Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Min-max partitioning problem with matroid constraint*

Biao WU†, En-yu YAO

(*Department of Mathematics, Zhejiang University, Hangzhou 310027, China*)

†E-mail: wubiao@zju.edu.cn

Received Dec. 8, 2007; revision accepted May 30, 2008

Abstract: In this paper, we consider the set partitioning problem with matroid constraint, which is a generation of the *k*-partitioning problem. The objective is to minimize the weight of the heaviest subset. We present an approximation algorithm, which consists of two sub-algorithms—the modified Edmonds' matroid partitioning algorithm and the exchange algorithm, for the problem. An estimation of the worst ratio for the algorithm is given.

Key words: Matroid, Matroid partition, Worst ratio
doi:10.1631/jzus.A071606 **Document code:** A **CLC number:** O221.7

INTRODUCTION

The partitioning problem has been studied by several researchers (Edmonds, 1965; Hwang, 1981; Hwang *et al*., 1985; Burkard and Yao, 1990). Generally, this problem can be described as follows. Let E be a finite set and $P=\{I_1, I_2, ..., I_m\}$ be a partition of E, that is, the subsets $I_1, I_2, ..., I_m$ are nonempty, pairwise disjoint and their union equals E. Now, let F be a real-valued function defined on the partitions of E. We ask for a partition P^* which minimizes the objective function $F(P)$: $F(P^*)=\min\{F(P): P \text{ is a partition of } E\}$.

Although the integer m may be fixed or unfixed, we assume m be fixed in this paper. In order to guarantee subsets $I_1, I_2, ..., I_m$ are nonempty, we always assume $m\leq|E|$.

Usually an objective function is defined in two ways: one is called a 'sum type' and defined as $F(P)=f(I_1)+f(I_2)+...+f(I_m)$, where f is a real-valued function defined on the subsets of E (Burkard and Yao, 1990; Lee and Liman, 1993); the other is called a 'bottleneck type' and defined as

$$F(P)=\max\{w(I_k): k=1, 2, ..., m\}, \quad (1)$$

where w is a positive function defined on E and $w(I)=\sum_{e\in I}w(e)$ for any subset I of E (Graham, 1969; Babel *et al*., 1998). We call $w(e)$ the weight of element e and $w(I)$ the weight of subset I.

If subsets of a partition P are required to be independent with respect to a given matroid (Welsh, 1976), P is called a 'matroid partition' and the corresponding problem is called a 'partitioning problem with matroid constraint'.

Edmonds (1965) and Burkard and Yao (1990) considered partitioning problems with matroid constraint. With the objective that the constructed partition $P=\{I_1, I_2, ..., I_m\}$ is lexicographically maximum, i.e., $|I_1|$ is maximum, $|I_2|$ is as large as possible subject to $|I_1|$ being maximum, and so on, Edmonds (1965) produced a polynomial algorithm. With a certain sum-type objective function, Burkard and Yao (1990) gave us a Greedy-like algorithm by which the optimal partition can be produced under certain order properties. So the problems discussed in (Edmonds, 1965; Burkard and Yao, 1990) are polynomially solvable.

In this paper we also consider a partitioning problem with matroid constraint, but the objective is the bottleneck type, which is different from those discussed in (Edmonds, 1965; Burkard and Yao, 1990). We call it the 'min-max partitioning problem with matriod constraint', simply denoted as MMP.

* Project (No. 10671177) supported by the National Natural Science Foundation of China

The problem is NP-hard (Garey and Johnson, 1978). We will give an algorithm to compute the approximate solution to this problem.

In Section 2 we describe the problem and present our algorithm denoted as 'Algorithm A', which consists of two sub-algorithms: one is the modified Edmonds' matroid partitioning algorithm (MEPA for short), which produces a feasible partition; the other is the exchange algorithm (EXA for short), which improves the objective value by exchanging elements among subsets of the partition produced by MEPA. Then in Section 3 we give an estimation of the worst ratio for Algorithm A.

PROBLEM DESCRIPTION AND ALGORITHMS

Let E be a finite set and $M=(E, \mathcal{P})$ be a given matroid defined on the set E, where $\mathcal{P}$ is the collection of all the independent sets. We always denote the rank and closure of a subset I by $r(I)$ and $\sigma(I)$, respectively. Let w: $E\rightarrow R$ be a positive weight function, where R is the set of positive real numbers.

Now we describe the MMP as follows.

Given a finite set E, a matroid $M=(E, \mathcal{P})$, a positive weight function w defined on E, an integer $m\geq 2$ and an objective function F defined as Eq.(1), find an optimal feasible partition P^* such that $F(P^*)=\min\{F(P): P \text{ is a feasible partition}\}$.

When we take $M=(E, \mathcal{P})$ with $\mathcal{P}=2^E$, MMP is just the partitioning problem without any constraint. The famous scheduling problem $P||C_{\min}$ (Graham, 1969) can be reduced to this problem.

When we take $M=(E, \mathcal{P})$ with $\mathcal{P}=\{I\subseteq E: |I|\leq k\}$ for a given integer k, MMP is just the k-partitioning problem. So, MMP is a generation of the k-partitioning problem (Babel *et al*., 1998).

Therefore, MMP is NP-hard.

With the same method as that in (Edmonds, 1965), it can be proved that MMP has feasible solution if and only if the inequality

$$|S|\leq m\cdot r(S) \tag{2}$$

holds for every $S\subseteq E$, where $r(S)$ is the rank of S.

Edmonds (1965) wrote an algorithm for constructing a matroid partition. Edmonds' partitioning algorithm (EPA) can be found in (Lawler, 1976). Because EPA does not take account of the weights of elements, it is not suitable for our problem. Therefore, we modify EPA such that each element is put with the subset whose weight is the lightest possible among I_1, I_2, ..., I_m.

The following is the sub-algorithm MEPA:

Step 0: Set $U=E$, $I_i=\varnothing$, $i=1, 2, ..., m$.

Step 1 (Construction of sequence S_0, S_1, ...)

Step 1.0: Set $S_0=E$ and $j=1$.

Step 1.1: Find the index k such that $w(I_k)=\min\{w(I_i): |I_i\cap S_{j-1}|<r(S_{j-1}), 1\leq i\leq m\}$. If there is no such subset I_k, stop (E is not partitionable).

Step 1.2: Set $S_j=S_{j-1}\cap\sigma(I_k\cap S_{j-1})$, $l(j)=k$.

Step 1.3: If $U\subseteq S_j$, set $j=j+1$, go to Step 1.1; otherwise, choose $e\in U\backslash S_j$ such that $w(e)=\max\{w(e): e\in U\backslash S_j\}$ and go to Step 2.

Step 2 (Augmentation of partition)

Step 2.0: Set $U=U-e$.

Step 2.1: If $I_{l(j)}+e$ is independent, set $I_{l(j)}=I_{l(j)}+e$ and go to Step 2.3.

Step 2.2: Find the unique circuit $C(e)\subseteq I_{l(j)}+e$, and choose $e'\in C(e)\backslash S_{j-1}$ such that e' is the lightest element in $C(e)\backslash S_{j-1}$; set $I_{l(j)}=I_{l(j)}+e-e'$ and $j=j-1$, $e=e'$, go to Step 2.1.

Step 2.3: If $U\neq\varnothing$, go to Step 1; otherwise, we have a partition of the given set E.

In the above algorithm, I_1, I_2, ..., I_m are subsets of the constructed partition P. The set U stores the elements which are not yet assigned to the partition currently. $l(j)$ is the label to indicate that $I_{l(j)}$ is the subset to accept the element removed from the subset $I_{l(j+1)}$ in Step 2.

The main difference between EPA and MEPA lies in the following two aspects:

(1) The newly selected element in MEPA is added to the set whose weight is the lightest possible among I_1, I_2, ..., I_m. So the heaviest m elements, denoted as e_1, e_2, ..., e_m in E, are added to I_1, I_2, ..., I_m, respectively, at the beginning.

(2) In Step 2.2, when $I_{l(j)}+e$ contains a circuit $C(e)$ and an element needs to be removed from $C(e)\backslash S_{j-1}$, we choose the lightest. Furthermore, we can prove that $C(e)\backslash S_{j-1}$ contains at least two elements in Remark 1.

Remark 1 In Step 2.2, there are at least two elements in the subset $C(e)\backslash S_{j-1}$.

Proof From EPA we know that $C(e)\backslash S_{j-1}$ is not empty. If there is only one element $x\in C(e)\backslash S_{j-1}$, then $C(e)-x\subseteq S_{j-1}$, therefore $x\in\sigma(C(e)-x)\subseteq\sigma(S_{j-1})=S_{j-1}$. That is a conflict.

As a result, the heaviest element in $I_{l(j)}$ will not be chosen and removed by Step 2.2, i.e., each subset I_i will keep its heaviest element e_i since it is assigned to I_i at the beginning. Therefore the feasible solution produced by MEPA, denoted as $P^0=\{I_1^0,\ I_2^0, ..., I_m^0\}$, has the following property:

$$e_i \in I_i^0, \quad i=1, 2, ..., m. \tag{3}$$

Example 1 Let m=2 and M be the graphic matroid defined on the edge set E={e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8} of the graph G as shown in Fig.1.

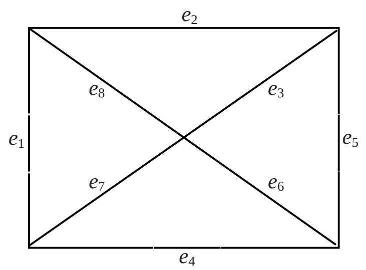

Fig.1 Graph *G*

The weights are given as: $w(e_1)=a$, $w(e_2)=a-3$, $w(e_3)=a-7$, $w(e_4)=a-8$, $w(e_5)=10$, $w(e_6)=9$, $w(e_7)=4$, $w(e_8)=1$, where a is a real number large enough. Obviously, I_1^* = {e_1, e_4, e_5, e_8}, I_2^* = {e_2, e_3, e_6, e_7} is an optimal partition with the optimal objective value $C^*=2a+3$. Performing MEPA, we add elements e_1, e_2, ..., e_7 to the partition directly and obtain $I_1^0=\{e_1, e_4, e_6\}$, $I_2^0=\{e_2, e_3, e_5, e_7\}$. But the assignment of e_8 is needed to execute Step 2.2: first we add it to I_2^0 and find the unique circuit $C(e_8)$={e_2, e_3, e_8}; then we choose and move e_3 from I_2^0 to I_1^0; finally we get the partition I_1^0 = {e_1, e_4, e_6, e_3}, I_2^0 = {e_2, e_5, e_7, e_8} with the objective value $C^{\mathrm{A}}=w(I_1^0)=3a-6$.

The example shows that Step 2.2 may seriously deteriorate the result. Therefore we want to adjust the feasible partition P^0 produced by MEPA. The adjustment is based on the idea that when removing the heaviest element from the heaviest subset and when its weight is still greater than the lightest subset, we move the heaviest element to the lightest subset.

The following is the sub-algorithm EXA:

Step 3.0: Set $I_i=I_i^0$ for i=1, 2, ..., m. Find the indices α and β such that $w(I_\alpha)=\max\{w(I_i)$: i=1, 2, ..., $m\}$, $w(I_\beta)=\min\{w(I_i)$: i=1, 2, ..., $m\}$.

Step 3.1: Find the heaviest elements z_α and z_β in subsets I_α and I_β, respectively. If $w(z_\alpha)\le w(z_\beta)$ or $w(I_\alpha-z_\alpha)\le w(I_\beta)$, stop.

Step 3.2: $I_\alpha=I_\alpha-z_\alpha$. If $I_\beta+z_\alpha$ is independent, $I_\beta=I_\beta+z_\alpha$, go to Step 3.3; else find the unique circuit $C(z_\alpha)\subseteq I_\beta+z_\alpha$, and choose $x\in C(z_\alpha)\backslash\sigma(I_\alpha)$ such that $w(x)$ is the smallest one except for z_α, and set $I_\beta=I_\beta+z_\alpha-x$, $I_\alpha=I_\alpha+x$.

Step 3.3: If $w(I_\alpha)<\max\{w(I_i)$: i=1, 2, ..., $m\}$, or $w(I_\beta)=\min\{w(I_i)$: i=1, 2, ..., $m\}$, stop; else find the index β such that $w(I_\beta)=\min\{w(I_i)$: i=1, 2, ..., $m\}$, go to Step 3.1.

Example 2 Consider Example 1 again. By MEPA, we get the partition I_1^0 = {e_1, e_4, e_6, e_3}, I_2^0 = {e_2, e_5, e_7, e_8} with the objective value $C^{\mathrm{A}}=3a-6$. Because $w(I_1^0\setminus e_1)>w(I_2^0)$, we move the element e_1 in I_1^0 to I_2^0, and choose and move element e_8 in I_2^0 to I_1^0 by Step 3.2. Then we get the partition I_1={e_4, e_6, e_3, e_8}, I_2={e_2, e_5, e_7, e_1} with the objective value $C^{\mathrm{A}}=w(I_2)=2a+11$. So the result is improved greatly.

Now combining MEPA with EXA, we get Algorithm A. The following remarks are needed for proving the correctness of EXA and estimating the approximation ratio of Algorithm A.

Remark 2 In Step 3.2 there must be an element in $C(z_\alpha)\backslash\sigma(I_\alpha)$ besides the element z_α.

Proof Otherwise, we would have $C(z_\alpha)-z_\alpha\subseteq\sigma(I_\alpha)$, then $z_\alpha\in\sigma(C(z_\alpha)-z_\alpha)\subseteq\sigma(\sigma(I_\alpha))=\sigma(I_\alpha)$, but by Step 3.2, $z_\alpha\notin\sigma(I_\alpha)$. That is a conflict.

Remark 3 The algorithm EXA must be completed after up to $m-1$ executions of Step 3.2.

Proof Let $P^0=\{I_1^0, I_2^0, ..., I_m^0\}$ be the matroid partition obtained by MEPA, $P^t=\{I_1^t, I_2^t, ..., I_m^t\}$ be the partition after the tth execution of Step 3.2 and z_i^t be the heaviest element in I_i^t for i=1, 2, ..., m. Obviously we have $z_i^0=e_i$ by Eq.(3). Suppose α_t and β_t be the indices such that $w(I_{\alpha_t}^t)=\max\{w(I_i^t):\ i=1, 2, ..., m\}$, $w(I_{\beta_t}^t)=\min\{w(I_i^t):\ i=1,\ 2,\ ...,\ m\}$. Let index set $\Lambda^t=\{k:\ w(z_k^t)<w(z_{\alpha_t}^t),\ 1\le k\le m\}$.

Now we have the following facts:

(1) $\Lambda^0\subseteq\{\alpha_0+1, \alpha_0+2, ..., m\}$, and $|\Lambda^0|\le m-1$;

(2) If $\Lambda^t=\varnothing$, this implies $w(z_{\alpha_t}^t)\le w(z_{\beta_t}^t)$, then P^t is the final partition according to Step 3.1;

(3) If P^t is not the final partition, then we can prove $\alpha_t=\alpha_0$.

In fact, if $\alpha_t \neq \alpha_{t-1}$, then $w(I^t_{\alpha_{t-1}}) < w(I^t_{\alpha_t}) = \max\{w(I^t_i): i=1, 2, ..., m\}$, the EXA will stop according to Step 3.3 and P^t is the final partition. So we can prove fact (3) by mathematical induction.

Now, let us prove $\Lambda^t \subset \Lambda^{t-1}$ and $|\Lambda^t|<|\Lambda^{t-1}|$. In fact if $k\in\Lambda^t$, then $k\neq\alpha_{t-1}, \beta_{t-1}$, and $w(z^{t-1}_k) = w(z^t_k) < w(z^t_{\alpha_t}) \le w(z^{t-1}_{\alpha_{t-1}})$, so $k\in\Lambda^{t-1}$. On the other hand, $\beta_{t-1}\in\Lambda^{t-1}\backslash\Lambda^t$.

Therefore, Λ^t will become empty after no more than $m-1$ executions of Step 3.2.

ESTIMATION OF THE WORST RATIO FOR ALGORITHM A

In this section we will analyze Algorithm A and give an estimation of the worst-case ratio. We adopt the notions used in the proof of Remark 3.

Lemma 1 Suppose $P=\{I_1, I_2, ..., I_m\}$ is the matroid partition of the set E obtained by Algorithm A and α, β are the indices of the heaviest and lightest subsets in the partition P, respectively, i.e., $w(I_\alpha)=\max\{w(I_i): i=1, 2, ..., m\}$, $w(I_\beta)=\min\{w(I_i): i=1, 2, ..., m\}$, then one of the following inequalities must be satisfied:

$$w(I_\alpha - z_\alpha) \le w(I_\beta), \tag{4}$$

$$w(z_\alpha) \le w(z_\beta), \tag{5}$$

$$w(I_\alpha) \le \sum\nolimits_{i\neq\alpha} w(I_i), \tag{6}$$

where z_α and z_β are the heaviest elements in subsets I_α and I_β, respectively.

Proof Suppose $P=\{I_1, I_2, ..., I_m\}$ is the final partition which is obtained from the partition $P^t=\{I^t_1, I^t_2, ..., I^t_m\}$ by Step 3.2. That is, if $I^t_{\beta_t} + z^t_{\alpha_t}$ is independent, then

$$I_{\beta_t} = I^t_{\beta_t} + z^t_{\alpha_t},\ \ I_{\alpha_t} = I^t_{\alpha_t} - z^t_{\alpha_t},$$

otherwise,

$$I_{\beta_t} = I^t_{\beta_t} + z^t_{\alpha_t} - x,\ \ I_{\alpha_t} = I^t_{\alpha_t} - z^t_{\alpha_t} + x,$$

where x is chosen as in Step 3.2. For $i\neq\alpha_t, \beta_t$, $I_i = I^t_i$.

It is easy to obtain

$$w(z_{\beta_t}) > w(z_{\alpha_t}), \tag{7}$$

$$w(I_{\beta_t} \setminus z_{\beta_t}) < w(I_i),\ \ i=1, 2, ..., m. \tag{8}$$

To prove Lemma 1, it is sufficient to show that if neither Eq.(4) nor Eq.(5) holds, then Eq.(6) must hold.

Now assume that neither Eq.(4) nor Eq.(5) holds. It is easy to see that one of the following must hold:

$$w(I_{\alpha_t}) < \max\{w(I_i): i=1, 2, ..., m\}, \tag{9}$$

$$w(I_\beta) = \min\{w(I_i): i=1, 2, ..., m\}. \tag{10}$$

If Eq.(9) does not hold, then $\alpha=\alpha_t$, and Eq.(10) holds, which implies $\beta=\beta_t$. From the inequality Eq.(7) we obtain $w(z_\alpha)\le w(z_\beta)$, i.e., the inequality Eq.(5) holds. It is a conflict. Therefore Eq.(9) must hold, which implies $\alpha\neq\alpha_t$. If $\alpha=\beta_t$, then Eq.(4) holds according to Eq.(8). It is a conflict. Therefore we have $\alpha\neq\alpha_t$ and $\alpha\neq\beta_t$. Hence, $w(I_\alpha) = w(I^t_\alpha) \le w(I^t_{\alpha_t}) + w(I^t_{\beta_t}) = w(I_{\alpha_t}) + w(I_{\beta_t})$, Eq.(6) holds.

Let C^{A} denote the objective value obtained by Algorithm A and C^* denote the optimal value. We have the following main theorem:

Theorem 1 The inequality

$$\frac{C^{\text{A}}}{C^*} \le \max\left\{2-\frac{1}{m},\ \frac{rm}{r+m-1},\ \frac{m}{2}\right\} \tag{11}$$

holds for every instance, where r is the rank of E.

Proof Let $P=\{I_1, I_2, ..., I_m\}$ be the partition produced by Algorithm A , α and β be the indices of the heaviest and lightest subsets of P, respectively, i.e., $w(I_\alpha)=\max\{w(I_i): i=1, 2, ..., m\}$, $w(I_\beta)=\min\{w(I_i): i=1, 2, ..., m\}$. Obviously, $C^{\text{A}}=w(I_\alpha)$. By Lemma 1, one of the inequalities Eqs.(4)~(6) holds.

If the inequality Eq.(4) holds, then

$$\frac{C^{\text{A}}}{C^*} \le \frac{\sum_{i=1}^m w(I_i)}{mC^*} + \frac{\sum_{i\neq\alpha}[w(I_\alpha)-w(I_i)]}{mC^*}$$
$$\le 1 + \frac{(m-1)w(z_\alpha)}{mC^*} \le 2-\frac{1}{m}.$$

If the inequality Eq.(5) holds, then

$$\frac{C^{\mathrm{A}}}{C^*} \le \frac{\sum_{i=1}^{m} w(I_i)}{mC^*} + \frac{(m-1)w(I_\alpha) - \sum_{i\ne\alpha} w(I_i)}{w(I_\alpha) + \sum_{i\ne\alpha} w(I_i)}$$

$$\le 1 + \frac{w(I_\alpha) - w(I_\beta)}{w(I_\alpha)/(m-1) + w(I_\beta)}$$

$$\le 1 + \frac{rw(z_\alpha) - w(z_\beta)}{rw(z_\alpha)/(m-1) + w(z_\beta)}$$

$$\le \frac{rm}{r+m-1}.$$

If the inequality Eq.(6) holds, then

$$\frac{C^{\mathrm{A}}}{C^*} = \frac{w(I_\alpha)}{C^*} \le \frac{w(I_\alpha) + \sum_{i\ne\alpha} w(I_i)}{2C^*} \le \frac{mC^*}{2C^*} = \frac{m}{2}.$$

So the inequality Eq.(11) holds.

CONCLUSION

We consider in this paper the set partitioning problem with matroid constraint. The objective is to minimize the weight of the heaviest subset. The problem is the generation of the famous scheduling problem $P||C_{\min}$ and the *k*-partitioning problem. We present an approximation algorithm for the problem and give an estimation of the worst ratio for the algorithm.

There still remains for further study the solution of the problem where the objective is to maximize the weight of the lightest subset.

References

Babel, L., Kellerer, H., Kotov, V., 1998. The *k*-partitioning problem. *Math. Methods Oper. Res.*, **47**(1):59-82. [doi:10.1007/BF01193837]

Burkard, R.E., Yao, E.Y., 1990. Constrained partitioning problems. *Discr. Appl. Math.*, **28**(1):21-34. [doi:10.1016/0166-218X(90)90091-P]

Edmonds, J., 1965. Minimum partition of a matroid into independent subsets. *J. Res. NBS*, **69B**:67-72.

Garey, M.R., Johnson, D.S., 1978. Computers and Intractability: A Guide to the Theory of NP-completeness. Freeman, San Francisco, CA.

Graham, R.L., 1969. Bounds on multiprocessing timing anomalies. *SIAM J. Appl. Math.*, **17**(2):416-429. [doi:10.1137/0117039]

Hwang, F.K., 1981. Optimal partitions. *J. Opt. Theory Appl.*, **34**(1):1-10. [doi:10.1007/BF00933355]

Hwang, F.K., Sun, J., Yao, E.Y., 1985. Optimal set partitioning. *J. Algebr. Methods*, **6**(1):163-170. [doi:10.1137/0606015]

Lawler, E.L., 1976. Combinatorial Optimization: Networks and Matroids. Holt, Rinehart and Winston.

Lee, C.Y., Liman, S.D., 1993. Capacitated two-parallel machines scheduling to minimize sum of job completion times. *Discr. Appl. Math.*, **41**(3):211-222. [doi:10.1016/0166-218X(90)90055-H]

Welsh, D.J.A., 1976. Matroid Theory. Academic Press, London.