A Fractional Packing Algorithm for Ideal Clutters

46226 Yuji MATSUOKA
Adviser: Satoru Iwata

February 8, 2006

1 Introduction

Many combinatorial optimization problems can be formulated as integer linear programming problems, and we have naturally linear programming relaxation problems for them. In general the optimal value of the relaxation problem is not equal to that of the original problem, and it is important to discuss the case where the equality holds. In clutter theory this property is characterized as idealness of clutters.

First we consider the clutter of dijoins. It follows from idealness of the clutter of dijoins that the capacity of a minimum dicut is equal to the total multiplicity of a maximum fractional packing of dijoins, while Schrijver[4] showed that the capacity of a minimum dicut is not equal to the total multiplicity of a maximum integral packing of dijoins. By applying the algorithm for a minimum cut proposed by Hao and Orlin[3], one can find a minimum dicut efficiently, but this algorithm does not yield a maximum fractional packing of dijoins. For this problem we give a combinatorial polynomial-time algorithm which runs in time $\mathrm{O}(\max\{m^2n^2, mn^3\log(n^2/m)\})$, where n and m denote the numbers of vertices and arcs, respectively.

Next we discuss the maximum fractional packing problem for an ideal clutter, which is a generalization of fractional packing of dijoins. We have proposed a combinatorial polynomial-time framework to find an optimal fractional packing. Letting n be the cardinality of the vertex set of a given clutter, our framework finds an optimal packing with at most n edges, performing at most n times minimizations for a given clutter and at most n^2 times minimizations for its blocker.

2 Fractional Packing of Dijions

2.1 Definitions

Consider a directed graph $D = (V, A)$ with a nonnegative integral capacity function w on arcs, and denote the cardinality of the vertex set V by n, that of the arc set A by m. A set of arcs $B \subseteq A$ is called a *dicut* (or a *directed cut*) if $B = \delta^-(U)$ for some U with $\emptyset \neq U \subsetneq V$ and $\delta^+(U) = \emptyset$, where $\delta^-(U)$ ($\delta^+(U)$) denotes arcs entering (leaving) U. A set of arcs $B \subseteq A$ is called a *dijoin* (or a *directed cut cover*) if it is an inclusionwise minimal arc set which intersects every dicut. A *packing of dijoins* is a family $\mathcal{B}$ of dijoins B, each with a multiplicity $\gamma(B) \in \mathbf{R}_+$ such that for any arc $a \in A$, $\sum\{\gamma(B) \mid a \in B \in \mathcal{B}\} \leq w(a)$. The *total multiplicity* of $\mathcal{B}$ is defined by $\sum\{\gamma(B) \mid B \in \mathcal{B}\}$. An *integral packing* has all multiplicities $\gamma(B)$ integral, and a *fractional packing* has $\gamma(B)$ rational. A *maximum packing* is a packing with the maximum total multiplicity.

2.2 A Packing Algorithm

We say that for $X, Y \subseteq V$, X and Y *cross* if $X \setminus Y$, $Y \setminus X$, $X \cap Y$ and $V \setminus (X \cup Y)$ are all nonempty. A family of subsets such that no two of them cross is called *cross-free*. We have the following lemma for the cross-free family:

Lemma 2.1. *If $\Phi \subseteq 2^V$ and Φ is cross-free then $|\Phi| \leq 4n - 4$.*

The main idea of the algorithm is to keep vertex sets of minimum dicuts found in the algorithm as a cross-free family Φ, which guides selection of dijoins. Outline of the algorithm is as follows:

Fractional Packing of Dijoins

Step 0: $\Phi \leftarrow \emptyset$.
Step 1: Find a dijoin B with $|B \cap \delta^-(U)| = 1$ for all $U \in \Phi$.
Step 2: Compute $\alpha(B)$, and $D \leftarrow D - \alpha(B)B$.
Step 3: If $\alpha(B) < \beta(B)$, we have a minimum dicut $\delta^-(S)$ with $|B \cap \delta^-(S)| > 1$, and perform Procedure UNCROSS(Φ, B, S).
Step 4 : If $\lambda(D) = 0$ then stop. Otherwise go back to Step 1.

Here we omit how to compute $\alpha(B)$ for a given dijoin B, the detail of Procedure UNCROSS(Φ, B, S), and proof for validity of the algorithm. For complexity of the algorithm, we have the following theorem:

Theorem 2.2. *Our fractional packing algorithm runs in* $\mathrm{O}(\max\{m^2n^2, mn^3\log(n^2/m)\})$ *time.*

Our algorithm has many similarities to an algorithm for fractional packing of arborescences proposed by Gabow and Manu[2], or an algorithm for T-joins given by Barahona[1].

3 Fractional Packing for Ideal Clutters

3.1 Definitions

A hypergraph $\mathcal{C} = (V, \mathcal{E})$ is called a *clutter* if no two sets in $\mathcal{E}$ are contained in each other. Let $\mathcal{C} = (V, \mathcal{E})$ be a clutter and w be a nonnegative vector on V, then consider the pair of linear programming as

$$\text{primal problem} : \min\{wx \mid x \geq 0,\ M(\mathcal{C})x \geq \mathbf{1}\},$$

$$\text{dual problem} : \max\{y\mathbf{1} \mid y \geq 0,\ yM(\mathcal{C}) \leq w\},$$

where $M(\mathcal{C})$ is the matrix whose row vectors are the characteristic vectors of edges. A clutter $\mathcal{C}$ is *ideal* if the primal problem has an integral optimal solution for each nonnegative vector w. The blocker of $\mathcal{C}$ is defined by the clutter $b(\mathcal{C}) = (V, \mathcal{F})$, where $\mathcal{F}$ is the collection of all inclusionwise minimal members of $\{F \subseteq V \mid |F \cap E| \geq 1, \forall E \in \mathcal{E}\}$.

3.2 Packing Problems for Clutters

Consider an ideal clutter $\mathcal{C} = (V, E(\mathcal{C}))$ and a nonnegative integral capacities w on vertices. We denote the cardinality of the vertex set V by n.

The *minimum edge problem for a clutter* $b(\mathcal{C})$ is defined as $\min\{w(C) \mid C \in E(b(\mathcal{C}))\}$. Since $E(b(\mathcal{C}))$ is an inclusionwise minimal members of $\{C \subseteq V \mid |C \cap B| \geq 1 \text{ for all } B \in E(\mathcal{C})\}$, this problem can be formulated as $\min\{wx \mid M(\mathcal{C})x \geq \mathbf{1}, x \in \mathbf{Z}^n_+\}$. Since a clutter $\mathcal{C}$ is ideal, this problem is equivalent to $\min\{wx \mid M(\mathcal{C})x \geq \mathbf{1}, x \geq \mathbf{0}\}$. Its dual problem is $\max\{y\mathbf{1} \mid yM(\mathcal{C}) \leq w, y \geq \mathbf{0}^T\}$, and this corresponds to the *maximum fractional packing problem for* $\mathcal{C}$.

3.3 A Packing Algorithm

Idealness of $\mathcal{C}$ implies that the total multiplicity of a maximum fractional packing of edges in $E(\mathcal{C})$ is equal to the capacity of a minimum edge in $E(b(\mathcal{C}))$. We denote this value by $\lambda(w)$. For an edge set $\Phi \subseteq E(b(\mathcal{C}))$, we define a face $F(\Phi)$ of Q as $Q \cap \{y\chi_B = 1 \text{ for all } B \in \Phi\} \cap \{y(v) = 0 \text{ for all } v \text{ with } w(v) = 0\}$.

The framework is as follows:

Fractional Packing for Ideal Clutters

Step 0: $\Phi \leftarrow \emptyset$.
Step 1: Find an edge $B \in E(\mathcal{C})$ such that χ_B is a vertex of $F(\Phi)$.
Step 2: Compute $\alpha(B)$, and $w \leftarrow w - \alpha(B)B$. If $\alpha(B) < \beta(B)$, then we have a minimum edge $S \in E(b(\mathcal{C}))$ with $|S \cap B| > 1$ and add S to Φ.
Step 3: If $\lambda(w) = 0$ then stop. Otherwise go back to Step 1.

How to find a vertex of $F(\Phi)$

Define a new cost $c(v)$ for each $v \in V$ as follows:

$$c(v) = \begin{cases} +\infty & w(v) = 0 \\ |\{C \in \Phi \mid v \in C\}| & w(v) > 0 \end{cases} \tag{1}$$

Find a minimum edge in $E(\mathcal{C})$ with respect to a new cost function c.

Computation of $\alpha(B)$

Step 0: $\alpha \leftarrow \beta(B)$ and $S \leftarrow \emptyset$.
Step 1: While($\lambda(w - \alpha\chi_B) < \lambda(D) - \alpha$) do:
$S \leftarrow$ a minimum edge for $w - \alpha\chi_B$.
$\alpha \leftarrow w(S) - \lambda(w)/|B \cap S| - 1$;
Step 2: Return α and S.

To examine the number of iterations in the algorithm, we have the following lemma:

Lemma 3.1. *The dimension of $F(\Phi)$ decreases by at least one at each iteration.*

The running time of the algorithm is as follows:

Theorem 3.2. *For an ideal clutter $\mathcal{C}$, our framework finds an optimal fractional packing of edges in $E(\mathcal{C})$, performing at most n minimizations for $\mathcal{C}$ and at most n^2 minimizations for $b(\mathcal{C})$.*

As a consequence of the algorithm, we have the following theorem:

Theorem 3.3. *For an ideal clutter on a finite set V, there exists an optimal fractional packing with at most $|V|$ edges.*

4 Conclusion

We have presented a combinatorial polynomial-time algorithm for fractinal packing of dijoins, which runs in time $O(\max\{m^2n^2, mn^3 \log(n^2/m)\})$. Next we deal with the maximum fractional packing problem for an ideal clutter, which is a natural generalization of fractional packing of dijoins. We have proposed a combinatorial polynomial-time framework to find an optimal fractional packing for an ideal clutter.

References

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

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

[3] 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.

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