An Approximation Algorithm for Hypergraph Max k-Cut with Given Sizes of Parts

Alexander A. Ageev[1]⋆ and Maxim I. Sviridenko[2]⋆⋆

[1] Sobolev Institute of Mathematics, pr. Koptyuga 4, 630090, Novosibirsk, Russia
ageev@math.nsc.ru
[2] BRICS, University of Aarhus, Aarhus, Denmark
sviri@brics.dk

Abstract. An instance of Hypergraph Max k-Cut with given sizes of parts (or Hyp Max k-Cut with gsp) consists of a hypergraph $H = (V, E)$, nonnegative weights w_S defined on its edges $S \in E$, and k positive integers $p_1, \dots, p_k$ such that $\sum_{i=1}^{k} p_i = |V|$. It is required to partition the vertex set V into k parts $X_1, \dots, X_k$, with each part X_i having size p_i, so as to maximize the total weight of edges not lying entirely in any part of the partition. The version of the problem in which $|X_i|$ may be arbitrary is known to be approximable within a factor of 0.72 of the optimum (Andersson and Engebretsen, 1998). The authors (1999) designed an 0.5-approximation for the special case when the hypergraph is a graph. The main result of this paper is that Hyp Max k-Cut with gsp can be approximated within a factor of $\min\{\lambda_{|S|} : S \in E\}$ of the optimum, where $\lambda_r = 1 - (1 - 1/r)^r - (1/r)^r$.

1 Introduction

The last decade has been marked by striking breakthroughs in designing approximation algorithms with provable performance guarantees. Most of them to all appearances are due to using novel methods of rounding polynomially solvable fractional relaxations. Applicability of a rounding method is highly dependent on the type of constraints in the relaxation. In [1] the authors presented a new rounding technique (the pipage rounding) especially oriented to tackle some NP-hard problems, which can be formulated as integer programs with assignment-type constraints. The paper [1] contains four approximation results demonstrating the efficiency of this technique. One of the problems treated in [1] is Max k-Cut with given sizes of parts (or Max k-Cut with gsp). An instance of this problem consists of an undirected graph $G = (V, E)$, nonnegative edge weights $w_e, e \in E$, and k positive integers $p_1, p_2, \dots, p_k$ such that $\sum_{i=1}^{k} p_i = |V|$. It is required to find a partition of V into k parts $V_1, V_2, \dots, V_k$ with each part V_i having size p_i, so as to maximize the total weight of edges whose ends lie in different parts of the partition. The paper [1] gives an 0.5-approximation for

⋆ Supported in part by the Russian Foundation for Basic Research, grant 99-01-00601.
⋆⋆ Supported in part by the Russian Foundation for Basic Research, grant 99-01-00510.

M. Paterson (Ed.): ESA 2000, LNCS 1879, pp. 32–41, 2000.

this problem. Very recently, Feige and Langberg [7] by a combination of the method of [1] with the semidefinite programming technique designed an $0.5+\varepsilon$-approximation for MAX 2-CUT WITH GSP, where ε is some unspecified small positive number.

The MAX CUT and MAX k-CUT problems are classical in combinatorial optimization and have been extensively studied in the absence of any restrictions on the sizes of parts. The best known approximation algorithm for MAX CUT is due to Goemans and Williamson [9] and has a performance guarantee of 0.878. Frieze and Jerrum [8] extended the technique of Goemans and Williamson to MAX k-CUT and designed a $(1-1/k+2\ln k/k^2)$-approximation algorithm. On the other hand, as it was shown by Kann et al. [10], no approximation algorithm for MAX k-CUT can have a performance guarantee better than $1-1/34k$, unless P=NP.

Approximation results for some special cases of MAX k-CUT WITH GSP have been also established. In particular, Frieze and Jerrum [8] present an 0.65-approximation algorithm for MAX BISECTION (the special case of MAX k-CUT WITH GSP where $k=2$ and $p_1=p_2=|V|/2$). Very recently, Ye [11] announced an algorithm with a better performance guarantee of 0.699. The best known approximation algorithm for MAX k-SECTION (the case where $p_1=\cdots=p_k=|V|/k$) is due to Andersson [2] and has a performance guarantee of $1-1/k+\Omega(1/k^3)$.

In this paper we consider a hypergraph generalization of MAX k-CUT WITH GSP — Hypergraph Max k-Cut with given sizes of parts or, for short, HYP MAX k-CUT WITH GSP. An instance of HYP MAX k-CUT WITH GSP consists of a hypergraph $H=(V,E)$, nonnegative weights w_S on its edges S, and k positive integers $p_1,\ldots,p_k$ such that $\sum_{i=1}^{k} p_i=|V|$. It is required to partition the vertex set V into k parts $X_1,X_2,\ldots,X_k$, with each part X_i having size p_i, so as to maximize the total weight of the edges of H, not lying entirely in any part of the partition (i.e., to maximize the total weight of $S\in E$ satisfying $S\not\subseteq X_i$ for all i).

Several closely related versions of HYP MAX k-CUT WITH GSP were studied in the literature but few results have been obtained. Andersson and Engebretsen [3] presented an 0.72-approximation algorithm for the ordinary HYP MAX CUT problem (i.e., for the version without any restrictions on the sizes of parts). Arora, Karger, and Karpinski [4] designed a PTAS for dense instances of this problem or, more precisely, for the case when the hypergraph H is restricted to have $\Theta(|V|^d)$ edges, under the assumption that $|S|\le d$ for each edge S and some constant d.

In this paper, by applying the pipage rounding method, we prove that HYP MAX k-CUT WITH GSP can be approximated within a factor of $\min\{\lambda_{|S|} : S\in E\}$ of the optimum, where $\lambda_r=1-(1-1/r)^r-(1/r)^r$. By direct calculations it easy to get some specific values of λ_r: $\lambda_2=1/2=0.5$, $\lambda_3=2/3\approx 0.666$, $\lambda_4=87/128\approx 0.679$, $\lambda_5=84/125=0.672$, $\lambda_6\approx 0.665$ and so on. It is clear that λ_r tends to $1-e^{-1}\approx 0.632$ as $r\to\infty$. A bit less trivial fact is that $\lambda_r>1-e^{-1}$ for each $r\ge 3$ (Lemma 2 in this paper). Summing up we arrive at the following conclusion: our algorithm finds a feasible cut of weight within a factor of 0.5 on general hypergraphs, i.e., in the case when each edge of the hypergraph has size at least 2, and within a factor of $1 - e^{-1} \approx 0.632$ in the case when each edge has size at least 3. Note that the first bound coincides with that obtained in [1] for the case of graphs. In this paper we also show that in the case of hypergraphs with each edge of size at least 3 the bound of $1 - e^{-1}$ cannot be improved, unless P=NP.

2 The Pipage Rounding: A General Scheme

We begin with a description of the pipage rounding [1] for the case of a slightly more general constraints.

Assume that a problem P can be reformulated as the following nonlinear binary program:

$$\max \quad F(x_{11}, \dots, x_{nk}) \tag{1}$$

$$\text{s. t.} \quad \sum_{i=1}^{n} x_{it} = p_t, \quad t = 1, \dots, k, \tag{2}$$

$$\sum_{t=1}^{k} x_{it} = 1, \quad i = 1, \dots, n, \tag{3}$$

$$x_{it} \in \{0, 1\}, \quad t = 1, \dots, k, \quad i = 1, \dots, n, \tag{4}$$

where $p_1, p_2, \dots, p_k$ are positive integers such that $\sum_t p_t = n$, $F(x)$ is a function defined on the rational points $x = (x_{it})$ of the $n \times k$-dimensional cube $[0,1]^{n \times k}$ and computable in polynomial time. Assume further that one can associate with $F(x)$ another function $L(x)$ that is defined and polynomially computable on the same set, coincides with $F(x)$ on binary x satisfying (2)–(3), and such that the program

$$\max \quad L(x) \tag{5}$$

$$\text{s. t.} \quad \sum_{i=1}^{n} x_{it} = p_t, \quad t = 1, \dots, k, \tag{6}$$

$$\sum_{t=1}^{k} x_{it} = 1, \quad i = 1, \dots, n, \tag{7}$$

$$0 \le x_{it} \le 1, \quad t = 1, \dots, k, \quad i = 1, \dots, n \tag{8}$$

(henceforth called the *nice relaxation*) is polynomially solvable. Assume next that the following two main conditions hold. The first—*F/L lower bound condition*—states: there exists $C > 0$ such that $F(x)/L(x) \ge C$ for each $x \in [0,1]^{n \times k}$. To formulate the second—*ε-convexity condition*—we need a description of the so-called pipage step.

Let x be a feasible solution to (5)–(8). Define the bipartite graph H with the bipartition $(\{1, \dots, n\}, \{1, \dots, k\})$ so that $jt \in E(H)$ if and only if x_{jt} is non-integral. Note that (6) and (7) imply that each vertex of H is either isolated or has degree at least 2. Assume that x has fractional components. Since H is bipartite it follows that H has a cycle D of even length. Let M_1 and M_2 be the matchings of H whose union is the cycle D. Define a new solution $x(\varepsilon)$ by the following rule: if jt is not an edge of D, then $x_{jt}(\varepsilon)$ coincides with x_{jt}, otherwise, $x_{jt}(\varepsilon) = x_{jt} + \varepsilon$ if $jt \in M_1$, and $x_{jt}(\varepsilon) = x_{jt} - \varepsilon$ if $jt \in M_2$.

By definition $x(\varepsilon)$ is a feasible solution to the linear relaxation of (5)–(8) for all $\varepsilon \in [-\varepsilon_1, \varepsilon_2]$ where

$$\varepsilon_1 = \min\{\min_{jt\in M_1} x_{jt}, \min_{jt\in M_2}(1 - x_{jt})\}$$

and

$$\varepsilon_2 = \min\{\min_{jt\in M_1}(1 - x_{jt}), \min_{jt\in M_2} x_{jt}\}.$$

The *ε-convexity condition* states that for every feasible x and every cycle D in the graph H, $\varphi(\varepsilon) = F(x(\varepsilon))$ is a convex function on the above interval.

Under the above assumptions we claim that there exists a polynomial-time C-approximation algorithm for solving P. Indeed, since the function $\varphi(\varepsilon) = F(x(\varepsilon))$ is convex,

$$F(x(\varepsilon^*)) \geq F(x) \geq CL(x)$$

for some $\varepsilon^* \in \{-\varepsilon_1, \varepsilon_2\}$. The new solution $x(\varepsilon^*)$, being feasible for (5)–(8), has a smaller number of fractional components. Set $x' = x(\varepsilon^*)$ and, if x' has fractional components, apply to x' the above described pipage step and so on. Ultimately, after at most nk steps, we arrive at a solution $\tilde{x}$ which is feasible for (1)–(4) and satisfies

$$F(\tilde{x}) \geq CL(x) \geq CF^*$$

where F^* is an optimal value of (1)–(4) (and of the original problem P). The rounding procedure described (and henceforth called the *pipage rounding*) can be clearly implemented in polynomial time.

Thus we obtain a C-approximation algorithm for P. It consists of two phases: the first phase is to find a feasible (fractional) solution to (5)–(8), and the second is to round off this solution by using the pipage rounding.

3 The Pipage Rounding: An Application to the Problem

It is easy to see that an instance of Hyp Max k-Cut with gsp can be equivalently formulated as the following (nonlinear) integer program:

$$\max \quad F(x) = \sum_{S\in E} w_S(1 - \sum_{t=1}^{k}\prod_{i\in S} x_{it}) \tag{9}$$

$$\text{s. t.} \quad \sum_{t=1}^{k} x_{it} = 1 \text{ for all } i, \tag{10}$$

$$\sum_{i=1}^{n} x_{it} = p_t \ \text{ for all } t, \tag{11}$$

$$x_{it} \in \{0,1\} \text{ for all } i \text{ and } t. \tag{12}$$

The equivalence is shown by the one-to-one correspondence between optimal solutions to the above program and optimal k-cuts $\{X_1, \dots, X_k\}$ of instance of Hyp Max k-Cut with gsp defined by the relation "$x_{it} = 1$ if and only if $i \in X_t$".

As a nice relaxation we consider the following linear program:

$$\max \quad \sum_{S \in E} w_S z_S \tag{13}$$

$$\text{s. t.} \quad z_S \le |S| - \sum_{i \in S} x_{it} \quad \text{ for all } S \in E, \tag{14}$$

$$\sum_{t=1}^{k} x_{it} = 1 \quad \text{ for all } i, \tag{15}$$

$$\sum_{i=1}^{n} x_{it} = p_t \quad \text{ for all } t, \tag{16}$$

$$0 \le x_{it} \le 1 \quad \text{ for all } i \text{ and } t, \tag{17}$$

$$0 \le z_S \le 1 \quad \text{ for all } S \in E. \tag{18}$$

It is easy to see that, given a feasible matrix x, the optimal values of z_S in the above program can be uniquely determined by simple formulas. Using this observation we can exclude the variables z_S and rewrite (13)–(18) in the following equivalent way:

$$\max \quad L(x) = \sum_{S \in E} w_S \min\{1, \min_t(|S| - \sum_{i \in S} x_{it})\} \tag{19}$$

subject to (15)–(17). Note that $F(x) = L(x)$ for each x satisfying (10)–(12).

We claim that, for every feasible x and every cycle D in the graph H (for definitions, see Section 2), the function $\varphi(\varepsilon) = F(x(\varepsilon))$ is a quadratic polynomial with a nonnegative leading coefficient. Indeed, observe that each product $\prod_{i \in S} x_{it}(\varepsilon)$ contains at most two modified variables. Assume that a product $\prod_{i \in S} x_{it}(\varepsilon)$ contains exactly two such variables $x_{i_1 t}(\varepsilon)$ and $x_{i_2 t}(\varepsilon)$. Then they can have only one of the following forms: either $x_{i_1 t} + \varepsilon$ and $x_{i_2 t} - \varepsilon$ or $x_{i_1 t} - \varepsilon$ and $x_{i_2 t} + \varepsilon$, respectively. In either case ε^2 has a nonnegative coefficient in the term corresponding to the product. This proves that the ε-convexity condition does hold.

For any $r \ge 1$, set $\lambda_r = 1 - (1 - 1/r)^r - (1/r)^r$.

Lemma 1. *Let $x = (x_{it})$ be a feasible solution to (19),(15)–(17) and $S \in E$. Then*

$$(1 - \sum_{t=1}^{k} \prod_{i \in S} x_{it}) \ge \lambda_{|S|} \min\{1, \min_t(|S| - \sum_{i \in S} x_{it})\}.$$

Proof. Let $z_S = \min\{1, \min_t(|S| - \sum_{i\in S} x_{it})\}$. Define q_S and t' by the equalities

$$q_S = \max_t \sum_{i\in S} x_{it} = \sum_{i\in S} x_{it'}.$$

Note that

$$z_S = \min\{1, |S| - q_S\}. \tag{20}$$

Using the arithmetic-geometric mean inequality and the fact that

$$\sum_{t=1}^{k}\sum_{i\in S} x_{it} = |S|$$

we obtain that

$$\begin{aligned}
1 - \sum_{t=1}^{k}\prod_{i\in S} x_{it} &= 1 - \prod_{i\in S} x_{it'} - \sum_{t\neq t'}\prod_{i\in S} x_{it} \\
&\geq 1 - \left(\frac{\sum_{i\in S} x_{it'}}{|S|}\right)^{|S|} - \sum_{t\neq t'}\left(\frac{\sum_{i\in S} x_{it}}{|S|}\right)^{|S|} \\
&\geq 1 - \left(\frac{q_S}{|S|}\right)^{|S|} - \left(\frac{\sum_{t\neq t'}\sum_{i\in S} x_{it}}{|S|}\right)^{|S|} \\
&= 1 - \left(\frac{q_S}{|S|}\right)^{|S|} - \left(\frac{|S| - \sum_{i\in S} x_{it'}}{|S|}\right)^{|S|} \\
&= 1 - \left(\frac{q_S}{|S|}\right)^{|S|} - \left(1 - \frac{q_S}{|S|}\right)^{|S|}.
\end{aligned} \tag{21}$$

Let $\psi(y) = 1 - \left(1 - \frac{y}{|S|}\right)^{|S|} - \left(\frac{y}{|S|}\right)^{|S|}$.

Case 1. $|S| - 1 \leq q_S \leq |S|$. Then by (20), $z_S = |S| - q_S$, and hence by (21),

$$1 - \sum_{t=1}^{k}\prod_{i\in S} x_{it} \geq 1 - \left(1 - \frac{z_S}{|S|}\right)^{|S|} - \left(\frac{z_S}{|S|}\right)^{|S|} = \psi(z_S).$$

Since the function ψ is concave and $\psi(0) = 0$, $\psi(1) = \lambda_{|S|}$, it follows that

$$1 - \sum_{t=1}^{k}\prod_{i\in S} x_{it} \geq \lambda_{|S|} z_S.$$

Case 2. $1 \leq q_S \leq |S| - 1$. Here $z_S = 1$. Since $\psi(y)$ is concave and $\psi(1) = \psi(|S| - 1) = \lambda_{|S|}$,

$$1 - \sum_{t=1}^{k}\prod_{i\in S} x_{it} \geq \lambda_{|S|}.$$

Case 3. $0 \le q_S \le 1$. Again, $z_S = 1$. For every t, set $\mu_t = \sum_{i\in S} x_{it}$. Note that, by the assumption of the case,

$$0 \le \mu_t \le 1, \tag{22}$$

and, moreover,

$$\sum_{t=1}^{k} \mu_t = |S|. \tag{23}$$

By the arithmetic-geometric mean inequality it follows that

$$\begin{aligned}
\sum_{t=1}^{k}\prod_{i\in S} x_{it} &\le \sum_{t=1}^{k}\Big(\frac{\mu_t}{|S|}\Big)^{|S|} \\
\text{(by (22))}\quad &\le |S|^{-|S|}\sum_{t=1}^{k}\mu_t \\
\text{(by (23))}\quad &= |S|^{-|S|}|S|.
\end{aligned}$$

Consequently,

$$\begin{aligned}
1-\sum_{t=1}^{k}\prod_{i\in S} x_{it} &\ge 1-|S|\Big(\frac{1}{|S|}\Big)^{|S|} \\
&= 1-\Big(\frac{1}{|S|}\Big)^{|S|} - (|S|-1)\Big(\frac{1}{|S|}\Big)^{|S|} \\
&\ge 1-\Big(\frac{1}{|S|}\Big)^{|S|} - (|S|-1)^{|S|}\Big(\frac{1}{|S|}\Big)^{|S|} \\
&= \lambda_{|S|}.
\end{aligned}$$

□

Corollary 1. *Let $x = (x_{it})$ be a feasible solution to (19),(15)–(17). Then*

$$F(x) \ge (\min_{S\in E}\lambda_{|S|})L(x).$$

□

The corollary states that the F/L lower bound condition holds with

$$C = \min_{S\in E}\lambda_{|S|}.$$

Hence the pipage rounding provides an algorithm that finds a feasible k-cut whose weight is within a factor of $\min_{S\in E}\lambda_{|S|}$ of the optimum.

Note that $\lambda_2 = 1/2$. We now establish a lower bound on λ_r for all $r \ge 3$.

Lemma 2. *For any $r \ge 3$,*

$$\lambda_r > 1 - e^{-1}.$$

Proof. We first deduce it from the following stronger inequality:

$$\left(1-\frac{1}{r}\right)^r < e^{-1}\left(1-\frac{1}{2r}\right) \text{ for all } r \geq 1. \tag{24}$$

Indeed, for any $r \geq 3$,

$$\begin{aligned}
\lambda_r &= 1-\frac{1}{r^r}-\left(1-\frac{1}{r}\right)^r \\
&> 1-\frac{1}{r^r}-e^{-1}\left(1-\frac{1}{2r}\right) \\
&= 1-e^{-1}+\frac{1}{r}\left(\frac{e^{-1}}{2}-\frac{1}{r^{r-1}}\right) \\
&> 1-e^{-1}.
\end{aligned}$$

To prove (24), by taking natural logarithm of both sides of (24) rewrite it in the following equivalent form:

$$1+r\ln(1-\frac{1}{r}) < \ln(1-\frac{1}{2r}) \text{ for all } r \geq 1.$$

Using the Taylor series expansion

$$\ln(1-\sigma) = -\sum_{i=1}^{\infty}\frac{\sigma^i}{i}$$

we obtain that for each $r = 1, 2, \ldots,$

$$\begin{aligned}
1+r\ln\left(1-\frac{1}{r}\right) &= 1+r\left(-\frac{1}{r}-\frac{1}{2r^2}-\frac{1}{3r^3}-\ldots\right) \\
&= -\frac{1}{2r}-\frac{1}{3r^2}-\frac{1}{4r^3}\cdots \\
&< -\frac{1}{2r}-\frac{1}{2(2r)^2}-\frac{1}{3(2r)^3}\cdots \\
&= \ln(1-\frac{1}{2r}),
\end{aligned}$$

as required. □

We now show that in the case of r-uniform hypergraphs the integrality gap for the relaxation (13)–(18) can be arbitrarily close to λ_r. It follows that no other rounding of this relaxation can provide an algorithm with a better performance guarantee.

Indeed, consider the following instance: the complete r-uniform hypergraph on $n = rq$ vertices, $k = 2$, $w_S = 1$ for all $S \in E$, $p_1 = q$ and $p_2 = n - q$. It is clear that any feasible cut in this hypergraph has weight

$$C_n^r - C_q^r - C_{n-q}^r.$$

Consider the feasible solution to (15)–(18) in which

$$x_{i1} = 1/r \text{ and } x_{i2} = 1 - 1/r \text{ for each } i.$$

The weight of this solution is equal to C_n^r, since for each edge S we have

$$r - \sum_{i \in S} x_{i1} \geq r - \sum_{i \in S} x_{i2} = 1$$

and therefore $z_S = 1$ for all $S \in E$. Thus the integrality gap for this instance is at most

$$\begin{aligned}
\frac{C_n^r - C_q^r - C_{n-q}^r}{C_n^r} &= 1 - \frac{q!(n-r)!}{(q-r)!n!} - \frac{(n-q)!(n-r)!}{(n-q-r)!n!} \\
&\leq 1 - \frac{q!}{(q-r)!n^r} - \frac{(n-q)!}{(n-q-r)!n^r} \\
&\leq 1 - \frac{(q-r)^r}{n^r} - \frac{(n-q-r)^r}{n^r} \\
&= 1 - \left(\frac{1}{r} - \frac{1}{q}\right)^r - \left(1 - \frac{1}{r} - \frac{1}{q}\right)^r,
\end{aligned}$$

which tends to λ_r as $q \to \infty$.

We conclude the paper with a proof that the performance bound of $1 - e^{-1}$, our algorithm provides on hypergraphs with each edge of size at least 3, cannot be improved, unless $P = NP$.

In the Maximum Coverage problem (Maximum Coverage for short), given a family $\mathcal{F} = \{S_j : j \in J\}$ of subsets of a set $I = \{1, \ldots, n\}$ with associated nonnegative weights w_j and a positive integer p, it is required to find a subset $X \subseteq I$ (called *coverage*) with $|X| = p$ so as to maximize the total weight of the sets in $\mathcal{F}$ having nonempty intersections with X. It is well known that a simple greedy algorithm solves Maximum Coverage approximately within a factor of $1 - e^{-1}$ of the optimum (Cornuejols, Fisher and Nemhauser [5]). Feige [6] proved that no polynomial algorithm can have better performance guarantee, unless P=NP.

Our proof consists in constructing an approximation preserving reduction from Maximum Coverage to Hyp Max k-Cut with gsp. Let a set I, a collection $S_1, \ldots, S_m \subseteq I$, nonnegative weights (w_j), and a positive number p form an instance A of Maximum Coverage. Construct an instance B of Hyp Max k-Cut with gsp as follows: $I' = I \cup \{u_1, \ldots, u_m\}$ (assuming that $I \cap \{u_1, \ldots, u_m\} = \emptyset$), $(S_1' = S_1 \cup \{u_1\}, \ldots, S_m' = S_m \cup \{u_m\})$, the same weights w_j, and $p_1 = p$, $p_2 = |I'| - p$. Let $(X, I' \setminus X)$ be a maximum weight cut in B with the sizes of parts p_1 and p_2. It is clear that its weight is at least the weight of a maximum coverage in A. Thus it remains to transform $(X, I' \setminus X)$ into a coverage of A with the same weight. If $X \subseteq I$, we are done. Assume that X contains u_j for some j. Then successively, for each such j, replace u_j in X by an arbitrary element in S_j that is not a member of X, or if $S_j \subseteq X$, by an arbitrary element of I that is not a member of X. After this transformation and after possibly including a few more elements from I to get exactly p elements, we arrive at a coverage $Y \subseteq I$ in A whose weight is at least the weight of the cut $(X, I' \setminus X)$ in B, as desired.

References

1. A. A. Ageev and M. I. Sviridenko, Approximation algorithms for Maximum Coverage and Max Cut with given sizes of parts, *Lecture Notes in Computer Science (Proceedings of IPCO'99)* **1610** (1999), 17–30. 32, 33, 34
2. G. Andersson, An approximation algorithm for Max p-Section, *Lecture Notes in Computer Science (Proceedings of STACS'99)* **1563** (1999), 237–247. 33
3. G. Andersson and L. Engebretsen, Better approximation algorithms for Set splitting and Not-All-Equal SAT, *Inform. Process. Letters* **65** (1998), 305–311. 33
4. S. Arora, D. Karger, and M. Karpinski, Polynomial Time Approximation Schemes for Dense Instances of NP-Hard Problems, *J. of Comput. and Syst. Sci.* **58** (1999), 193–210. 33
5. G. Cornuejols, M. L. Fisher, and G. L. Nemhauser, Location of bank accounts to optimize float: an analytic study exact and approximate algorithms, *Management Science* **23** (1977), 789–810. 40
6. U. Feige, A threshold of $\ln n$ for approximating set cover, *J. of ACM* **45** (1998), 634–652. 40
7. U. Feige and M. Langberg, Approximation algorithms for maximization problems arising in graph partitioning, manuscript, 1999. 33
8. A. Frieze and M. Jerrum, Improved approximation algorithms for MAX k-CUT and MAX BISECTION, *Algorithmica* **18** (1997), 67–81. 33
9. M. X. Goemans and D. P. Williamson, Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming, *J. of ACM* **42** (1995), 1115–1145. 33
10. V. Kann, S. Khanna, J. Lagergren, and A. Panconesi, On the hardness of approximating Max k-Cut and its dual, *Chic. J. Theor. Comput. Sci.* 1997, Article no. 2 (1997). 33
11. Y. Ye, An 0.699-approximation algorithm for Max-Bisection, manuscript, 1999. 33