互联网信息组织和规划中的带拒绝装箱问题

何 勇[1),2)] 谈之奕[1),2)] 任 峰[1)]

[1)] (浙江大学数学系 杭州 310027)

[2)] (浙江大学 CAD & CG 国家重点实验室 杭州 310027)

摘 要 讨论如下定义的带拒绝装箱问题:设有许多等长的一维箱子，给定一个物品集，每个物品有两个参数:长度和罚值.物品可以放入箱子也可被拒绝放入箱子.如果将物品放入箱子，则使该箱剩余长度减少.一旦需将某一物品放入某一箱中，而该箱的剩余长度不够时，则需启用新箱子.如果物品被拒绝放入任何箱中，则产生惩罚.问怎样安排物品使所用箱子数与未装箱的物品总罚值之和最小.该问题是一个新的组合优化问题，来源于内部互联网的信息组织和规划.该文首先给出一个最优解值的下界估计，它可用于分枝定界法求最优解.由于该问题是强NP-难的，该文进一步研究它的离线和在线近似算法的设计与分析.文中给出一个离线算法，其绝对性能比为 2;同时给出一个在线算法，其绝对性能比不超过 3，渐近性能比为 2，还对算法性能比的下界进行了讨论.

关键词 装箱问题;算法设计与分析;近似算法;因特网通信;信息管理

中图法分类号 TP393

Bin Packing Problem with Rejection in the Information Organization and Allocation of Internet and Web

HE Yong[1),2)] TAN Zhi-Yi[1),2)] REN Feng[1)]

[1)] (*Department of Mathematics*, *Zhejiang University*, *Hangzhou* 310027)

[2)] (*State Key Laboratory of CAD & CG*, *Zhejiang University*, *Hangzhou* 310027)

Abstract In this paper, we consider the following bin packing problem with rejection. We are given n items, each with a capacity and penalty, and unlimited bins with equal capacity. An item can be either rejected, in which case we pay its penalty, or put in one of the bins, in which case it causes the decreasing of the capacity of a certain bin. The objective is to minimize the sum of the number of the used bins and the penalties of all rejected items. This is a new combinatorial optimization problem which has many applications in practice, such as in organizing and allocating information on web servers. We first provide a lower bound of its optimal value which can be used to solve the problem optimally by branch and bound method. Since the problem is strongly NP-hard, we further consider the design and analysis of its online and offline approximation algorithms. We present an offline algorithm with absolute worst case ratio 2, and an online algorithm with asymptotic worst case ratio 2 and absolute worst case ratio no greater than 3. Moreover we provide lower worst-case bounds of any online and offline algorithms.

Keywords bin packing; the design and analysis of algorithm; approximation algorithm; Internet communication; information management

收稿日期:2002-03-05;修改稿收到日期:2003-07-03.本课题得到高等学校优秀青年教师教学科研奖励计划、国家自然科学基金(10271110)和国家“九七三”重点基础研究发展规划项目(1998030401)资助.**何 勇**，1969 年生，男，博士，教授，博士生导师，主要研究方向为组合与网络优化、计算复杂性、数学建模等. E-mail: mathhey@zju.edu.cn. **谈之奕**，1975 年生，男，博士，副教授，研究方向为组合优化等. **任 峰**，男，1976 年生，硕士，主要研究方向为信息技术中的组合优化问题.

1 引 言

面对内外众多客户对Web服务信息的高频率访问要求,如何有效地对内部网(Intranet)的分布式信息系统进行信息组织和规划,以达到低费用、高效率的目的,是非常重要的关键技术,有许多问题有待解决.文献[1]首先提出这类问题,进行了一定深度的研究,并提出了一个公开问题.文献[2]对此公开问题进行了探讨.本文考虑此类问题中的如下一个新问题:一个企业的专用网即Intranet网,在因特网(Internet)上有两种功能.对外,它主动发布信息,介绍其最新产品和技术,为客户提供服务,在公众面前为企业作宣传等;对内它自身也是Internet用户,要访问内部网以外的各种信息以了解市场,在商业竞争中保持有利地位.在内部发布信息时,将相应的信息主题分成块结构,分布在企业内部不同的网络服务器上,另外企业对外访问是有针对性的,对某些外部信息块的频繁访问会造成通信费用的增长.为了有效地降低通信费用,可以将那些被访问频繁的外部Internet信息块下载至Intranet内部的网络服务器上,使之成为内部信息块.一旦成为内部信息,对它的通信费用可大大降低而访问速度大大提高.但由于网络服务器本身内存的限制及访问该服务器上的信息的速度的要求,企业要有选择地下载Internet外部信息块或在适当的时候购买新的网络服务器以满足需要.企业要考虑的是如何使通信费用以及购买网络服务器的费用的总和最小.

在此问题中,对每个可能有用的外部信息块,把不从Intranet网上将其下载而产生的通信费用作为它的罚值;对内部信息块,由于它一定要放在服务器上,故定义一个充分大的罚值.这样每个信息块都有两个参数:信息块容量,表示其内存大小;信息块罚值,表示不放在服务器上产生的惩罚值.定义每个服务器允许的信息总容量为C,且购买新服务器的费用为F.为方便计,将所有信息块的罚值除以F后作为新的罚值,则购买新服务器的费用为1,从而目标变为购买的服务器数与未放在服务器上的信息块的总罚值之和尽可能的小.一般来说,该问题可描述为下述组合优化问题:

设有长为C的一维箱子,给定物品集$U=\{u_1, u_2, \cdots, u_n\}$,$u_i$长为$\omega_i$($0<\omega_i<C$),罚值为$p_i$,$i=1,2,\cdots,n$.物品可以放入箱子也可被拒绝放入箱子.如果将物品$u_i$放入某箱子,则使该箱剩余长度减少$\omega_i$.一旦要将某一物品放入某一箱中,而该箱的剩余长度不够时,则需启用新箱子.如果某物品u_i被拒绝放入箱中,则产生惩罚p_i.问怎样安排物品使所用箱子数与未装箱的物品总罚值之和最小.我们称此问题为带拒绝装箱问题.

注意到如果令每个物品的惩罚都充分大(例如大于购买新服务器的费用1,注意:当某一物品的罚值大于1时,在最优解中该物品一定不会拒绝),则带拒绝装箱问题就变成经典装箱问题,它是强NP-难的[3],因此带拒绝装箱问题也是强NP-难的.经典装箱问题是组合优化领域的基本问题,已有很多好的结果[4,5],而对带拒绝装箱问题,尚未见文献研究.由于它的NP-难解性,设计与分析其近似算法是重要的研究内容.设A为一个近似算法,用$A(I)$表示算法A解带拒绝装箱问题的实例I所得的目标函数值,$OPT(I)$表示最优值.算法A的绝对性能比定义为$R_A=\sup\limits_I\{A(I)/OPT(I)\}$,渐近性能比定义为$R_A^\infty=\lim\limits_{n\to\infty}\sup\limits_I\{A(I)/OPT(I)\mid OPT(I)\geq n\}$.

根据对物品信息的依赖程度,近似算法分为离线(offline)算法和在线(online)算法.如果在装箱前所有物品的信息(长度、罚值)均已知,并可利用这些信息来设计算法(包括决定哪些物品拒绝以及如何装不拒绝的物品),则称为离线算法.如果物品的信息是一个个释放的,只有在安排好当前物品后,下一个物品的信息才变为已知,且对已安排好的物品不允许改变其安排方案(包括是否拒绝,如不拒绝怎么装),则称为在线算法.

最早研究的此类带拒绝的组合优化问题是可拒绝并行处理器调度问题,它与本文所讨论的问题有一定联系.该问题可描述如下:设$M=\{M_1, M_2, \cdots, M_m\}$为同型并行处理器(identical multiprocessor)集,$J=\{J_1, J_2, \cdots, J_n\}$为相互独立的工件集,其中每个工件$J_i$带有两个参数:工件长度$t_i$和罚值$p_i$.对工件$J_i$,或是加工,则消耗加工时间;或是拒绝,则付出相应的罚值p_i.设在某一调度下被加工工件集为T,被拒绝工件集为R.目标是T中最后一个完工工件的完工时间$C(T)$与被拒绝工件的总罚值$P(R)$之和达到最小.这个问题最早由Bartal等人提出[6],他们对离线和在线算法的研究都得到了很好的结果.He和Min[7]将此问题推广到同类并行处理器(uniform multiprocessor)上的调度问题,对在线算法的设计进行了探讨,而Epstein和Sgall[8]研究了该问题的离线算法.

本文考虑带拒绝装箱问题. 第 2 节给出它的整线性规划描述，由此给出最优解的一个下界；第 3 节讨论离线算法，将给出一个绝对性能比和渐近性能比都为 2 的算法 RFF_1；第 4 节讨论在线算法，将给出绝对性能比至多为 3 且渐近性能比为 2 的算法 RFF_2，并证明任何在线算法的绝对性能比至少为 2；第 5 节进行数值仿真，计算表明，给出的算法在平均意义下是相当好的.

2　整线性规划表示及最优解下界

显然，所需的箱子至多为 n 个. 为了给出整线性规划描述，引入以下变量：

$$y_i=\begin{cases}1, & \text{第 } i \text{ 个箱子被使用}\\ 0, & \text{否则}\end{cases};\quad i=1,2,\cdots,n$$

$$x_{ij}=\begin{cases}1, & \text{第 } j \text{ 个物品放入第 } i \text{ 个箱子}\\ 0, & \text{否则}\end{cases};$$

$$i,j=1,2,\cdots,n.$$

则带拒绝装箱问题(BPR)可表述为

$$\min \quad Z=\sum_{i=1}^{n} y_i+\sum_{j=1}^{n} p_j\left(1-\sum_{i=1}^{n} x_{ij}\right)$$

$$\text{s.t.}\quad \sum_{j=1}^{n}\omega_j x_{ij}\le Cy_i,\ i=1,2,\cdots,n \tag{1}$$

$$\text{BPR}\quad \sum_{i=1}^{n} x_{ij}\le 1,\ j=1,2,\cdots,n \tag{2}$$

$$y_i=0 \text{ 或 } 1,\ i=1,2,\cdots,n \tag{3}$$

$$x_{ij}=0 \text{ 或 } 1,\ i,j=1,2,\cdots,n \tag{4}$$

其中式(1)表示若第 i 个箱子被使用，则装入其中的物品总长度不超过箱子长度；如不使用该箱子，则不能放入任何物品. 式(2)结合式(4)表示每个物品要么被全部装入某个箱子中，要么全部被拒绝.

我们称 Cp_i/ω_i 为物品 u_i 的罚值密度，据此我们将物品分成两大类：

$$M_1=\{i\mid Cp_i/\omega_i>1, i=1,2,\cdots,n\}$$

罚值密度大于 1 的物品集；

$$M_2=\{i\mid Cp_i/\omega_i\le 1, i=1,2,\cdots,n\}$$

罚值密度不大于 1 的物品集.

对 BPR 进行连续松弛，即把 BPR 中的式(3)，(4)分别松弛为 $0\le y_i\le 1$，$i=1,2,\cdots,n$ 及 $0\le x_{ij}\le 1$，$i,j=1,2,\cdots,n$，记为(3)′，(4)′，而其余不变，就得到一个线性规划问题，记为 C(BPR).

定理 1.　C(BPR)的最优解为：

(i) 对 $k\in M_1$，有 $\sum_{i=1}^{n}x_{ik}=1$；

(ii) 对 $k\in M_2$，有 $\sum_{i=1}^{n}x_{ik}=0$；

(iii) $y_i=\sum_{j=1}^{n}\omega_j x_{ij}/C$，$i=1,2,\cdots,n$；

(iv) 最优解值 $Z(C(\text{BPR}))=\sum_{j\in M_1}\frac{\omega_j}{C}+\sum_{j\in M_2}p_j$.

证明.　(i) 对某个 $k\in M_1$，如果在最优解中有 $\sum_{i=1}^{n}x_{ik}<1$，则一定存在 q，$1\le q\le n$，使该箱子未装满，也即 $\sum_{j=1}^{n}\omega_j x_{qj}\le Cy_q$，$y_q\le 1$，且至少有一个为严格不等式. 注意到由于每个 $\omega_i<C$，所以即使所有物品全部转入箱中，必有某些箱子未满，因此这样的 q 是存在的. 我们取充分小的 $\varepsilon>0$，使得①若有 $\sum_{j=1}^{n}\omega_j x_{qj}<Cy_q$ 则将 x_{qk} 增加 ε，其余不变，记新解为 x'_{ij}，y'_i，则当 ε 充分小时，可保证 $\sum_{j=1}^{n}\omega_j x'_{qj}\le Cy'_q$ 及其它所有约束成立，因此得到的新解仍是可行解；②若有 $\sum_{j=1}^{n}\omega_j x_{qj}=Cy_q$ 但 $y_q<1$，则将 x_{qk} 增加 ε，将 y_q 增加 $w_k\varepsilon/C$，其余不变. 记新解为 x'_{ij}，y'_i，则当 ε 充分小时，仍保证 $\sum_{j=1}^{n}\omega_j x'_{qj}\le Cy'_q$ 和 $y'_q\le 1$ 成立，而且其它所有约束也成立，因此得到的新解仍是可行. 目标函数值改变了 $\Delta\le\varepsilon\frac{\omega_k}{C}-p_k\varepsilon=\varepsilon\frac{\omega_k}{C}\left(1-\frac{Cp_k}{\omega_k}\right)<0$，矛盾.

(ii) 与(i)同理可证.

(iii) 由式(1)得 $y_i\ge\sum_{j=1}^{n}\omega_j x_{ij}/C$，由于在目标函数式中 $\sum_{i=1}^{n}y_i$ 越小越好，所以在最优解中有

$$y_i=\sum_{j=1}^{n}\omega_j x_{ij}/C.$$

(iv) 将以上解代入目标函数表达式，直接计算即得结果.　　证毕.

例如，由上可知

$$y_i=\begin{cases}\omega_i/C, & i\in M_1,\\ 0, & i\in M_2;\end{cases}$$

$$x_{ij}=\begin{cases}1, & j\in M_1, i=j,\\ 0, & \text{否则}\end{cases}$$

是松弛问题 C(BPR)的一个最优解，它表示 M_1 中物品全部装箱，M_2 中物品全部拒绝. 后面给出的近似算法就是受此启发而设计的.

记下述装箱问题的实例为 I'：设含 n' 个物品的

物品集为 M,物品长分别为 $\omega_i, i\in M$,箱子长为 C.用 $OPT'(I')$表示对 I'的最优装箱数. 算法 FF 是装箱问题的一个近似算法,它可描述为:将当前待装箱的物品放入下标最小且剩余容量足够的箱子. 我们有下面引理.

引理 1[9]. (1)设 $OPT'(I')$为装箱问题实例 I'的最优解值,则 $\sum_{i=1}^{n'}\frac{\omega_i}{C}>\frac{1}{2}(OPT'(I')-1)$;(2) $FF(I')\le 2\sum_{i=1}^{n'}\frac{\omega_i}{C}$.

至此我们可以得到带拒绝装箱问题最优解值的一个下界,它可用于分枝定界法求解带拒绝装箱问题的最优解.

定理 2. 对任意带拒绝装箱问题实例 I, $L=\sum_{i\in M_1}\frac{\omega_i}{C}+P(M_2)$是其最优解值 $OPT(I)$的下界,即有 $OPT(I)\ge L$;且成立 $L\ge\frac{1}{2}(OPT(I)-1)$. 这里 $P(M_2)=\sum_{i\in M_2}p_i$为 M_2中物品的总罚值.

证明. 由定理 1 即知 L 是最优解值的一个下界. 设 I_1'表示物品集为 M_1的装箱问题的实例. 将 I_1'按最优装箱,且 M_2中物品全部拒绝,则得到带拒绝装箱问题实例 I 的一个可行解. 该解的目标函数值为 $OPT'(I_1')+P(M_2)$,因此有 $OPT'(I'_1)+P(M_2)\ge OPT(I)$,也即

$$OPT'(I_1')\ge OPT(I)-P(M_2) \qquad (5)$$

由引理 1 中的(1)得

$$\sum_{i\in M_1}\frac{\omega_i}{C}>\frac{1}{2}(OPT'(I'_1)-1) \qquad (6)$$

由式(5) 和式(6)可得

$$\sum_{i\in M_1}\frac{\omega_i}{C}>\frac{1}{2}(OPT(I)-P(M_2)-1) \qquad (7)$$

将式(7)两边加上 $P(M_2)$得

$$\begin{aligned}L&=\sum_{i\in M_1}\frac{\omega_i}{C}+P(M_2)\\&>\frac{1}{2}(OPT(I)-P(M_2)-1)+P(M_2)\\&\ge\frac{1}{2}(OPT(I)-1)\end{aligned}$$

证毕.

3 离线近似算法的设计与分析

算法 RFF_1.

1. 当$\sum_{i=1}^{n}\omega_i<\frac{1}{2}C$ 时,如果$\sum_{i=1}^{n}p_i>1$,则将所有物品装箱;否则$\sum_{i=1}^{n}p_i\le 1$,则拒绝所有物品.

2. 当$\sum_{i=1}^{n}\omega_i\ge\frac{1}{2}C$ 时,对于物品 u_i,如果 $i\in M_1$,则用 FF 算法将该物品装箱;否则拒绝该物品.

定理 3. 对带拒绝装箱问题的所有实例 I 成立 $RFF_1(I)\le 2OPT(I)$,该界是不可改进的.

证明. 当$\sum_{i=1}^{n}\omega_i<C/2$ 时,根据算法规则知:① 如果$\sum_{i=1}^{n}p_i>1$,则 $RFF_1(I)=OPT(I)=1$;② 如果$\sum_{i=1}^{n}p_i\le 1$,则仍有 $RFF_1(I)=OPT(I)=\sum_{i=1}^{n}p_i$. 因此总得到最优解,显然成立 $RFF_1(I)\le 2OPT(I)$.

当 $\sum_{i=1}^{n}\omega_i\ge C/2$ 时,由引理 1 中的(2),我们有 $FF(M_1)\le 2\sum_{i\in M_1}\omega_i/C$. 又 $RFF_1(I)=FF(M_1)+P(M_2)$,所以结合定理 2 知

$$\begin{aligned}RFF_1(I)&\le 2\sum_{i\in M_1}\frac{\omega_i}{C}+P(M_2)\\&\le 2\left(\sum_{i\in M_1}\frac{\omega_i}{C}+P(M_2)\right)\le 2OPT(I).\end{aligned}$$

为证明该界是不可改进的,考虑以下的实例:对任意自然数 N,取充分小正数 ε,令物品集$\{u_1,u_2,\cdots,u_{2N}\}=\{1/2+\varepsilon,1/2+\varepsilon,\cdots,1/2+\varepsilon\}$,每个物品的罚值都为 $1/2+2\varepsilon$, $C=1$. 则所有物品都属于 M_1, RFF_1算法需用 $2N$ 个箱子,而 $OPT=N+4N\varepsilon$,故 $RFF_1/OPT=2N/(N+4N\varepsilon)\to 2(\varepsilon\to 0)$. 证毕.

由定理 3 及证明中给出的实例可知 $R_{RFF_1}=R_{RFF_1}^{\infty}=2$. 算法 RFF_1的时间复杂性为 $O(n\log n)$. 由该实例还可看出,即使在执行算法 RFF_1前先将所有物品按它们的容量从大到小整序,其性能比也不会好于 2,这与经典装箱问题明显不同[9].

由于装箱问题不存在绝对性能比小于 3/2 的多项式时间离线算法,因此带拒绝装箱问题也不存在这样的算法.

4 在线近似算法的设计与分析

本节我们给出一个时间复杂性为 $O(n\log n)$的在线算法 RFF_2,并讨论在线算法的近似程度的下界.

算法 RFF_2.

1. 物品 u_1到达,如果 $p_1<1$ 则拒绝,转步 2;否则转步 3.
2. 物品 u_i到达,设已到达物品的总罚值为 P,如果 $P+$

$p_i<1$，则拒绝 u_i，否则转步 3.

3. 自当前物品 u_i 开始，所有物品按下述规则决定是否拒绝以及如何装箱：如果该物品属于 M_2，则拒绝；如果该物品属于 M_1，则按 FF 算法将该物品装箱.

定理 4. 对于带拒绝装箱问题的所有实例 I，成立 $RFF_2(I)\le 3OPT(I)$，$RFF_2(I)\le 2OPT(I)+1$.

证明. 如果所有物品的总罚值小于 1，则显然算法 RFF_2 得到问题最优解. 故以下假定所有物品的总罚值至少为 1，因此我们一定有 $OPT(I)\ge 1$. 令 I_1 为算法 RFF_2 中按步 3 安排的物品集组成的实例. 由于算法 RFF_2 步 1～步 2 中被拒绝物品的总罚值不超过 1，且未将任何物品装箱，所以 $RFF_2(I\backslash I_1)\le 1$. 下面就 I_1 中物品总长度分两种情况考虑：

① 如果 I_1 中物品总长度小于 $C/2$，按 M_2 的定义知，I_1 中被拒绝物品的总罚值不超过它们的总长度除以 C，因此总罚值也不超过 1/2，且未被拒绝的物品总长度仍小于 $C/2$，因此 FF 算法至多用了一个箱子，所以我们知 $RFF_2(I_1)\le 3/2$. 因此有 $RFF_2(I)=RFF_2(I\backslash I_1)+RFF_2(I_1)\le 1+RFF_2(I_1)=5/2$. 由于 $1\le OPT(I)$，显然有 $RFF_2(I)\le 5OPT(I)/2$ 及 $RFF_2(I)\le 2OPT(I)+1$.

② 如果 I_1 中物品总长度至少为 $C/2$，同定理 4 证明我们有 $RFF_2(I_1)\le 2OPT(I_1)$，因此有 $RFF_2(I)=RFF_2(I\backslash I_1)+RFF_2(I_1)\le 1+RFF_2(I_1)\le 1+2OPT(I_1)\le 1+2OPT(I)\le 3OPT(I)$. 证毕.

由定理 5 知算法 RFF_2 的绝对性能比至多是 3，渐近性能比是 2(定理 4 中的例子说明 2 是不可改进的). 算法 RFF_2 的绝对性能比是否严格好于 3 未知.

定理 5. 对带拒绝装箱问题，不存在在线算法，其绝对性能比小于 2.

证明. 用对手法证明. 为此考虑一系列实例 I，其中每个物品长度都为 $1/n$，每个物品的罚值均为 $2/n$. 这里 n 为正整数且 $n\ge 6$. 令 $A(I)$ 为任一在线算法 A 的解值. 首先第一个物品 u_1 到达，我们分两种情况考虑：

若算法将 u_1 装箱则不再来任何物品，因此 $A(I)=1$，而易验证 $OPT(I)=2/n$，所以 $A(I)/OPT(I)=n/2$，由于 $n\ge 6$，有 $A(I)/OPT(I)\ge 3$. 若算法拒绝 u_1，考虑再来一个物品 u_2，同样继续分两种情况考虑：

若算法将 u_2 装箱则不再来任何物品，此时有 $A(I)=1+2/n$，易验证 $OPT(I)=4/n$，所以 $A(I)/OPT(I)=n/4+1/2$，由于 $n\ge 6$，有 $A(I)/OPT(I)\ge 2$. 若算法拒绝 u_2，考虑再来一个物品 u_3，…，重复此过程. 一般地，当物品 u_k 到达时，分两种情况考虑：

①将物品 u_k 装箱. 则不再来任何物品，则有 $A(I)=1+(k-1)\dfrac{2}{n}$. 易知当 $k\le\dfrac{n}{2}$ 时 $OPT(I)=\dfrac{2k}{n}$，当 $k>\dfrac{n}{2}$ 时，$OPT(I)=1$. 因此如果 $k\le\dfrac{n}{2}$，则有

$$\frac{A(I)}{OPT(I)}=\frac{1+(k-1)2/n}{2k/n}=\frac{1}{k}\left(\frac{n}{2}-1\right)+1$$

$$\ge\frac{2}{n}\left(\frac{n}{2}-1\right)+1\to 2(n\to\infty);$$

如果 $k>\dfrac{n}{2}$，则有

$$\frac{A(I)}{OPT(I)}=1+(k-1)\frac{2}{n}$$

$$>1+\frac{2}{n}\left(\frac{n}{2}-1\right)\to 2(n\to\infty).$$

② 拒绝 u_k. 则再来一个物品 u_{k+1}. 重复以上分析，…，如此直到第 n 个物品 u_n 到达. 若将 u_n 装箱，则 $A(I)=1+2(n-1)/n$，$OPT(I)=1$. 我们得

$$\frac{A(I)}{OPT(I)}=1+(n-1)\frac{2}{n}=3-\frac{2}{n}\to 3\ (n\to\infty).$$

如果拒绝 u_n，则有 $A(I)=2$，$OPT(I)=1$. 因此 $A(I)/OPT(I)=2$.

综上所述无论什么算法，总存在实例使得 $A(I)/OPT(I)\to 2$(当 $n\to\infty$ 时). 因此，我们证明了不存在在线算法，其绝对性能比小于 2. 证毕.

5 数值模拟

本节我们用计算机对不同算法进行模拟，比较它们在平均意义下的性能比. 我们在配置为 C500，128M 的计算机上进行. 我们用解整数线性规划 BPR 的分枝定界法(深探法)求出最优解，这里第 2 节给出的最优解下界可用来定界. 我们用近似算法 RFF_1，RFF_2 的解值与最优解值的比值来衡量它们对最优解的近似程度. 分别取物品个数 $n=20$，30，50，100 和箱子长度 $C=100$，300，500. 物品长度为区间[1, C] 中随机产生的整数，罚值为区间[0,2] 中随机产生的实数. 对每一种组合，随机产生 50 个实例. 求出 50 个比值的平均值，结果见表 1. 由表 1 我们可知近似算法在大部分情况下能得到较好的近似解. 因此在平均意义下这些算法都是非常有效的：一方面算法运算速度快(它们所用时间均不超过 0.1s)，另一方面算法的解非常接近最优解.

表 1 用近似算法求得的解

n	C	$\frac{RFF_1(I)}{OPT(I)}$	$\frac{RFF_2(I)}{OPT(I)}$
20	100	1.1594	1.2417
20	300	1.1597	1.2257
20	500	1.1457	1.2148
30	100	1.1326	1.1729
30	300	1.1293	1.1779
30	500	1.1303	1.1715
50	100	1.1133	1.1445
50	300	1.1123	1.1315
50	500	1.1042	1.1275
100	100	1.0793	1.0968
100	300	1.0794	1.0930
100	500	1.0822	1.1006

6 结束语

本文从内联网的分布式信息组织和规划问题中提出了一个新的组合优化问题——极小化所用箱子数与总罚值之和的带拒绝装箱问题,就在线与离线分别讨论了近似算法的设计与分析.我们提出了两个近似算法,证明了它们的绝对性能比或渐近性能比,通过数值仿真试验知道在平均意义下这些算法的效果相当好.一个值得进一步深入研究的问题是:能否设计出绝对性能比或渐近性能比更好的算法?对此,本文给出了两个下界估计:即不存在绝对性能比好于 3/2 的多项式时间离线算法;不存在绝对性能比好于 2 的在线算法.这两个结果对此问题的讨论有一定的帮助.另外从文中算法的性能比分析可知对接受物品采用不同的方法装箱不会提高算法性能比,要得到最坏情况意义下更好的算法似乎只能从如何决定哪些物品要装,哪些要拒绝入手.

参 考 文 献

1 Chen W D, Yang J J, Lu D M, Pan Y H. Two mathematical models and algorithms of internet communications. Chinese Journal of Computers, 1999, 22(1): 51~55(in Chinese)
(陈卫东,杨建军,鲁东明,潘云鹤.互联网通信中的两个数学模型及求解.计算机学报,1999,22(1):51~55)

2 He Y. The modeling and algorithms of information organizing and allocating problem on internet communications. Chinese Journal of Computers, 2001,24(6): 596~601(in Chinese)
(何 勇.互联网通信中的信息选取与分布问题的建模与求解.计算机学报,2001,24(6):596~601)

3 Garey M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York: Freeman, 1979

4 Csirik J, Woeginger G. On-line packing and covering problems. In:Lecture Notes in Computer Science 1442, Berlin: Springer, 1998. 147~177

5 Coffman E G, Garey M R, Johnson D S. Approximation algorithms for bin packing: A survey. In:Hochbaum D ed. Approximation algorithms for NP-hard problems. Boston: PWS Publishing, 1997. 46~93

6 Batal Y, Leonardi S. Marchetti-Spaccamela A *et al*. Multiprocessor scheduling with rejection. SIAM Journal of Discrete Mathematics, 2000, 13(1): 64~78

7 He Y, Min X. On-line uniform machine scheduling with rejection. Computing, 2000, 65(1): 1~12

8 Epstein L, Sgall J. Approximation schemes for scheduling on uniformly related and identical parallel machine. In:Lecture Notes in Computer Science 1643, Berlin: Springer, 1999. 151~162

9 Baase S. Computer Algorithms: Introduction to Design and Analysis: Reading: Addison-Wesley, 1994

HE Yong, born in 1969, Ph. D., professor, Ph. D. supervisor. His current interests include combinatorial and network optimization, computational complexity and mathematical modeling, etc.

TAN Zhi-Yi, born in 1975, Ph. D. and associate professor. His current interests include combinatorial and network optimization.

REN Feng, born in 1976, master. His current interests include combinatorial optimization in information technology.