinatorial Optimization — Packing and Covering*. SIAM, Philadelphia, Pennsylvania, 2001.

[3] J. Edmonds and D. R. Fulkerson: Bottleneck extrema. *Journal of Combinatorial Theory*, 8, 1970, pp. 299–306.

[4] A. Frank: How to make a digraph strongly connected. *Combinatorica*, 1, 1981, pp. 145–153.

[5] D. R. Fulkerson: Blocking polyhedra. In *Graph Theory and Its Applications* (Proceedings Advanced Seminar Madison, Wisconsin, 1969; B. Harris, ed.), Academic Press, New York, 1970, pp. 93–112.

[6] D. R. Fulkerson: Blocking and anti-blocking pairs polyhedra. *Mathematical Programming*, 1, 1971, pp. 168–194.

[7] H. N. Gabow: Centroids, representations, and submodular flows. *Journal of Algorithms*, 18, 1995, pp.586–628.

[8] H. N. Gabow and K. S. Manu: Packing algorithms for arborescences (and spaninng trees) in capacitated graphs. *Mathematical Programming*, 82, 1998, pp. 83–109.

[9] M. Grötschel, L. Lovász, and A. Schrijver: *Geometric Algorithms and Combinatorial Optimization*. Springer, Berlin, 1988.

[10] J. Hao and J. B. Orlin: A faster algorithm for finding the minimum cut in a directed graph. *Journal of Algorithms*, 17, 1994, pp. 424–446.

[11] S. Iwata and Y. Kobayashi: An algorithm for minimum cost arc-connectivity orientations. METR 2005-16, University of Tokyo, 2005.

[12] A. Lehman: On the width-length inequaity. *Mathematical Programming*, 17, 1979, pp. 403–417.

[13] C. L. Lucchesi and D. H. Younger: A minimax theorem for directed graphs. *The Journal of the London Mathematical Society* (2), 17, 1978, pp. 369–374.

[14] A. Schrijver: A counterexample to a conjecture of Edmonds and Giles. *Discrete Mathematics*, 32, 1980, pp. 213–214.

[15] A. Schrijver: Min–max results in combinatorial optimization. In *Mathematical Programming — The State of the Art* (Bonn, 1982; A. Bachem, M. Gtötschel, B. Korte, eds.), Springer, Berlin, 1983, pp. 439–500.

[16] A. Schrijver: *Combinatorial Optimization — Polyhedra and Efficiency.* Springer, Berlin, 2003.

[17] F. B. Shepherd and A. Vetta: Visualizing, finding and packing dijoins. In *Graph Theory and Combinatorial Optimization GERAD 25th Anniversary Volumes*, Springer, 2003, pp. 1–36.