第42卷 第7期 计 算 机 科 学 Vol. 42 No. 7
2015年7月 Computer Science July 2015

具有数目约束的负载均衡问题

李伟东 李建平
(云南大学 昆明 650091)

摘 要 考虑了具有数目约束的负载平衡问题的一种特殊情形,称之为2-半匹配问题。分析了此问题在3种目标函数下的计算复杂性,并设计了相应的近似算法。

关键词 负载均衡,NP-难,APX-难,近似算法

中图法分类号 TP30,O223 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.7.016

Cardinality-constrained Load Balancing Problem

LI Wei-dong LI Jian-ping
(Yunnan University, Kunming 650091, China)

Abstract A special case of the cardinality-constrained load balancing problem, called 2-semi-matching problem was considered. The computational complexity and three approximation algorithms were presented under three different objectives, respectively.

Keywords Load balancing, NP-hard, APX-hard, Approximation algorithms

1 引言

以最小化最大机器负载[1,2]为目标的均衡问题因其在工业生产、网络设计、并行计算和网络资源分配等领域的广泛应用,自20世纪60年代起就成为了理论计算机科学和运筹学等领域研究的重点之一。最近10年,以最大化最小负载[3,4]和最小化负载向量的p范数[5-7]为目标的负载均衡问题及其推广引起了学者的兴趣。本文重点研究具有数目约束的负载均衡问题(简记为CCLC问题),其定义如下:

给定机器集$M=\{M_1,M_2,\cdots,M_m\}$和任务集$J=\{J_1,J_2,\cdots,J_{km}\}$,任务$J_j$在机器$M_i$上的加工时间(或称大小)为$p_{ij}$,将这$km$个任务分配到$m$台机器上,每台机器上恰好分配$k$个任务,使得各机器的负载尽可能地均衡。令$S_i$表示在机器$M_i$上加工的任务集,机器$M_i$的负载定义为在其上加工的所有任务的加工时间之和,记为$l_i=\sum_{j:J_j\in S_i}p_{ij}$。令向量$\boldsymbol{L}=(l_1,l_2,\cdots,l_m)$,主要考虑下面3种目标函数:

(1)最大机器负载尽可能达到最小,即
$\min_L\max\{l_i\mid 1\leqslant i\leqslant m\}$,简记为min-max;

(2)最小机器负载尽可能达到最大,即
$\max_L\min\{l_i\mid 1\leqslant i\leqslant m\}$,简记为max-min;

(3)向量$\boldsymbol{L}$的p范数$|\boldsymbol{L}|_p$尽可能达到最小,即
$\min_{\mathscr{L}}(\sum_{i=1}^{m}l_i^p)^{\frac{1}{p}}$,简记为$\min|\boldsymbol{L}|_p$,这里$p\in(1,+\infty)$。

为更好地陈述相关的研究成果,给出理论计算机科学领域内与本文相关的基本定义。

定义1 令Π表示一个最小(大)化问题,I表示该问题的任一实例,A表示问题Π的一个多项式时间算法,$A(I)$和$OPT(I)$分别表示算法A解实例I所得到的可行解的目标函数值和最优值,则算法A的近似值(又称最坏情形界)定义为:

$$r_A=\inf\{r_A\geqslant 1\mid \frac{A(I)}{OPT(I)}\leqslant r_A,\forall I\}$$

$$r_A=\sup\{r_A\leqslant 1\mid \frac{A(I)}{OPT(I)}\geqslant r_A,\forall I\}$$

定义2 如果一个优化问题Π存在近似值为常数的多项式时间算法,则称该问题属于APX类。

定义3[8] 令Π和Π'分别表示两个优化问题,如果存在多项式时间算法A_1,A_2及常数α,β,对Π的任一实例I都满足:

(1)算法A_1得到Π'的一个实例I',并且$OPT(I')\leqslant\alpha\cdot OPT(I)$,这里$OPT(I')$和$OPT(I)$分别表示实例$I'$和实例$I$的最优值;

(2)对实例I'的任意可行解s',其目标函数值为$c'(s')$,算法A_2能生成实例I的一个可行解$s=\tau(s')$,其目标函数值为$c(s)$,并且

$$|c(s)-OPT(I)|\leqslant\beta|c'(s')-OPT(I')|$$

则称问题Π能够L-归约到问题Π'。

定义4 如果APX类中的每一个问题都能够L-归约到问题Π,则称问题Π是APX-难的。

定义5 对于最大(小)化问题Π,若对任意的实数$\epsilon>0$,算法簇A_ϵ都能得到一个$1-\epsilon(1+\epsilon)$-近似解,则称算法簇A_ϵ是一个多项式时间近似方案(Polynomial-time Approximation Scheme, PTAS)。

Papadimitriou和Yannakakis[8]得到下面重要结果:

定理1 如果问题Π是APX-难的,则该问题不存在

到稿日期:2014-06-14 返修日期:2014-09-25 本文受国家自然科学基金(11126315,11301466)资助。

李伟东(1981-),男,博士,副教授,主要研究方向为算法及其复杂性、组合最优化,E-mail:weidong@ynu.edu.cn;李建平(1965-),男,博士,教授,主要研究方向为组合最优化。

PTAS。

因其固有的困难性，目前为止没有发现一般情形下CCLC问题的研究结果。当 $p_{ij}=p_j$ 时，即每个任务在不同的机器上的加工时间都相同时，CCLC问题即为 k-划分问题。当目标函数为min-max时，Babel等人[9]设计出了 k-划分问题的一个4/3-近似算法，这是目前最好的结果，关于 k-划分问题上下界的讨论参见文献[10-12]。当目标函数为max-min时，何勇等人[13]设计了 k-划分问题的一个 $\max\{2/k,1/m\}$-近似算法，更多相关结果见文献[14]。

当 $p_{ij}=p_j$ 且 $k=3$ 时，CCLC问题即为3-划分问题。当目标函数为min-max时，Kellerer和Woeginger[15]证明了LPT算法的近似值为 $4/3-1/(3m)$；Kellerer和Kotov[16]设计了一个7/6-近似算法。当目标函数为max-min时，Chen等人[17]证明了修正的LPT算法的近似值为 $(2m-1)/(3m-2)$。

本文考虑 $k=2$ 时的CCLC问题，即每个实例中共含有 m 台机器和 $2m$ 个工件，称此问题为2-半匹配问题。当目标函数为min-max时，证明了2-半匹配问题是 $3/2-\epsilon$-不可近似的（这里 $\epsilon>0$），并设计出了一个强多项式时间的2-近似算法；当目标函数为max-min时，证明了2-半匹配问题是 $1/2+\epsilon$-不可近似的，并设计出了一个1/2-近似算法；当目标函数为 $\min|\boldsymbol{L}|_p$ 时，证明了2-半匹配问题是APX-难的，并设计出了一个 $2^{1-1/p}$-近似算法，这里 $1<p<+\infty$。

2 目标函数为min-max

本节利用多项式归约方法分析目标函数为min-max的2-半匹配问题的不可近似比，并结合二分法和网络流技术设计强多项式时间的2-近似算法。

定理2 目标函数为min-max的2-半匹配问题是 $3/2-\epsilon$-不可近似的。

证明：采用类似于文献[2]中定理5的证明方法，将3维匹配问题多项式归约到目标函数为min-max的2-半匹配问题。

3维匹配问题的定义如下：给定3个集合 $X=\{x_1,x_2,\cdots,x_n\}$，$Y=\{y_1,y_2,\cdots,y_n\}$ 和 $Z=\{z_1,z_2,\cdots,z_n\}$ 及三元组集 $T=\{T_1,T_2,\cdots,T_m\}\subseteq X\times Y\times Z$，问：是否存在一个3维完美匹配 $T'\subseteq T$ 满足 $|T'|=n$，且 X,Y,Z 中的每个元素在 T' 中恰好出现一次。此问题是NP-难的[18]。

给定3维匹配问题的任一实例 I，构造目标函数为min-max的2-半匹配问题的一个由 m 台机器和 $2m$ 个任务组成的实例 I'。机器集 M 中的机器 M_l 代表集合 T 中的三元组 T_l。任务集 J 中有3类任务，分别用 B,C,D 来表示，其中：

- B 中包含有 n 个大小为1的任务，其中任务 $J_{b_j}(j=1,2,\cdots,n)$ 代表 Y 中的元素 y_j，任务 J_{b_j} 能在机器 M_l 上加工当且仅当 $y_j\in T_l$；
- C 中包含有 n 个大小为1的任务，其中任务 $J_{c_k}(k=1,2,\cdots,n)$ 代表 Z 中的元素 z_k，任务 J_{c_k} 能在机器 M_l 上加工当且仅当 $z_k\in T_l$；
- D 中包含有 $m-n(\geqslant 0)$ 个大小为2的"虚拟"任务和 $m-n$ 个大小为0的"虚拟"任务，其中有 $a_i(i=1,2,\cdots,n)$ 个大小为2的任务和 $a_i(i=1,2,\cdots,n)$ 个大小为0的任务代表元素 x_i，这里 a_i 等于元素 x_i 在 T 中的三元组中出现的次数减1，即 $a_i=\sum_{l=1}^{n}|T_l\cap\{x_i\}|-1$（容易验证 $\sum_{i=1}^{n}a_i=m-n$），这 $2a_i$ 个任务能在机器 M_l 上加工当且仅当 $x_i\in T_l$。

如果实例 I 存在一个3维完美匹配 $T^*=\{T_{i_1},T_{i_2},\cdots,T_{i_n}\}$，可以构造实例 I' 的一个可行解，方法如下：B 中的任务 $J_{b_j}(j=1,2,\cdots,n)$ 在机器 M_{i_l} 上加工，这里 M_{i_l} 满足 $y_j\in T_{i_l}\in T^*$；C 中的任务 $J_{c_k}(k=1,2,\cdots,n)$ 在机器 M_{i_l} 上加工，这里 M_{i_l} 满足 $z_k\in T_{i_l}\in T^*$；D 中相应于 x_i 的一个大小为2的任务和一个大小为0的任务在机器 M_{i_l} 上加工，这里 M_{i_l} 满足 $x_i\in T_{i_l}\in T\backslash T^*$。容易验证，此可行解的目标函数值为2。由于机器的平均负载为2，因此此解是最优解。

如果实例 I' 的最优值为2，则显然大小为2的任务和大小为0的任务分配在同一台机器上，2个大小为1的任务分配给同一台机器。令 $M_{i_1},M_{i_2},\cdots,M_{i_n}$ 表示分配有2个大小为1的机器，不难验证它们所代表的三元组集 $T^*=\{T_{i_1},T_{i_2},\cdots,T_{i_n}\}$ 正是实例 I 的一个3维完美匹配。

综上所述，实例 I 存在3维匹配当且仅当实例 I' 的最优值为2。又因为实例 I' 的任一可行解的目标函数值是整数，所以当实例 I 不存在3维完美匹配时，实例 I' 的最优值至少为3。这就说明了目标函数为min-max的2-匹配问题是 $3/2-\epsilon$-不可近似的。证毕。

给定2-半匹配问题的任一实例 I_{sm}，由于任务总数为 $2m$ 且每台机器恰好分配两个任务，因此当目标函数为min-max，2-半匹配问题的目标函数值至多有 mC_{2m}^2 种可能。对任意正数 T，通过构造相应的网络流问题的实例，设计出如下算法 A_1：

第1步 构造一个网络 $N=(U\cup V\cup\{s,t\},A,c)$，这里 U 中的顶点 u_i 代表机器 $M_i(i=1,2,\cdots,m)$，V 中的顶点 v_j 代表任务 $J_j(j=1,2,\cdots,2m)$，弧集合 $A=\{(s,u_i)\,|\,i=1,2,\cdots,m\}\cup\{(u_i,v_j)\,|\,p_{ij}\leqslant T\}\cup\{(v_j,t)\,|\,j=1,2,\cdots,2m\}$，相应的容量函数 c 满足

$c(s,u_i)=2,c(u_i,v_j)=c(v_j,t)=1$。

第2步 利用网络最大流算法求出 N 中的最大流 f，如果流值小于 $2m$，则输出 $OPT>T$，否则转下一步。

第3步 对每条弧 $(u_i,v_j)\in A$，如果 $f(u_i,v_j)=1$，则将任务 J_j 分配给机器 M_i，得到2-半匹配问题的一个可行解。

定理3 当目标函数为min-max时，2-半匹配问题存在着2-近似算法，其时间复杂性为 $O(m^3\log m)$，这里 m 为机器数。

证明：如果实例 I_{sm} 存在一个目标函数至多为 T 的可行解，则算法 A_1 能够输出一个目标函数值至多为 $2T$ 的可行解。如若不然，当 $OPT(I)\leqslant T$ 时，如果算法 A_1 得不到目标函数值至多为 $2T$ 的可行解，这意味着网络 N 不存在流值为 $2m$ 的流，等价于实例 I_{sm} 中的任一可行解中必存在一个分配给机器 M_i 的任务 J_j 满足 $p_{ij}>T$，即 $OPT(I_{sm})>T$，矛盾。因此，当 $OPT(I_{sm})\leqslant T$ 时，算法 A_1 能得到一个目标函数值至多为 $2T$ 的可行解。

算法 A_1 中第1步和第3步的运行时间均为 $O(m^2)$；由于网络 N 是一个最大容量为2的简单网络，由文献[19]知，第2步可以在 $O(m^{5/2})$ 时间内完成。因此，对任意可能的取值 T，算法 A_1 的运行时间为 $O(m^{5/2})$。对目标函数值的每个可

能的取值 T(至多 mC_{2m}^2 个),执行算法 A_1,在所有可行解中找到一个目标函数值最小的可行解,此可行解的目标函数值不超过 $2OPT(I_{sm})$,共需时间为 $O(mC_{2m}^2 \cdot m^{5/2})=O(m^{11/2})$,是关于输入规模的强多项式函数。容易验证,对 T 所有可能的取值在 $O(m^3\log m)$ 时间内进行排序,再用二分法来调用 $O(\log m)$ 次算法 A_1,可以得到一个 2-近似解,总的运行时间为 $O(m^3\log m)$。证毕。

3 目标函数为 max-min

当目标函数为 max-min 时,利用定理 2 中的证明方法,容易证明 2-半匹配问题是 $1/2+\varepsilon$-不可近似的,证明过程从略。进一步,结合二分法和匹配算法设计了强多项式时间的 1/2-近似算法。

定理 4 目标函数为 max-min 的 2-半匹配问题是 $1/2+\varepsilon$-不可近似的。

为了设计目标函数为 max-min 的 2-半匹配问题的近似算法,对任意正数 T,设计如下算法 A_2 求可行解:

第 1 步 构造图 $G=(U\cup V,E)$,这里 U 中的顶点 u_i 代表机器 $M_i(i=1,2,\cdots,m)$,V 中的顶点 v_j 代表任务 $J_j(j=1,2,\cdots,2m)$,边集 $E=\{(u_i,v_j)\mid p_{ij}\geqslant T/2\}$。

第 2 步 如果图 G 存在一个基数为 m 的最大匹配 $MM\subseteq E$,对边 $(u_i,v_j)\in MM$,将任务 J_j 分配给机器 M_i,最后分配剩余的 m 个任务使得每台机器恰好分配两个任务。否则,输出 $OPT(I_{sm})<T$。

定理 5 当目标函数为 max-min 时,2-半匹配问题存在着一个 1/2-近似算法,其时间复杂性为 $O(m^3\log m)$。

证明:如果实例 I_{sm} 存在一个目标函数至少为 T 的可行解,算法 A_2 能得到一个目标函数值至少为 $T/2$ 的可行解。否则,如果算法 A_2 得不到目标函数值至少为 $T/2$ 的可行解,这意味着图 G 不存在基数为 m 的匹配,这等价于实例 I_{sm} 的任一可行解中存在一台机器 M_i 所分配的两个任务大小都小于 $T/2$,即 $OPT(I_{sm})<T$。矛盾。

在算法 A_2 中,第 1 步的运行时间为 $O(m^2)$;调用文献[19]中求最大基数匹配的算法,第 2 步可以在 $O(m^{5/2})$ 时间内完成。对目标函数值的每个可能的取值 T(至多 mC_{2m}^2 个),运行算法 A_2,在所有可行解中找到一个目标函数值最大的可行解,共需时间为 $O(mC_{2m}^2\cdot m^{5/2})=O(m^{11/2})$。同定理 3 类似,算法的运行时间可以降为 $O(m^3\log m)$。证毕。

4 目标函数为 $\min|\boldsymbol{L}|_p$

当目标函数为 $\min|\boldsymbol{L}|_p$ 时,利用文献[8]中 L-归约技术,证明 2-半匹配问题是 APX-难的,即不存在 PTAS。进一步,结合匹配算法和凸函数的性质设计了 $2^{1-1/p}$-近似算法。

定理 6 目标函数为 $\min|\boldsymbol{L}|_p$ 的 2-半匹配问题是 APX-难的。

证明:只要证明了目标函数为 $\min_L\sum_{i=1}^m l_i^2$ 的 2-半匹配问题是 APX-难的,可以直接推出目标函数为 $\min|\boldsymbol{L}|_2$ 的 2-半匹配问题是 APX-难的。对于其余的实数 $p>1$,可以类似地得到相同的结果。

下面将限制的 $Max3\text{-}DM\text{-}3$ 问题[20] L-归约到 2-半匹配问题,从而证明目标函数为 $\min_L\sum_{i=1}^m l_i^2$ 的 2-半匹配问题是 APX-难的。

限制的 $Max\text{-}3DM\text{-}3$ 问题的定义如下:给定 3 个集合 $X=\{x_1,x_2,\cdots,x_n\}$,$Y=\{y_1,y_2,\cdots,y_n\}$ 和 $Z=\{z_1,z_2,\cdots,z_n\}$ 及三元组集 $T=\{T_1,T_2,\cdots,T_m\}\subseteq X\times Y\times Z$,这里 X,Y,Z 中的每个元素在 T 中的三元组出现的次数为 1,2 或 3,这意味着 $n\leqslant m\leqslant 3n$。特别地,T 中包含一个完美匹配,即存在 $T^*\subseteq T$ 满足 $|T^*|=n$ 且 X,Y,Z 中的每个元素在 T^* 中的三元组中恰好出现一次。寻找一个集合 $T'\subseteq T$,使得 X,Y,Z 中的每个元素在 T' 中至多出现一次,其目标是使得 $|T'|$ 达到最大。Petrank[20] 证明了此问题是 APX-难的。

给定限制的 $Max\text{-}3DM\text{-}3$ 问题的任一实例 I,构造 2-半匹配问题的一个由 $3n$ 台机器和 $6n$ 个任务组成的实例 I',其中含有 m 台机器的集合 M 中的机器 M_l 代表集合 T 中的三元组 T_l,机器集 N 中的 $3n-m$ 台机器起辅助作用。任务集 J 包含 4 类任务,分别用 B,C,D,E 来表示,其中:

- B 中包含有 n 个大小为 1 的任务,其中任务 $J_{b_j}(j=1,2,\cdots,n)$ 代表 Y 中的元素 y_j,任务 J_{b_j} 能在机器 M_l 上加工当且仅当 $y_j\in T_l$;
- C 中包含有 n 个大小为 1 的任务,其中任务 $J_{c_k}(k=1,2,\cdots,n)$ 代表 Z 中的元素 z_k,任务 J_{c_k} 能在机器 M_l 上加工当且仅当 $z_k\in T_l$;
- D 中包含有 $m-n(\geqslant 0)$ 个大小为 2 的"虚拟"任务和 $m-n$ 个大小为 0 的"虚拟"任务,其中有 $a_i(i=1,2,\cdots,n)$ 个大小为 2 的任务和 $a_i(i=1,2,\cdots,n)$ 个大小为 0 的任务代表元素 x_i,这里 a_i 等于元素 x_i 在 T 中的三元组中出现的次数减 1,即 $a_i=\sum_{l=1}^n|T_l\cap\{x_i\}|-1$,这 $2a_i$ 个任务能在机器 M_l 上加工当且仅当 $x_i\in T_l$。
- E 中包含有 $6n-2m(\geqslant 0)$ 个大小为 1 的"辅助"任务,它们可以在 N 中的任何一台机器上加工。

由限制的 $Max\text{-}3DM\text{-}3$ 问题实例的定义知,$OPT(I)=n$。给定实例 I 的一个最优解 $T^*=\{T_{i_1},T_{i_2},\cdots,T_{i_n}\}$,构造实例 I' 的一个可行解,方法如下:

- B 中的任务 $J_{b_j}(j=1,2,\cdots,n)$ 在机器 M_{i_l} 上加工,这里 M_{i_l} 满足 $y_j\in T_{i_l}\in T^*$;
- C 中的任务 $J_{c_k}(k=1,2,\cdots,n)$ 在机器 M_{i_l} 上加工,这里 M_{i_l} 满足 $z_k\in T_{i_l}\in T^*$;
- D 中对应于 x_i 的一个大小为 2 的任务和一个大小为 0 的任务在机器 M_{i_l} 上加工,这里 M_{i_l} 满足 $x_i\in T_{i_l}\in T\backslash T^*$;
- E 中两个大小为 1 的任务在 N 中的一台机器上加工。

容易验证,此可行解中 $3n$ 台机器的负载都是 2。因此,得到

$$OPT(I')\leqslant 2^2\times 3n=12n=12OPT(I)$$

这就完成了 L-归约中第一个条件的证明,其中 $\alpha=12$。

令 s' 表示实例 I' 的任一可行解,下面将构造实例 I 的一个可行解 $s=\tau(s')$。令 $m_i(i=0,1,2)$ 表示可行解 s 中加工 $B\cup C$ 中的 i 个任务的机器数。由于 M 中代表三元组中含有变量 x_i 的机器共有 a_i+1 个,并且 D 中共有 $2a_i$ 个任务必须在这 a_i+1 台机器上加工,因此,这 a_i+1 台机器中至多有一台机器加工 $B\cup C$ 中的两个任务。s' 中加工 $B\cup C$ 中的两个任

务的机器所对应的 T 中的三元组构成实例 I 的一个可行解 s，并且此可行解的目标函数值 $c(s)=m_2$。

根据 m_i 的定义，可以得到下面的结果：

$$m_1+2m_2=2n \tag{1}$$

$$m_0+m_1+m_2=m \tag{2}$$

由式(1)得

$$m_2\leqslant n \tag{3}$$

式(2)－式(1)，得

$$m_0=m-2n+m_2\leqslant m-n;$$

由式(2)和式(3)得到

$$m_0+m_1=m-m_2\geqslant m-n$$

即

$$m_0\leqslant m-n\leqslant m_0+m_1 \tag{4}$$

下面分析实例 I' 的可行解 s' 的目标函数值的下界。$B\cup C$ 中的任务分配方式不变，重新分配 $D\cup E$ 中的任务。由 p-范数的凸性知，目标函数值最小的排序方式是将 D 中 m_0 个大小为 2 的任务分配给当前负载为 0 的 M 中的机器（由式(4)中 $m_0\leqslant m-n$ 知，此法可行）；D 中 $m-n-m_0$ 个大小为 2 的任务分配给当前负载为 1 的 M 中的机器；D 中大小为 0 的任务分配给只加工一个任务的机器；2 个 E 中的任务分配给一台 N 中的机器。此时，负载为 3 的机器共有 $m-n-m_0$ 台，负载为 2 的机器共有 m_0+m_2+3n-m 台（M 中有 m_0+m_2 台，N 中有 $3n-m$ 台），负载为 1 的机器共有 $m_1-(m-n-m_0)$ 台。因此，对任意可行解 s'，其目标函数值为：

$$\begin{aligned}c'(s')&\geqslant 3^2(m-n-m_0)+2^2(m_0+m_2+3n-m)+m_0+m_1-m+n\\&=-4m_0+m_1+4m_2+4m+4n\\&=-4(m-m_1-m_2)+m_1+4m_2+4m+4n\\&=-4(m-(2n-2m_2)-m_2)+(2n-2m_2)+4m_2+4m+4n\\&=14n-2m_2\end{aligned}$$

这里第二个等式是由式(2)得到的，第三个等式是由式(1)得到的。因此，

$$|c(s)-OPT(I)|=n-m_2=\frac{1}{2}(14n-2m_2-12n)\leqslant\frac{1}{2}|c'(s')-OPT(I')|$$

这就证明了 L-归约的第二个条件是成立的，其中 $\beta=\frac{1}{2}$。

由于限制的 Max-$3DM$-3 问题是 APX-难的[20]，因此目标函数为 $\min|\boldsymbol{L}|_2$ 的 2-半匹配问题也是 APX-难的，即不存在 PTAS。证毕。

下面给出目标函数为 $\min|\boldsymbol{L}|_p$ 的 2-半匹配问题的一个多项式时间算法 A_3：

第 1 步　构造一个完全二部图 $G=(U\cup V,E,w)$，$|U|=|V|=2m$，对 $i=1,2,\cdots,m$，U 中的顶点 u_i 和 $u_{i'}$ 代表机器 M_i；对 $j=1,2,\cdots,2m$，V 中的顶点 v_j 代表任务 J_j，边 $(u_i,v_j)\in E$ 的权重 $w(u_i,v_j)$ 和 $(u_{i'},v_j)\in E$ 的权重 $w(u_{i'},v_j)$ 均为 p_{ij}^p。

第 2 步　求出二部图 G 的最小权重的完美匹配 $PM\subseteq E$。

第 3 步　若边 $(u_i,v_j)\in PM$ 或 $(u_{i'},v_j)\in PM$，则将任务 J_j 分配给机器 M_i。

定理 7　算法 A_3 的近似比为 $2^{1-1/p}$，时间复杂性为 $O(m^3)$，这里 m 为机器数。

证明：对任一给定的实例 I_{sm}，执行算法 A_3，令 J_{j_1} 和 J_{j_2} 表示算法 A_3 的输出解中分配给机器 M_i 的任务，$J_{j_1^*}$ 和 $J_{j_2^*}$ 表示最优解中分配给机器 M_i 的任务。令 OUT 表示输出解的目标函数值，有

$$\begin{aligned}OUT^p&=\sum_{i=1}^{m}(p_{ij_1}+p_{ij_2})^p\\&\leqslant 2^{p-1}\sum_{i=1}^{m}(p_{ij_1}^p+p_{ij_2}^p)\leqslant 2^{p-1}\sum_{i=1}^{m}(p_{ij_1^*}^p+p_{ij_2^*}^p)\\&\leqslant 2^{p-1}\sum_{i=1}^{m}(p_{ij_1^*}+p_{ij_2^*})^p\\&\leqslant 2^{p-1}OPT^p\end{aligned}$$

这里第一个不等式是由不等式 $(\frac{x+y}{2})^p\leqslant\frac{1}{2}(x^p+y^p)$（因 $f(t)=t^p$ 是凸函数）得到的，第二个不等式是因为算法 A_3 选择了最小权重的完美匹配，最后一个不等式是因为 $p_{ij_1^*},p_{ij_2^*}\geqslant 0$。因此，$OUT\leqslant 2^{1-1/p}OPT$。

算法 A_3 的第 1 步和第 3 步均可在 $O(m)$ 时间内完成，第 2 步的运行时间为求二部图最小权重完美匹配所需的时间 $O(m^3)$[19]。因此，定理成立。证毕。

结束语　设计了 3 种不同目标函数下的 2-半匹配问题的多项式时间近似算法，未来的研究重点之一是设计目标函数为 min-max 或 $\min|\boldsymbol{L}|_p$ 的 2-半匹配问题近似比更好的多项式时间算法。

参 考 文 献

[1] Graham R L. Bounds for certain multiprocessing anomalies [J]. Bell System Technical Journal, 1966, 45(9), 1563-1581

[2] Lenstra J K, Shmoys D B, Tardos E. Approximation algorithms for scheduling unrelated parallel machines [J]. Mathematical Programming, 1990, 46(3): 259-271

[3] Chakrabarty D, Chuzhoy J, Khanna S. On allocating goods to maximize fairness [C]// Proceedings of 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS). 2009: 107-116

[4] Asadpour A, Saberi A. An approximation algorithm for max-min fair allocation of indivisible goods [J]. SIAM Journal on Computing, 2010, 39(7): 2970-2989

[5] Azar Y, Epstein A. Convex programming for scheduling unrelated parallel machines [C]// Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC). 2005: 331-337

[6] Kumar V S A, Marathe M V, Parthasarathy S, et al. A unified approach to scheduling on unrelated parallel machines [J]. Journal of the ACM, 2009, 56(5): 28

[7] Bansal N, Vredeveld T, Zwaan R. Approximating vector scheduling: Almost matching upper and lower bounds [C]// Lecture Notes in Computer Science 8392. 2014: 47-59

[8] Papadimitriou C, Yannakakis M. Optimization, approximation, and complexity classes [J]. Journal of Computer and System Sciences, 1991, 43(3): 425-440

[9] Babel L, Kellerer H, Kotov V. The k-partitioning problem [J]. Mathematical Methods of Operations Research, 1998, 47(1): 59-82

（下转第 90 页）

明本文算法能够保持灰度均值算法的优势。

因变换域算法在现今被广泛使用，而且与空域图像水印相比，基于DCT的数字水印对压缩、滤波和其他一些攻击具有更强的鲁棒性，故将本文算法与文献[5]进行比较。实验中本文步长$\delta=4$，这与文献[5]中量化步长$Q=70$时的PSNR相当。

由于文献[5]使用的水印图像大小为32×32，而嵌入的信息量直接影响水印的鲁棒性，因此本文采用相同的水印与文献[5]进行鲁棒性比较。根据本文的分区方法，分区大小变为16×16。同样，先将Lena彩色图像转换成256级灰度图像并使用式(8)来修改载体图像。水印图像如图11所示，实验结果如表8所列。

图11　与文献[5]比较所用的水印

表8　本文算法与文献[5]的对比结果

攻击类型	本文算法NC值	文献[5]的NC值
裁剪1/8	最坏0.875*	0.9477
裁剪1/4	最坏0.75*	0.7908
3×3均值滤波	0.9331	0.9656
5×5均值滤波	0.8317	0.8765
椒盐噪声0.2%	0.9978	0.9774
椒盐噪声0.5%	0.9859	0.9570
JPEG压缩因子:40	1.0000	0.9720
JPEG压缩因子:20	0.9913	0.8840

因为算法的抗裁剪攻击能力与被裁剪区域的位置有关，所以此处表8中测定的是没有发生区域替换的情况下的最坏NC值(以"*"标记)。在有区域替换的情况下最坏NC值由受损最轻的区域决定。最坏情况是指被裁剪区域嵌入的水印信息全部丢失。由表8可以看出本文算法在对抗噪声攻击与压缩攻击方面优于文献[5]，这是分区的大小增加了4倍而使均值更加稳定的缘故。因此，在分区较大的条件下本文算法对于噪声与压缩攻击效果更好。

结束语　本文提出了一种适用于Android智能手机的灰度均值图像水印算法。该算法实现简单、时间复杂度低，适合Android平台。实验结果表明，本算法不仅具有良好的透明性，而且对JPEG压缩、加噪等攻击具有较强的鲁棒性，同时有较好的抵抗裁剪攻击能力。由于水印的1,2,3,4区域一般是相似的，因此可以考虑存储区域之间的差异。由于相似区域的差异很少，甚至有时完全没有或者少得可忽略，此时便可将整个区域当作一个整体来量化以提高区域均值的稳定性，这为今后的研究提供了方向。

参 考 文 献

[1] 张晓强，王蒙蒙，朱贵良. 图像水印算法研究新进展[J]. 计算机工程与科学，2012，34(4)：17-22
Zhang Xiao-qiang, Wang Meng-meng, Zhu Gui-liang. A Novel Survey on the Image Watermarking Algorithms[J]. Computer Engineering & Science, 2012, 34(4): 17-22

[2] Zhu Shao-min, Liu Jian-ming. A novel fragile watermarking scheme for image tamper detection and recovery[J]. Chinese Optics Letters, 2010, 8(7): 661-665

[3] Xiang S, Kim H, Huang J. Invariant image watermarking based on statistical features in the low-frequency domain [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2008, 18(6): 777-790

[4] 李旭东. 基于图像灰度平均值的数字水印算法[J]. 武汉大学学报，2007，32(6)：556-559
Li Xu-dong. Mean Gray Value Based Image Watermarking Algorithm[J]. Geomatics and Information Science of Wuhan University, 2007, 32(6): 556-559

[5] 王友卫，申铉京，吕颖达，等. 基于DCT和SVD变换的盲数字水印算法[J]. 计算机工程与应用，2011，41(21)：157-161
Wang You-wei, Shen Xuan-jing, Lv Ying-da, et al. Blind digital watermarking algorithm based on DCT and SVD transform[J]. Computer Engineering and Applications, 2011, 47(21): 157-161

[6] 陈利，何选森，赵天娇. 一种自适应的奇偶量化水印算法[J]. 计算机工程与应用，2011，47(35)：203-205
Chen Li, He Xuan-sen, Zhao Tian-jiao. Adaptive odd-even quantization watermarking algorithm[J]. Computer Engineering and Applications, 2011, 47(35): 203-205

[7] 陈益飞，曹瑞. 混沌理论在水印置乱中的应用[J]. 电子设计工程，2011，19(1)：157-160
Chen Yi-fei, CAO Rui. Application of the chaos theory in the watermarking scrambling[J]. Electronic Design Engineering, 2011, 19(1): 157-160

(上接第77页)

[10] Dell'Amico M, Martello S. Bounds for the cardinality constrained $P|C_{max}$ problem [J]. Journal of Scheduling, 2001, 4(3): 123-138

[11] Dell'Amico M, Iori M, Martello S. Heuristic algorithms and scatter search for the cardinality constrained $P|C_{max}$ problem [J]. Journal of Heuristics, 2004, 10(2): 169-204

[12] Dell'Amico M, Iori M, Martello S, et al. Lower bound and heuristic algorithms for the k_i partitioning problem [J]. European Journal of Operational Research, 2006, 171(3): 725-742

[13] He Y, Tan Z, Zhu J, et al. k-Partitioning problems for maximizing the minimum load [J]. Computers and Mathematics with Applications, 2003, 46(10/11): 1671-1681

[14] Bruglieri M, Ehrgott M, Hamacher H W, et al. An annotated bibliography of combinatorial optimization problems with fixed cardinality constraints [J]. Discrete Applied Mathematics, 2006, 154(9): 1344-1357

[15] Kellerer H, Woeginger G. A tight bound for 3-partitioning [J]. Discrete Applied Mathematics, 1993, 45(3): 249-259

[16] Kellerer H, Kotov V A. 7/6-approximation algorithm for 3-partitioning and its application to multiprocessor scheduling [J]. INFOR: Information Systems and Operational Research, 1999, 37(1): 48-56

[17] Chen S P, He Y, Lin G H. 3-partitioning for maximizing the minimum load [J]. Journal of Combinatorial Optimization, 2002, 6(1): 67-80

[18] Garey M R, Johnson D S. Computer and Intractability: A Guide to The Theory of NP-Completeness [D]. San Francisco: W. H. Freeman and Company, 1979

[19] Ahuja R K, Magnanti T L, Orlin J B. Network Flows: Theory, Algorithms, and Applications [M]. Prentice Hall, NJ, 1993

[20] Petrank E. The hardness of approximation: gap location [J]. Computational Complexity, 1994, 4(2): 133-157