SOLVING PACKING PROBLEM WITH WEAKER BLOCK SOLVERS *

Hu Zhang
Institute of Computer Science and Applied Mathematics
University of Kiel, Germany
hzh@informatik.uni-kiel.de

Abstract We study the general packing problem with M constraints. In [Jansen and Zhang, TCS 2002] a $c(1+\varepsilon)$-approximation algorithm for the general packing problem was proposed. A block solver $ABS(p, \varepsilon/6, c)$ with price vector p, given accuracy ε and ratio c is required. In addition, in [Villavicencio and Grigoriadis, Network Optimization (1997)] a $(1+\varepsilon)$-approximation algorithm for standard packing problem and its dual problem was studied, with a block solver $ABS(p, \varepsilon/10)$ (i.e., $c = 1$). In this paper we develop $c(1+\varepsilon)$-approximation algorithms for the general packing problem (or with its dual problem), with only weaker block solvers $ABS(p, O(\varepsilon'), c)$ with same structure as in previous algorithms, where $\varepsilon' > \varepsilon$. For both primal and dual problems we design an algorithm with an $ABS(p, \varepsilon_1/10, c)$ and $\varepsilon_1 > \varepsilon$. The bound on the number of iterations is polynomial in M, ε and c. Furthermore we show an algorithm for the primal problem with an $ABS(p, \varepsilon_3/6, c)$ and $\varepsilon_3 > \varepsilon$. And the bound on the number of iterations is polynomial in only M and ε. In both cases running times are further improved with corresponding weaker block solvers. This is the first attempt to solve the packing problem with weaker block solvers.

1. Introduction

An interesting class of optimization problems is the *packing problem* or *convex min-max resource-sharing problem* defined as follows:

$$(P) \qquad \lambda^* = \min\{\lambda | f(x) \leq \lambda e, x \in B\},$$

*This research was supported in part by the DFG Graduiertenkolleg 357, Effiziente Algorithmen und Mehrskalenmethoden, by EU Thematic Network APPOL, Approximation and Online Algorithms for Optimization Problems, IST-2001-32007, and by EU Project CRESCCO, Critical Resource Sharing for Cooperation in Complex Systems, IST-2001-33135.

where $f : B \rightarrow \mathbb{R}_+^M$ is a vector of M continuous convex functions defined on a nonempty convex set $B \subseteq \mathbb{R}^N$, and e is the vector of all ones. Without loss of generality we can assume $\lambda^* > 0$. The functions f_m, $1 \leq m \leq M$, are packing constraints. In addition, we denote by $\lambda(x) = \max_{1 \leq m \leq M} f_m(x)$ for any given $x \in B$. There are many applications of the packing problem. Typical examples include scheduling on unrelated machines, job shop scheduling, network embeddings, Held-Karp bound for TSP, minimum-cost multicommodity flows, maximum concurrent flow, bin covering, spreading metrics, approximating metric space, graph partitioning, multicast congestion in communication networks, and energy consumption problem in ad-hoc networks on general metric spaces [1–4, 7, 9, 13, 17, 19, 22].

The problem (P) could be solved exactly in polynomial time in its size usually. However, in some cases an approximate solution is enough (e.g. [16]). In addition, it is possible that the size of (P) is exponential in the size of input (e.g. [1, 15, 19]). Thus we consider fast but approximation algorithms for problem (P). Given an accuracy tolerance $\varepsilon > 0$, the approximate problem is as follows:

$$(P_\varepsilon) \qquad \text{compute } x \in B \text{ such that } f(x) \leq (1+\varepsilon)\lambda^* e.$$

Grigoriadis et al. [5, 6] proposed algorithms for the above problem based on the Lagrangian duality relation $\lambda^* = \min_{x \in B} \max_{p \in P} p^T f(x) = \max_{p \in P} \min_{x \in B} p^T f(x)$, where $P = \{p \in \mathbb{R}^M | \sum_{m=1}^M p_m = 1, p_m \geq 0\}$. Denoting by $\Lambda(p) = \min_{x \in B} p^T f(x)$, it can be verified that a pair $x \in B$ and $p \in P$ is optimal if and only if $\lambda(x) = \Lambda(p)$. On the other hand, the corresponding approximate dual problem is:

$$(D_\varepsilon) \qquad \text{compute } p \in P \text{ such that } \Lambda(p) \geq (1-\varepsilon)\lambda^*.$$

In addition, the Lagrangian or price-directive decomposition method is applied in their algorithms, which is an iterative approach that solves (P_ε) and (D_ε) by computing a sequence of pairs x and p to approximate the exact solution from above and below, respectively. Grigoriadis and Khachiyan [6] proved that (P_ε) and (D_ε) can be solved in $O(M(\ln M + \varepsilon^{-2} \ln \varepsilon^{-1}))$ iterations or calls to a standard approximate block solver $ABS(p,t)$ that solves the block problem for a given tolerance $t = O(\varepsilon)$: to compute $\hat{x} = \hat{x}(p) \in B$ such that $p^T f(\hat{x}) \leq (1+t) \min_{y \in B} p^T f(y)$. Villavicencio and Grigoriadis [18] proposed a modified logarithmic potential function to avoid the ternary search and the number of iterations is also $O(M(\ln M + \varepsilon^{-2} \ln \varepsilon^{-1}))$. In [14] the bound was improved to $O(M(\ln M + \varepsilon^{-2}))$ for both (P_ε) and (D_ε).

However, in general the block problem may be hard to approximate [1–3, 15, 19], i.e., the assumption to have a block solver with accuracy

$t = O(\varepsilon)$ is too strict. Therefore in [14] the authors considered the case that only a weak approximate block solver $ABS(p,t,c)$ is available, which is defined to compute $\hat{x} = \hat{x}(p) \in B$ such that $p^T f(\hat{x}) \leq c(1+t)\min_{y\in B} p^T f(y)$, where $c \geq 1$ is the approximation ratio. The main goal is to solve the following primal problem (using the weak block solver):

$$(P_{\varepsilon,c}) \qquad \text{compute } x \in B \text{ such that } f(x) \leq c(1+\varepsilon)\lambda^* e.$$

And the corresponding dual problem is:

$$(D_{\varepsilon,c}) \qquad \text{compute } p \in P \text{ such that } \Lambda(p) \geq \frac{1}{c}(1-\varepsilon)\lambda^*.$$

Jansen et al. [14] developed an approximation algorithm that for any accuracy $\varepsilon \in (0,1]$ solves the $(P_{\varepsilon,c})$ in $O(M(\ln M + \varepsilon^{-2}\ln\varepsilon^{-1}))$ iterations by adding a new stopping rule. Each step calls the weak block solver $ABS(p, O(\varepsilon), c)$ once and has an overhead of $O(M \ln\ln(M\varepsilon^{-1}))$ arithmetic operations. In addition, for small ratio c with $\ln c = O(\varepsilon)$ they improved the bound to $O(M(\ln M + \varepsilon^{-2}))$.

Related results: Plotkin et al. [17] considered the linear feasibility variants of both problems: either to find a point $x \in B$ such that $f(x) = Ax \geq (1-\varepsilon)b$ or to find a point $x \in B$ such that $f(x) = Ax \leq (1+\varepsilon)b$ where A is the coefficient matrix with M rows and b is an M-dimensional vector. The problems are solved by Lagrangian decomposition with exponential potential reductions. The numbers of iterations in these algorithms are $O(\varepsilon^{-2}\rho\ln(M\varepsilon^{-1}))$ and $O(M + \rho\ln^2 M + \varepsilon^{-2}\rho\ln(M\varepsilon^{-1}))$ respectively, where $\rho = \max_{1\leq m\leq M}\max_{x\in B} a_m^T x/b_m$ is the width of B relative to $Ax \geq b$. However, their algorithms could have only pseudo polynomial running time due to the parameter ρ. Garg and Könemann [4] proposed a $(1+\varepsilon)$-approximation algorithm for the linear packing problem within $O(M\varepsilon^{-2}\ln M)$ iterations, which is independent of the width. Unfortunately implementation results show that their algorithm is very impractical [1]. Young [20] studied also the linear case of the packing problem but weak block solvers are allowed. His algorithm uses $O(\rho'(\lambda^*)^{-1}\varepsilon^{-2}\ln M)$ calls to the block solver, where $\rho' = \max_{1\leq m\leq M}\max_{x\in B} a_m^T x/b_m - \min_{1\leq m\leq M}\min_{x\in B} a_m^T x/b_m$ and λ^* is the optimal value of the packing problem. Similar to [17], this result also depends on input data. Furthermore, Charikar et al. [2] noticed that the result in [17] for the packing problem can be extended also to the case of weak block solvers in the same number $O(\varepsilon^{-2}\rho\ln(M\varepsilon^{-1}))$ of iterations. For the covering problem, which is related to the packing problem, Grigoriadis et al [8] showed that the number of iterations is $O(M(\ln M + \varepsilon^{-2}))$ with $c = 1$. Jansen and Porkolab [12] studied the general covering problem with only weak approximate block solver and showed that at most

$O(M(\ln M + \varepsilon^{-2} + \varepsilon^{-3} \ln c))$ coordination steps are necessary. Recently Jansen [10] improved the bound to $O(M(\ln M + \varepsilon^{-2} \ln \varepsilon^{-1}))$, which is same as the bound in [14] for the packing problem. In addition, Young [21] proposed an approximation algorithm for a mixed linear packing and covering problem with the number of iterations $O(Md\varepsilon^{-2} \ln M)$, where d is the maximum number of constraints any variable appears in. Jansen [11] further improved the bound to $O(M\varepsilon^{-2} \ln(M\varepsilon^{-1}))$, which is also the first result independent of data for the mixed problem.

Our contribution: We notice that in [18] a $(1+\varepsilon)$-approximate solution for (P_ε) and (D_ε) can be obtained with a $(1+\varepsilon/10)$-approximate block solver, while a $c(1+\varepsilon)$-approximate solution for $(P_{\varepsilon,c})$ with a $c(1+\varepsilon/6)$-approximate block solver as well. Thus it is an interesting problem to solve $(P_{\varepsilon,c})$ with either a $c(1+\varepsilon'/10)$- or a $c(1+\varepsilon'/6)$-approximate block solver, where $\varepsilon \leq \varepsilon' \leq 1$. In this paper we develop algorithms for the above problems. First, we show a $c(1+\varepsilon)$-approximation algorithm $\mathcal{A}$ for both $(P_{\varepsilon,c})$ and $(D_{\varepsilon,c})$, with only an $ABS(p, \varepsilon_1/10, c)$, where $\varepsilon_1 = 8\varepsilon/(7-\varepsilon) > 8\varepsilon/7$ and the number of iteration is bounded by $O(Mc^2(\ln M + \varepsilon^{-4} \ln c + \varepsilon^{-3} \ln \varepsilon^{-1}))$. We then improve the coordination complexity to $O(Mc^2(\ln M + \varepsilon^{-3} \ln c + \varepsilon^{-2} \ln \varepsilon^{-1}))$ in $\mathcal{A}'$, but an $ABS(p, \varepsilon_2/10, c)$ is required, where $\varepsilon_2 = 8\varepsilon/((7+8r)-(1-8r)\varepsilon) \geq 8\varepsilon/(7+8r)$ for a constant $r \in (0, 1/8)$. Notice here $\varepsilon_2 \in (\varepsilon, \varepsilon_1)$. Furthermore, for only $(P_{\varepsilon,c})$ we develop a $c(1+\varepsilon)$-approximation algorithm $\mathcal{F}$ with an $ABS(p, \varepsilon_3/6, c)$, with $\varepsilon_3 = (43 - \sqrt{1849 - 1176\varepsilon})/12 \geq 49\varepsilon/43$. And the bound on the number of iterations is $O(M(\ln M + \varepsilon^{-4} \ln \varepsilon^{-1}))$, which is also improved to $O(M(\ln M + \varepsilon^{-2} \ln \varepsilon^{-1}))$ in $\mathcal{F}'$ with an $ABS(p, \varepsilon_4/6, c)$ for an $\varepsilon_4 \in (\varepsilon, \varepsilon_3)$. These algorithms are the first ones to solve general packing problems with only weaker block solvers.

Applications: One application is the case that the block problem only has an algorithm with a running time depending on input value power to a function of ε^{-1}, for instance, $O(n^{1/\varepsilon})$. In this case, to solve both problem $(P_{\varepsilon,c})$ and $(D_{\varepsilon,c})$, we just need a $c(1+\varepsilon'/10)$-approximate block solver, or a $c(1+\varepsilon'/6)$-approximate block solver to solve only $(P_{\varepsilon,c})$. Thus the running time of ABS can be reduced and the overall running time, which is the product of the bound on number of iterations and the running time of ABS, can decrease considerably. Another very interesting case is that the requirement of approximation ratio to block solver is too strict, i.e., only a $c'(1+\varepsilon/10)$- or $c'(1+\varepsilon/6)$-approximate block solver exists, where $c' > c$. For this case, if $c' \leq c(1+\varepsilon'/10)/(1+\varepsilon/10)$ or $c' \leq c(1+\varepsilon'/6)/(1+\varepsilon/6)$, respectively, we are able to also use our algorithms to solve the instance.

The paper is organized as follows: In Section 2 the potential function, price vector and their properties are reviewed. The algorithm $\mathcal{A}$ is presented in Section 3, as well as algorithm $\mathcal{F}$ in Section 4. Improvements of running times are also addressed. Due to the limit of space, proofs are not given in this version.

2. Modified logarithmic potential function

We use potential function to relax the packing constraints and show that an approximation of the minimum value of potential function corresponds to an approximation of λ^*. Thus the original problem can be replaced by finding a good approximate minimum point of the (smooth) potential function. The modified potential function is defined as follows [14, 18]:

$$\Phi_t(\theta, x) = \ln \theta - \frac{t}{M} \sum_{m=1}^{M} \ln(\theta - f_m(x)), \tag{1}$$

where $\theta \in \mathbb{R}_+$ and $x \in B$ are variables, and $t \in (0, 1]$ is a given tolerance parameter, which is also used in the approximate block solver $ABS(p, t, c)$. Same as [14, 18], in our algorithm, the values of t will be from $O(1)$ down to $O(\varepsilon)$, where ε is the desired accuracy tolerance. Since $\lambda(x) < \theta < \infty$ where $\lambda(x) = \max\{f_1(x), \ldots, f_M(x)\}$, the function Φ_t is well-defined. In addition, it has the *barrier property*: $\Phi_t(\theta, x) \to \infty$ for $\theta \to \lambda(x)$ and $\theta \to \infty$.

The reduced potential function is defined as the minimum of $\Phi_t(\theta, x)$ over $\theta \in (\lambda(x), \infty)$ for a fixed $x \in B$, i.e.

$$\phi_t(x) = \min_{\lambda(x) < \theta < \infty} \Phi_t(\theta, x). \tag{2}$$

It can be proved that $\theta(x)$ is the solution to the following equation:

$$\frac{t}{M} \sum_{m=1}^{M} \frac{\theta}{\theta - f_m(x)} = 1. \tag{3}$$

The function $g(\theta) = (t/M) \sum_{m=1}^{M} \theta/(\theta - f_m)$ is strictly decreasing in θ in $(\lambda(x), \infty)$. Therefore the implicit function $\theta(x)$ is the unique root of (3) in the interval $(\lambda(x), \infty)$. $\theta(x)$ and $\phi_t(x)$ are bounded by the following lemmas, same as [14, 18].

LEMMA 1 $\lambda(x)/(1 - t/M) \leq \theta(x) \leq \lambda(x)/(1 - t)$ *for any* $x \in B$.

LEMMA 2 $(1 - t) \ln \lambda(x) \leq \phi_t(x) \leq (1 - t) \ln \lambda(x) + t \ln(\exp(1)/t)$ *for any* $x \in B$.

The price vector $p(x) \in \mathbb{R}^M$ is defined as follows [14, 18]:

$$p_m(x) = \frac{t}{M} \frac{\theta(x)}{\theta(x) - f_m(x)}, \qquad m = 1, \ldots, M. \tag{4}$$

The following lemma holds:

LEMMA 3 *$p(x) \in P$ and $p(x)^T f(x) = \theta(x)(1-t)$ for any $x \in B$.*

3. Approximation algorithm $\mathcal{A}$

In this section we will study the algorithm $\mathcal{A}$, which solves both primal problem $(P_{\varepsilon,c})$ and dual problem $(D_{\varepsilon,c})$ with a weaker block solver $ABS(p, \varepsilon_1/10, c)$, where $\varepsilon_1 = 8\varepsilon/(7-\varepsilon) > 8\varepsilon/7$. Compared with that in [18], the block solver employed here is weaker.

The algorithm works as follows. We apply the scaling phase strategy. In each scaling phase an error tolerance σ is set. Based on the known pair of x and p, a solution $\hat{x}$ is delivered by the approximate block solver. Afterwards an appropriate linear combination of the old solution x and block solution $\hat{x}$ is computed as the new iterate. The iteration stops when the solution satisfies a stopping rule (defined later). After one scaling phase, the error tolerance σ is halved and the next scaling phase starts until the error tolerance $\sigma \leq \varepsilon$. The pair x and p generated by the last scaling phase solves both $(P_{\varepsilon,c})$ and $(D_{\varepsilon,c})$ (see Subsection 3.1).

The minimum dual value $\Lambda(p)$ can be approximated by $p^T f(\hat{x})$, where $\hat{x}$ is the solution computed by the weak approximate block solver for the current price vector p. Furthermore, to establish the stopping rule of the scaling phase in algorithm $\mathcal{A}$, the value of duality gap should be estimated in each iteration. Thus we define the stopping rule as follows:

$$(1 + \sigma'/4)p^T f(x) - p^T f(\hat{x}) \leq \sigma' \theta(x)/2, \tag{5}$$

where $\sigma' = 8\sigma/(7-\sigma) > 8\sigma/7$. This stopping rule is similar to that in [18] (which is only valid for a standard block solver), but with an additive term in the left hand side and σ is replaced by σ'. Only in this way can we obtain the desired solution with a weaker block solver. In addition, the stopping rule (5) is also different from the first one in [14]. We set $t = \sigma'/10$ for the error tolerance in the block solver $ABS(p, t, c)$. To run the algorithm, we need an initial solution $x_0 \in B$. Here we use the solution of the block solver $ABS(e/M, \sigma/10, c)$ as x_0, where the price vector e/M is the vector of all $1/M$'s and the initial error tolerance $\sigma = 1/2$.

Algorithm $\mathcal{A}(f, B, \varepsilon, c)$:

```
initialize: $\sigma := 1/2$, $\sigma' := 8\sigma/(7-\sigma)$, $t := \sigma'/10$, $p := e/M$;
$x = x_0 := ABS(p,t,c)$, $finished\_scaling := false$;
while $not(finished\_scaling)$ do {$scaling\ phase$}
  $\sigma := \sigma/2$, $\sigma' := 8\sigma/(7-\sigma)$, $t := \sigma'/10$;
  $finished\_coordination := false$;
  while $not(finished\_coordination)$ do {$coordination\ step$}
    compute $\theta(x)$ from (3) and $p = p(x) \in P$ from (4);
    $\hat{x} := ABS(p,t,c)$;
    if (5) then
      $finished\_coordination := true$;
    else
      $x := (1-\tau)x + \tau\hat{x}$ for an appropriate step length $\tau \in (0,1]$;
    end
  end
  if $(\sigma \le \varepsilon)$ then $finished\_scaling := true$;
end
```

We set the step length τ as $\tau = (3t\sigma')/(32Mc^2)$ here.

3.1 Analysis of algorithm $\mathcal{A}$

In this subsection we first show the correctness of algorithm $\mathcal{A}$ by proving that if the algorithm stops, the delivered pair x and p is the solution to $(P_{\varepsilon,c})$ and $(D_{\varepsilon,c})$. Afterwards, we will prove that the algorithm stops in each scaling phase after a finite number of iterations. From now on for convenience we denote $\theta = \theta(x)$, $\theta' = \theta(x')$, $f = f(x)$, $f' = f(x')$ and $\hat{f} = f(\hat{x})$ in this section. First we have the following bound on the initial solution x_0 similar to that in [14].

Lemma 4 *If x_0 is the solution of $ABS(e/M, 1/20, c)$, then $\lambda(x_0) \le (21/20)Mc\lambda^*$.*

Before proving the correctness of the algorithm $\mathcal{A}$, the following technical lemma is needed to show that even though there is no guarantee that the sequence of values $\lambda(x)$ computed by algorithm $\mathcal{A}$ is decreasing, $\lambda(x)$ can increases only slightly in each coordination step:

Lemma 5 *For any two consecutive iterates x and $x' \in B$ within a scaling phase of algorithm $\mathcal{A}$, $\lambda(x') \le \theta(x) \le \lambda(x)/(1-\sigma'/10)$.*

Now we show that algorithm $\mathcal{A}$ is correct.

Theorem 6 *If algorithm $\mathcal{A}$ stops, then for any $\varepsilon \in (0, 1/2]$ the pair x and p delivered solves $(P_{\varepsilon,c})$ and $(D_{\varepsilon,c})$ with an approximate block solver $ABS(p, \varepsilon_1/10, c)$, where $\varepsilon_1 = 8\varepsilon/(7-\varepsilon)$.*

The remaining task is to prove that the algorithm $\mathcal{A}$ will halt in finite number of iterations. In order to do so, in the next lemma we show that the reduced potential function ϕ_t decreases boundedly by a constant factor (depending on parameters t, τ and σ) in each coordination step. This helps us to prove an upper bound on the number of iterations.

LEMMA 7 *For any two consecutive iterates x, $x' \in B$ within a scaling phase of algorithm $\mathcal{A}$, $\phi_t(x') \leq \phi_t(x) - t\tau\sigma'/4$.*

THEOREM 8 *For a given accuracy tolerance $\varepsilon \in (0, 1/2]$, the number of coordination steps of algorithm $\mathcal{A}$ is bounded by $N_\varepsilon = O(Mc^2(\ln M + \varepsilon^{-4} \ln c + \varepsilon^{-3} \ln \varepsilon^{-1}))$.*

Similar to the special case of small c discussed in [14], from the Theorem 6 we immediately have the following result:

COROLLARY 9 *If $c \leq 1 + \varepsilon/8$, algorithm $\mathcal{A}$ generates a pair x and p as the solution to (P_ε) and (D_ε) with a weak block solver $ABS(p, \varepsilon_1/10, c)$ within $O(M(\ln M + \varepsilon^{-3} \ln \varepsilon^{-1}))$ iterations.*

Remark: It is worth noting that to compute price vector (4), equation (3) should be solved to obtain $\theta(x)$, while it only can be solved approximately by numerical methods. The way to avoid the influence of numerical error is discussed in [8, 14, 18]. And the numerical overhead in each coordination step can be bounded by $O(M \ln\ln(M\varepsilon^{-1}))$ with the Newton's method.

3.2 Better running time

In [14] it is mentioned that both $(P_{\varepsilon,c})$ and $(D_{\varepsilon,c})$ can be solved in $O(Mc^2(\ln M + \varepsilon^{-3} \ln c + \varepsilon^{-3} \ln \varepsilon^{-1}))$ iterations with an $ABS(p, \varepsilon/10, c)$. However, by Theorem 6 we are only able to show that the bound is $O(Mc^2(\ln M + \varepsilon^{-4} \ln c + \varepsilon^{-3} \ln \varepsilon^{-1}))$ for algorithm $\mathcal{A}$ to solve $(P_{\varepsilon,c})$ and $(D_{\varepsilon,c})$ with $ABS(p, \varepsilon_1/10, c)$. In this way we are going to develop another algorithm $\mathcal{A}'$ to obtain a better running time with a new block solver $ABS(p, \varepsilon_2/10, t)$, where $\varepsilon_2 = 8\varepsilon/((7 + 8r) - (1 - 8r)\varepsilon) \geq 8\varepsilon/(7 + 8r)$ for a constant $r \in (0, 1/8)$. Notice here $\varepsilon_2 \in (\varepsilon, \varepsilon_1)$.

The stopping rule of $\mathcal{A}'$ is as follows:

$$(1 + (1/4 - 2r)\sigma')p^T f(x) - p^T f(\hat{x}) \leq \sigma'\theta(x)/2, \tag{6}$$

where $\sigma' = 8\sigma/((7 + 8r) - (1 - 8r)\sigma)$. And other parts of algorithm $\mathcal{A}'$ are same as $\mathcal{A}$. Lemma 4 and 5 are still valid in this case. Then the following theorem holds:

THEOREM 10 *If algorithm $\mathcal{A}'$ stops, then for any $\varepsilon \in (0, 1/2]$ the pair x and p delivered solves $(P_{\varepsilon,c})$ and $(D_{\varepsilon,c})$ with an approximate block solver $ABS(p, \varepsilon_2/10, c)$.*

As for the running time, we have the following lemma for the bound on increase of reduced potential function similar to Lemma 7:

LEMMA 11 *For any two consecutive iterates x, $x' \in B$ within a scaling phase of algorithm $\mathcal{A}'$, $\phi_t(x') \leq \phi_t(x) - r\tau\sigma'$.*

In this way we can follow the proof of Theorem 8 to obtain the number of iterations of algorithm $\mathcal{A}'$. Since here r is a constant in $(0, 1/8)$, we have the following bound:

THEOREM 12 *For a given accuracy tolerance $\varepsilon \in (0, 1/2]$, the number of coordination steps of algorithm $\mathcal{A}'$ is bounded by $N_\varepsilon = O(Mc^2(\ln M + \varepsilon^{-3}\ln c + \varepsilon^{-2}\ln \varepsilon^{-1}))$.*

This bound is exactly the same as mentioned in [14]. But here we still get some improvement of the approximation ratio though it is not as good as algorithm $\mathcal{A}$.

4. Fast approximation algorithm $\mathcal{F}$ for $(P_{\varepsilon,c})$

In this section, based on the algorithm in [14], we will propose a fast approximation algorithm $\mathcal{F}$ only for $(P_{\varepsilon,c})$ with $ABS(p, \varepsilon_3/6, c)$, where $\varepsilon_3 = (43 - \sqrt{1849 - 1176\varepsilon})/12 \geq 49\varepsilon/43$.

The algorithm works similarly to $\mathcal{A}$. The scaling phase strategy is employed, and in each scaling phase a relative error tolerance σ_s is set. We have two stopping rules here and the iterative procedure in one scaling phase stops if any one of them is fulfilled. Then the error tolerance σ_s is halved and the new scaling phase starts in the same way as in algorithm $\mathcal{A}$, until the error tolerance $\sigma_s \leq \varepsilon$. The solution x_s delivered in the last scaling phase solves $(P_{\varepsilon,c})$ (see also Subsection 4.1).

We also estimate the duality gap to construct the stop rule. For our first stopping rule a parameter ν is defined as follows (same as [14, 18]):

$$\nu = \nu(x, \hat{x}) = \frac{p^T f(x) - p^T f(\hat{x})}{p^T f(x) + p^T f(\hat{x})}. \tag{7}$$

If $\nu = O(\varepsilon)$, then the duality gap is small. However, in the case that ν is large and close to 1, the gap may be extremely large [14]. To obtain a better bound on the number of iterations, we define another parameter to connect the function value with the solution of previous scaling phase.

Let σ_s be the relative error tolerance of the s-th scaling phase. Then similar to [14], the parameter w_s is defined as follows:

$$w_s = \begin{cases} \frac{1+\sigma_1}{(1+\sigma_0/6)M}, & \text{for the first scaling phase;} \\ \frac{1+\sigma_s}{1+2\sigma_s}, & \text{otherwise.} \end{cases} \tag{8}$$

Let x_s be the solution of s-th scaling phase. Then the two stopping rules used in the s-th scaling phase are:

$$\begin{array}{ll} \textit{Rule } 1: & \nu \le \sigma_s'^2/36; \\ \textit{Rule } 2: & \lambda(x) \le w_s\, \lambda(x_{s-1}), \end{array} \tag{9}$$

where $\sigma' = k_\varepsilon \sigma$ and the parameter $k_\varepsilon = (55 + \sqrt{1849 - 1176\varepsilon})/98 < 1$. The stopping rules here are similar to those in [14]. But the latter are only for the case of an $ABS(p, \varepsilon/6, c)$.

We set $t = \sigma_s'/6$ for the error tolerance in the block solver $ABS(p, t, c)$ in algorithm $\mathcal{F}$. We use the solution of the block solver $ABS(e/M, 1/6, c)$ as initial solution x_0, where the price vector e/M is still the vector of all $1/M$'s and the initial error tolerance $\sigma_0 = 1$.

```
Algorithm $\mathcal{F}(f, B, \varepsilon, c)$:
initialize: $s := 0$, $\sigma_1 = \sigma_0 := 1$, $\sigma_1' = \sigma_0' := k_\varepsilon \sigma_0$, $t := \sigma_0'/6$, $p := e/M$;
$x_0 := ABS(p, t, c)$ and $finished\_scaling := false$;
while $not(finished\_scaling)$ do {scaling phase}
  $s := s + 1$, $x := x_{s-1}$ and $finished\_coordination := false$;
  compute $w_s$ from (8);
  while $not(finished\_coordination)$ do {coordination step}
    compute $\theta(x)$ from (3) and $p = p(x) \in P$ from (4);
    $\hat{x} := ABS(p, t, c)$;
    compute $\nu = \nu(x, \hat{x})$ from (7);
    if (Stopping Rule 1 or 2) then
      $x_s := x$ and $finished\_coordination := true$;
    else
      $x := (1 - \tau)x + \tau\hat{x}$ for an appropriate step length $\tau \in (0, 1]$;
    end
  end
  $\sigma_{s+1} := \sigma_s/2$, $\sigma_{s+1}' := k_\varepsilon \sigma_{s+1}$ and $t := \sigma_{s+1}'/6$;
  if $(\sigma_{s+1} \le \varepsilon/2)$ then $finished\_scaling := true$;
end
```

The step length is set as $\tau = t\theta(x)\nu/(2M(p(x)^T f(x) + p(x)^T f(\hat{x})))$, similar to [14, 18].

4.1 Analysis of the algorithm $\mathcal{F}$

We are going to analyze the algorithm $\mathcal{F}$ is this section. We will show the correctness, i.e., to prove that the solution x_s of the last scaling phase is a solution to $(P_{\varepsilon,c})$. Then we will prove that the bound on the number of iterations such that the algorithm stops is polynomial only in M and ε. From now on we denote $\theta = \theta(x)$, $\theta' = \theta(x')$, $f = f(x)$, $f' = f(x')$ and $\hat{f} = f(\hat{x})$. First we can obtain the following bound on the function value of the initial solution x_0, similar to that in [14, 18].

Lemma 13 *If x_0 is the solution of $ABS(e/M, t, c)$ with $t = 1/6$, then $\lambda(x_0) \leq (7/6)cM\lambda^*$.*

We can prove the following theorem by showing that at the end of the s-th scaling phase the solution satisfies $\lambda(x) \leq c(1+\sigma_s)\lambda^*$:

Theorem 14 *If algorithm $\mathcal{F}$ stops, then for any $\varepsilon \in (0,1]$ the computed solution $x \in B$ fulfils $(P_{\varepsilon,c})$ with $ABS(p, \varepsilon_3/6, c)$, where $\varepsilon_3 = (43 - \sqrt{1849 - 1176\varepsilon})/12$.*

Then we are to find the bound on the number of iterations of the algorithm $\mathcal{F}$. In the next lemma we show that the decrease of the reduced potential function ϕ_t in each iteration is lower-bounded by a parameter depending only on t, ν and M, similar to Lemma 7. This helps us to prove an upper bound on the number of iterations.

Lemma 15 *For any two consecutive iterates x, $x' \in B$ within a scaling phase of algorithm $\mathcal{F}$, $\phi_t(x') \leq \phi_t(x) - t\nu^2/(4M)$.*

From the above bound we are able to obtain the bound on the number of iterations of algorithm $\mathcal{F}$.

Theorem 16 *For a given relative accuracy tolerance $\varepsilon \in (0,1]$, algorithm $\mathcal{F}$ delivers a solution x satisfying $\lambda(x) \leq c(1+\varepsilon)\lambda^*$ with a weak block solver $ABS(p, \varepsilon_3/6, c)$ in $N = O(M(\ln M + \varepsilon^{-4}\ln\varepsilon^{-1}))$ coordination steps.*

Remark: The running time here is worse than that in [14]. However, a block solver $ABS(p, \varepsilon/6, c)$ is required in [14] while here we only need a $ABS(p, \varepsilon_3/6, c)$. In addition, different from Theorem 8, we have got the first algorithm with the iteration complexity independent of c in the case of weaker block solvers.

Similar to the special case of small c ($\ln c = O(\varepsilon)$) discussed in [14], we here can also design a faster algorithm $\bar{\mathcal{F}}$ with only the first stopping rule. It can be proved that $\bar{\mathcal{F}}$ can solve both primal and dual problems with a better bound on the number of iterations. Therefore we have the following result:

COROLLARY 17 *If $c \leq 1/k_\varepsilon$, the algorithm $\bar{\mathcal{F}}$ can generate a pair x and p solving both (P_ε) and (D_ε) with only the weak approximate block solver $ABS(p, \varepsilon_3/6, c)$ within $O(M(\ln M + \varepsilon^{-4}))$ iterations.*

4.2 Better running time

The number of iterations of the algorithm for primal problem in [14] is bounded by $O(M(\ln M + \varepsilon^{-2} \ln \varepsilon))$, which is better than the bound in Theorem 16. In addition, in Subsection 3.2 it has been showed that a better bound on running time can be achieved with a different weak block solver. Here we also get such an algorithm with this technique.

Similar to the cases in Subsection 3.2, we can develop an algorithm $\mathcal{F}'$ by slight modification of the stopping rules. Suppose $r \in (0,1)$ is a constant. And here a function $h(r)$ is defined as:

$$h(r) = \begin{cases} \frac{2r(1-r)}{3(1+r)^2}, & \text{if } r \geq \frac{3}{4}; \\ \frac{6(1-r)}{7(6-r)}, & \text{otherwise.} \end{cases} \tag{10}$$

And $\varepsilon_4 = ((1-h(r)) - \sqrt{(1-h(r))^2 - 4h(r)\varepsilon})/(2h(r)) > \varepsilon/(1-h(r))$, and $k_{r,\varepsilon} = 1 - h(r)\varepsilon_4$. Define $\sigma'_s = k_{r,\varepsilon}\sigma_s$. Then the stopping rules of $\mathcal{F}'$ are as follows:

$$\begin{array}{ll} \textit{Rule } 1: & \nu \leq r\sigma'_s/6; \\ \textit{Rule } 2: & \lambda(x) \leq w_s \, \lambda(x_{s-1}), \end{array} \tag{11}$$

Lemma 13 is still valid for algorithm $\mathcal{F}'$. Similar to that for $\mathcal{F}$, we have the following theorem:

THEOREM 18 *If algorithm $\mathcal{F}'$ stops, then for any $\varepsilon \in (0,1]$ the solution x delivered satisfies $(P_{\varepsilon,c})$ with $ABS(p, \varepsilon_4/6, c)$.*

As for the running time, we have also the same bound on increase of reduced potential function for $\mathcal{F}'$ as in Lemma 15. To find the bound on number of iterations of algorithm $\mathcal{F}'$, we can just apply the similar argument to the proof of Theorem 16. Since here r is a constant in $(0,1)$, we have the following theorem:

THEOREM 19 *For a given relative accuracy $\varepsilon \in (0,1]$, the number of coordination steps of algorithm $\mathcal{F}'$ is bounded by $N = O(M(\ln M + \varepsilon^{-2} \ln \varepsilon^{-1}))$.*

This bound exactly matches the bound in [14]. But here we just need a weaker block solver.

Remark: We find that if we design the first stopping rule as $\nu \leq v$ for any $v < t$, we can always have a $\varepsilon' > \varepsilon$ for $ABS(p, \varepsilon'/6, c)$ called in algorithm. A reasonable choice, $v = t^q$ for large q, can generate a large ε'. Unfortunately this kind of improvement is very limited and the running time increases considerable for the bound on the number of iterations is $O(M(\ln M + \varepsilon^{-2q} \ln \varepsilon^{-1}))$.

5. Conclusion and open problem

In this paper we have presented the first $c(1+\varepsilon)$-approximation algorithms for the general packing problem (or with its dual problem), with only weaker block solvers. The number of iterations is bounded by polynomials in M, ε and c or even only in M and ε. We also reduced the bounds to the same as in [14].

An interesting problem is whether one can find $c'(1+\varepsilon)$-approximation algorithms for general packing problem with only an approximate block solver $ABS(p, O(\varepsilon), c)$, where $c' < c$. By the gap between $p^T f(x)$ and $p^T f(\hat{x})$ we conjecture that it is possible and the lower bound on c' is $(c+1)/2$.

References

[1] A. Baltz and A. Srivastav, Fast Approximation of Minimum Multicast Congestion - Implementation versus Theory, *Proceedings of 5th Conference on Algorithms and Complexity*, CIAC 2003.

[2] M. Charikar, C. Chekuri, A. Goel, S. Guha and S. Plotkin, Approximating a finite metric by a small number of tree metrics, *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, FOCS 1998, 379-388.

[3] G. Even, J. S. Naor, S. Rao and B. Schieber, Fast approximate graph partitioning algorithms, *SIAM. Journal on Computing*, 6 (1999), 2187-2214.

[4] N. Garg and J. Könemann, Fast and simpler algorithms for multicommodity flow and other fractional packing problems, *Proceedings of the 39th IEEE Annual Symposium on Foundations of Computer Science*, FOCS 1998, 300-309.

[5] M. D. Grigoriadis and L. G. Khachiyan, Fast approximation schemes for convex programs with many blocks and coupling constraints, *SIAM Journal on Optimization*, 4 (1994), 86-107.

[6] M. D. Grigoriadis and L. G. Khachiyan, Coordination complexity of parallel price-directive decomposition, *Mathematics of Operations Research*, 2 (1996), 321-340.

[7] M. D. Grigoriadis and L. G. Khachiyan, Approximate minimum-cost multicommodity flows in $O(\varepsilon^{-2}knm)$ time, *Mathematical Programming*, 75 (1996), 477-482.

[8] M. D. Grigoriadis, L. G. Khachiyan, L. Porkolab and J.Villavicencio, Approximate max-min resource sharing for structured concave optimization, *SIAM Journal on Optimization*, 11 (2001), 1081-1091.

[9] K. Jansen, Approximation algorithms for fractional covering and packing problems, and applications, Manuscript, (2001).

[10] K. Jansen, Approximation algorithms for the general max-min resource sharing problem: faster and simpler, *Proceedings of the 9th Scandinavian Workshop on Algorithm Theory*, SWAT 2004, LNCS.

[11] K. Jansen, Approximation algorithms for the mixed fractional packing and covering problem, *these proceedings*.

[12] K. Jansen and L. Porkolab, On preemptive resource constrained scheduling: polynomial-time approximation schemes, *Proceedings of the 9th Conference on Integer Programming and Combinatorial Optimization*, IPCO 2002, LNCS 2337 329-349.

[13] K. Jansen and R. Solis-Oba, An asymptotic fully polynomial time approximation scheme for bin covering, *Proceedings of 13th International Symposium on Algorithms and Computation*, ISAAC 2002.

[14] K. Jansen and H. Zhang, Approximation algorithms for general packing problems with modified logarithmic potential function, *Proceedings of 2nd IFIP International Conference on Theoretical Computer Science*, TCS 2002.

[15] K. Jansen and H. Zhang, An approximation algorithm for the multicast congestion problem via minimum Steiner trees, *Proceedings of 3rd International Workshop on Approximation and Randomized Algorithms in Communication Networks*, ARANCE 2002.

[16] C. Kenyon and E. Rémila, Approximate strip packing, *Proceedings of 37th Annual Symposium on Foundations of Computer Science*, FOCS 1996, 31-36.

[17] S. A. Plotkin, D. B. Shmoys and E. Tardos, Fast Approximation algorithms for fractional packing and covering problems, *Mathematics of Operations Research*, 2 (1995), 257-301.

[18] J. Villavicencio and M. D. Grigoriadis, Approximate Lagrangian decomposition with a modified Karmarkar logarithmic potential, *Network Optimization, P. Pardalos, D. W. Hearn and W. W. Hager, Eds, Lecture Notes in Economics and Mathematical Systems 450, Springer-Verlag, Berlin*, (1997), 471-485.

[19] D. Ye and H. Zhang The Range Assignment Problem in Static Ad-Hoc Networks on Metric Spaces, *Proceedings of the 11th Colloquium on Structural Information and Communication Complexity* Sirocco 2004, LNCS.

[20] N. E. Young, Randomized rounding without solving the linear program, *Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms*, SODA 1995, 170–178.

[21] N. E. Young, Sequential and parallel algorithms for mixed packing and covering, *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, FOCS 2001, 538-546.

[22] H. Zhang, Packing: Scheduling, Embedding and Approximating Metrics, *Proceedings of the 2004 International Conference on Computational Science and its Applications* ICCSA 2004, LNCS 3045.