BRICS

Basic Research in Computer Science

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

Alexander A. Ageev
Maxim I. Sviridenko

BRICS Report Series **RS-99-49**

ISSN 0909-0878 **December 1999**

See back inner page for a list of recent BRICS Report Series publications.
Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark
Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/99/49/`

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

Alexander A. Ageev
Sobolev Institute
of Mathematics
Novosibirsk, Russia
ageev@math.nsc.ru

Maxim I. Sviridenko
BRICS
University of Aarhus
Aarhus, Denmark
sviri@brics.dk

BRICS*

1 Introduction

Probably most of the recent striking breakthroughs in designing approximation algorithms with provable performance guarantees are due to using novel methods of rounding polynomially solvable fractional relaxations. Applicability of the known rounding methods is highly dependent on the type of the constraints in such relaxations. In [1] the authors presented a new rounding (pipage) method especially oriented to tackle some NP-hard problems which can be equivalently reformulated as integer programs with cardinality or a bit more general constraints. The paper [1] contains four results demonstrating the strength of the pipage rounding. One of them is an 1/2-approximation algorithm for Max k-Cut with given sizes of parts. An instance of this problem consists of an undirected graph $G = (V, E)$, a collection of nonnegative weights w_e associated with its edges 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

*Basic Research in Computer Science,
Centre of the Danish National Research Foundation.

weight of edges whose ends lie in different parts of the partition. The Max Cut and Max k-Cut problems are classical in combinatorial optimization and have been extensively studied in the absence of cardinality constraints. The best known approximation algorithm for Max Cut is due to Goemans and Williamson [8] and has performance guarantee of 0.878. Frieze and Jerrum [7] extended the technique of Goemans and Williamson to Max k-Cut and designed a $(1-1/k+2\ln k/k^2)$-approximation algorithm. Few approximation algorithms are known for some special cases of Max k-Cut with given sizes of parts. In particular, Frieze and Jerrum [7] present an 0.65-approximation algorithm for Max Bisection (in this problem $k = 2$ and $p_1 = p_2 = |V|/2$). Very recently, Ye [9] announced an algorithm with a better performance guarantee of 0.699. The best known approximation algorithm for Max k-Section (in this problem $p_1 = \cdots = p_k = |V|/k$) is due to Andersson [2] and has performance guarantee of $1 - 1/k + \Theta(1/k^3)$. In this paper we consider a natural hypergraph generalization of Max k-Cut with given sizes of parts — Hypergraph Max k-Cut with given sizes of parts (HMkC for short). An instance of HMkC consists of a hypergraph $H = (V, E)$, a collection of 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 $(X_1, \ldots, X_k)$ with $|X_i| = p_i$ for each i, so as to maximize the total weight of edges of H not lying wholly in any part of the partition (that is, to maximize the total weight of edges S such that $S \setminus X_i \neq \emptyset$ for each i). Several closely related versions of Hypergraph Max k-Cut were studied in the literature but very few results have been obtained. Andersson and Engebretsen [3] presented an 0.72-approximation algorithm for the ordinary Hypergraph Max Cut problem. Arora, Karger and Karpinski [4] designed a PTAS for dense instances of this problem (i.e. in the case of hypergraphs H having $\Theta(|V(H)|^d)$ edges) under the condition that $|S| \leq d$ for each edge S and some constant d.

In this paper by applying the pipage rounding method we prove that HMkC 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: $\lambda_2 = 1/2$, $\lambda_3 = 2/3 \approx 0.666$, $\lambda_4 = 87/128 \approx 0.679$, $\lambda_5 = 84/125 = 0.672$, $\lambda_6 \approx 0.665$ and so on. It is clear that λ_r tends to $1 - e^{-1} \approx 0.632$ as $r \to \infty$. A less trivial fact is that $\lambda_r > 1 - e^{-1}$ for each $r \geq 3$ (Lemma 2 in this paper). Adding up we arrive at the following conclusions: our algorithm finds a feasible cut of weight within a factor of 1/2 on general hypergraphs (we assume that each edge in a hypergraph has size at least 2), and within a factor of $1 - e^{-1}$ in the case when each edge has size at least 3. Note that the first bound coincides with that we obtained in [1] for the case of graphs. In this paper we also show that in the case of hypergraphs without two-vertex edges the bound of $1 - e^{-1}$ cannot be improved unless P=NP.

2 Pipage rounding: a general scheme

We begin with a description of the pipage rounding method [1] in 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 $L(x)$ which is defined and polynomially computable on the same set, coincides with $F(x)$ on binary x satisfying (2)–(3), and the program (which we call *a nice relaxation*)

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

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

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

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

is polynomially solvable. Assume next that the following two main conditions hold. The first — *F/L-lowerbound condition* — states: there exists $C > 0$

such that $F(x)/L(x) \geq C$ 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 $(\{1,\ldots,n\},\{1,\ldots,k\})$ so that $jt \in E(H)$ if and only if x_{jt} is fractional. 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 C of even length. Let M_1 and M_2 be the matchings of H whose union is the cycle C. Define a new solution $x(\varepsilon)$ by the following rule: if jt is not an edge of C, 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 each feasible x and each cycle C 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). Thus we obtain a C-approximation algorithm for P.

3 Pipage rounding: Hypergraph Max k-Cut with given sizes of parts

It is easy to see that an instance of HMkC can be reformulated as the following (nonlinear) integer program:

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

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

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

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

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

We claim that for each feasible x and each cycle C in the graph H (for definitions, see Section 2) the function $\varphi(\varepsilon) = F(x(\varepsilon))$ is a quadratic polynomial with a nonnegative main 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 may 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. Thus the ε-convexity condition holds.

As a nice relaxation we consider the following linear program:

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

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

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

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

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

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

It is easy to see that, given a feasible matrix x, the optimal values of z_S in the above program can be 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).

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*

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

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

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

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 by (21),

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

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. $0 \leq q_S \leq 1$. Again, $z_S = 1$. Let $\mu_t = \sum_{i \in S} x_{it}$ for each t. Note that, by the assumption of the case,

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

and, moreover,

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

By the arithmetic-geometric mean inequality it follows that

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

Consequently,

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

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

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

The corollary states that the F/L-lowerbound condition holds with $C = \min_{S \in E} \lambda_{|S|}$ and thus, by applying the pipage rounding we obtain an algorithm which finds a feasible k-cut whose weight is within a factor of $\min_{S \in E} \lambda_{|S|}$ of the optimum.

Note that $\lambda_2 = 1/2$.

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

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

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

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

Indeed, for any $r \geq 3$,

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

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 $r = 1, 2, \dots$,

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

as required. $\square$

We now show that the integrality gap ratio for the relaxation (15)–(18) can be arbitrarily close to λ_r in the case of r-uniform hypergraphs. It follows that there is no chance to get a better bound by applying any rounding method to this relaxation.

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 ratio 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 $1-e^{-1}$ that our algorithm has on the class of hypergraphs in which each edge has size at least 3, cannot be improved unless $P = NP$.

In the Maximum Coverage problem (MC for short), given a family $\mathcal{F} = \{S_j : j \in J\}$ of subsets of a set $I = \{1, \dots, n\}$ with associated nonnegative weights w_j and a positive integer p, it is required to find a subset $X \subseteq I$ (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 MC 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 provided that P$\neq$NP.

The proof consists in constructing a ratio preserving polynomial-time reduction from MC to HMkC. 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 MC. Construct an instance B of HMkC 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 which is not a member of X, or if $S_j \subseteq X$, by an arbitrary element of I, which is not a member of X. After this transformation and possibly including a few more elements from I to get exactly p, 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 required.

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.

[2] G. Andersson, An approximation algorithm for Max p-Section. Lecture Notes in Computer Science (Proceedings of STACS'99) 1563 (1999) 237–247

[3] G. Andersson and L. Engebretsen, Better approximation algorithms for Set splitting and Not-All-Equal SAT. Inform. Process. Letters 65 (1998) 305–311.

[4] S. Arora, D. Karger, and M. Karpinski, Polynomial Time Approximation Schemes for Dense Instances of NP-Hard Problems. Journal of Computer and System Science 58 (1999) 193–210.

[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.

[6] U. Feige, A threshold of $\ln n$ for approximating set cover. J. of ACM 45 (1998) 634–652.

[7] A. Frieze and M. Jerrum, Improved approximation algorithms for MAX k-CUT and MAX BISECTION. Algorithmica 18 (1997) 67–81.

[8] M. X. Goemans and D. P. Williamson, Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. J. of ACM (1995) 42, 1115–1145.

[9] Y. Ye, A 0.699-approximation algorithm for Max-Bisection, (1999) manuscript.

Recent BRICS Report Series Publications

RS-99-49 **Alexander A. Ageev and Maxim I. Sviridenko.** ***An Approximation Algorithm for Hypergraph Max k-Cut with Given Sizes of Parts.*** **December 1999. 12 pp.**

RS-99-48 **Rasmus Pagh.** ***Faster Deterministic Dictionaries.*** **December 1999. 14 pp. To appear in** ***The Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*****, SODA '00 Proceedings, 2000.**

RS-99-47 **Peter Bro Miltersen and Vinodchandran N. Variyam.** ***Derandomizing Arthur-Merlin Games using Hitting Sets.*** **December 1999. 21 pp. Appears in Beame, editor,** ***40th Annual Symposium on Foundations of Computer Science*****, FOCS '99 Proceedings, 1999, pages 71–80.**

RS-99-46 **Peter Bro Miltersen, Vinodchandran N. Variyam, and Osamu Watanabe.** ***Super-Polynomial Versus Half-Exponential Circuit Size in the Exponential Hierarchy.*** **December 1999. 14 pp. Appears in Asano, Imai, Lee, Nakano and Tokuyama, editors,** ***Computing and Combinatorics: 5th Annual International Conference*****, COCOON '99 Proceedings, LNCS 1627, 1999, pages 210–220.**

RS-99-45 **Torben Amtoft.** ***Partial Evaluation for Designing Efficient Algorithms—A Case Study.*** **December 1999.**

RS-99-44 **Uwe Nestmann, Hans Hüttel, Josva Kleist, and Massimo Merro.** ***Aliasing Models for Mobile Objects.*** **December 1999. To appear in a special FOOL '99 issue of** ***Information and Computation.***

RS-99-43 **Uwe Nestmann.** ***What Is a 'Good' Encoding of Guarded Choice?*** **December 1999. To appear in a special EXPRESS '97 issue of** ***Information and Computation.*** **This revised report supersedes the earlier BRICS report RS-97-45.**

RS-99-42 **Uwe Nestmann and Benjamin C. Pierce.** ***Decoding Choice Encodings.*** **December 1999. To appear in** ***Journal of Information and Computation.***