Angewandte Mathematik und Informatik
Universität zu Köln

Report No. 92.123

Weighted Fractional and Integral $k-$Matching in Hypergraphs

by

Anand Srivastav[1], Peter Stangier

1994

Discrete Applied Maths, February 1995

Anand Srivastav

Institut für Informatik — Lehrstuhl Algorithmen und Komplexität
Humboldt Universität zu Berlin
Unter den Linden 6
10099 Berlin
Germany
srivasta@informatik.hu-berlin.de

Peter Stangier

Institut für Informatik
Universität zu Köln
Pohligstr.1
50969 Köln
Germany
e-mail: stangier@informatik.uni-koeln.de

Weighted Fractional and Integral $k-$Matching in Hypergraphs

Anand Srivastav   Peter Stangier

December 1994

Abstract

We consider the problem of finding polynomial-time approximations of maximal weighted $k-$matchings in a hypergraph and investigate the relationship between the integral and fractional maxima of the corresponding 0-1 integer linear program and its LP-relaxation. We extend results of Raghavan, who gave a deterministic approximation algorithm for unweighted $k-$matching, to the weighted case and compare the so obtained lower bound for the ratio of the integer and fractional maximum with a lower bound of Aharoni, Erdős and Linial.

0   Introduction

The weighted $k-$matching problem in a hypergraph is an interesting generalization of the classical matching problem in graphs. It is stated as follows: Let $H = (V, E)$ be a hypergraph with $|V| = n$, $|E| = m$ and k a positive integer. Let $w_i \geq 0$ be rational weights of the hyperedges, $i = 1, ..., m$. The objective is to find a subset of hyperedges with maximal weight, but with the restriction that no vertex is contained in more than k of these hyperedges. While the 1-matching problem in graphs is well-known to be solvable in polynomial time, finding a maximal weight k-matching in a hypergraph is NP-hard. Closely related to the $k-$matching problem is the k-set covering problem, where the vertices of V have non-negative rational weights and the goal is to find a subset of the vertices with minimal weight, whose intersection with each hyperedge has cardinality at least k. We call the problems unweighted, if all the weights are identical to 1. Let us denote by M_R resp. S_R the fractional and by M_{opt} resp. S_{opt} the integral k-matching resp. k-set covering number. There are two basic questions of combinatorial optimization, which have been investigated for hypergraph matching and set covering in the last years:

1) For which instances of hypergraph matching and set covering a (deterministic) polynomial-time approximation algorithm can be constructed ?

2) What is the relationship between the integral and fractional matching resp. set covering numbers ?

The investigation of the second question has been initiated by the work of Faber and Lovász [9], Lovász [14] and Füredi [10]. Lovász proved in [14] for unweighted 1-set covering the inequality $S_{opt} \leq (1 + \log n)S_R$. Recently this inequality was generalized by Kuzyurin [13] to general integer programming (minimization problem) with non-negative integer datas. Kuzyurin's result implies in particular for weighted k-set covering with non-negative integer weights $S_{opt} \leq (1 + \log kn)S_R$. In the unweighted case of 1-set covering Aharoni, Erdős and Linial [1] improved on the bound of Lovász and obtained to our knowledge the first tight bound for the ratio of the fractional and integral matching number, $\frac{M_{opt}}{M_R} \geq \frac{M_R}{n}$. Recently Füredi, Kahn and Seymour [11] confirmed a conjecture of Füredi [10] and showed for the weighted 1-matching problem with uniform or intersecting hypergraph H, or constant edge weights, the existence of a set of matching edges $\mathcal{M}$

obeying the stronger inequality

$$M_R \leq \sum_{e \in \mathcal{M}} (|e| - 1 + \frac{1}{|e|}) w(e).$$

A different direction of research was undertaken by Raghavan and Thompson [19], who gave in the unweighted case assuming that $k \geq 6 \ln n$ a probabilistic approximation algorithm finding a $k-$matching M such that $M \geq (1 - \delta) M_{opt} - O(\sqrt{(1 - \delta) M_R})$, with some $\delta \in (0, \frac{1}{2})$. Later Raghavan introduced the concept of pessimistic estimators, extending the derandomization technique of conditional probabilities, and transformed this probabilistic result into a deterministic algorithm with nearly the same approximation guarantee as achieved by the probabilistic algorithm [20].

But for $k-$matching with *rational* weights the problem of finding polynomial-time approximation algorithms remained open. The purpose of this paper is to contribute to the solution of this approximation problem.

Since the $k-$matching problem in hypergraphs is strongly NP-hard, there cannot exist an arbitrarily good fully polynomial-time approximation algorithm [18]. Furthermore the results of Arora et al [5] raise the conjecture that for arbitrary instances of weighted $k-$matching even a polynomial-time approximation scheme might be out of reach. Nevertheless, not all instances must be intractable for approximation. We will exhibit a large class of instances of the weighted $k-$matching problem for which tight polynomial-time approximation algorithms do exist. Our main result is

Theorem *(Corollary 2.6, 2.7 and 2.8)*

(i) *Let $0 < \epsilon < 1$ and let the edge weights be in $[0, 1]$. Suppose that $k \geq \frac{24 \ln n}{\epsilon^2}$ and k edges have total weight at least $18 \epsilon^{-2}$. Then with a derandomized algorithm we can find in polynomial-time a $k-$matching M such that $M \geq (1 - \epsilon) M_{opt}$.*

(ii) *In the unweighted case the algorithm gives for all instances with $k \geq \frac{24 \ln n}{\epsilon^2}$ a k-matching M such that $M \geq M_{opt}(1 - \epsilon)$.*

Furthermore we observe that the arguments of Aharoni, Erdős and Linial [1] give for k-matching the inequality $\frac{M_{opt}}{M_R} \geq \frac{M_R}{nk^2}$ and show that the lower bound for $\frac{M_{opt}}{M_R}$ of the randomized approach is better than this "combinatorial" lower bound. The interesting open question arising here is whether or not the stronger inequality $\frac{M_{opt}}{M_R} \geq \frac{M_R}{nk}$ holds.

The essential methods we use are randomized rounding and derandomization. While randomized rounding can be performed as in [19], derandomization causes the main computational difficulties. The problem in the case of rational weights is that the basic method of conditional probabilities/pessimistic estimators necessarily requires the computation of the exponential function on the RAM model of computation. We solve this problem constructing a new class of pessimistic estimators for the conditional probabilities under consideration, which can be derived from Mc-Diarmids [15] proof of the Angluin-Valiant inequality on deviation of weighted sums of Bernoulli trials from their mean. In [23] we gave a comprehensive analysis of this approach and showed algorithmic counterparts of the classical large deviation inequalities on Binomial type distributions due to Bernstein, Chernoff, Hoeffding and Angluin/Valiant. These algorithmic inequalities can be considered as an implementation of the conditional probability method of Erdős, Selfridge [8], and Spencer [2] and the pessimistic estimator technique of Raghavan [20] on the RAM model of computation.

The total running time of our algorithm is the sum of the (dominating) time to solve the linear programming relaxation of the integer program associated to the weighted hypergraph k-matching problem *and* the time of derandomized rounding. A direct application of the results in [23] would imply for weighted k-matching in hypergraphs an $O(nm^2 \log \frac{mn}{\epsilon^2})$-time derandomized rounding algorithm. In this paper we show that at least for the weighted hypergraph matching problem

derandomized rounding only needs $O(m^2 \log m + mn)$-time, so we have a strongly-polynomial time rounding algorithm. At this moment we do not have a LP algorithm for hypergraph matching, which matches the nearly quadratic running time of the rounding procedure. This motivates in further work the search for a *fast* strongly polynomial-time LP algorithm for the hypergraph matching problem.

The model of computation throughout this paper is the RAM-model (see [16]). It can be briefly described as follows. By the size of an input we mean the number of data entries in the description of the input, while the encoding length of the input is the maximal binary encoding length of numbers in the input. In the RAM-model an algorithm runs in polynomial-time (resp. strongly polynomial-time), if the number of elementary arithmetic operations (briefly called running time) is polynomially bounded in the size and the encoding length of the input (resp. *only* in the size of the input) *and* the maximal binary encoding length of a number appearing during the execution of the algorithm (briefly called space) is polynomially bounded in the size and encoding length of the input. In the following let L denote the encoding length of the edge weights, let *log* be the binary and *ln* the natural logarithm.

1   Randomized Approximation

The basic randomized algorithm for $k-$matching was introduced by Raghavan and Thompson [19] and consists of essentially two steps: randomized rounding and scaling down the probability of setting the variable x_i to 1 by a factor of $1 - \frac{\epsilon}{2}$ for all $i = 1, \ldots, m$.

Algorithm $P-$HYPERMATCH

Input:

Hypergraph $H = (V, E)$ with $|V| = n$, $|E| = m$, edge weights $w : E \to [0, 1] \cap \mathbb{Q}_0^+$, the vertex-edge incidence matrix A of H and a positive integer k.

Algorithm:

1) (LP-Relaxation)
Solve the linear program

$$\max \{\sum_{i=1}^{m} w_i x_i; A x \leq k, x \in [0, 1]^m\}$$

with rational solution vector $\tilde{x} \in [0, 1]^m$.

2) (Scaling)
Choose $\epsilon \in [0, 1]$ and replace $\tilde{x}$ by $(1 - \frac{\epsilon}{2})\tilde{x}$

3 (Randomized Rounding)
For $i = 1, \ldots, m$ set independently $x_i = 1$ with probability $(1 - \frac{\epsilon}{2})\tilde{x}_i$ and $x_i = 0$ with probability $1 - (1 - \frac{\epsilon}{2})\tilde{x}_i$

4) Output the vector $x = (x_1, \ldots, x_m) \in \{0, 1\}^m$

$\square$

Let us denote by M_{opt} the optimal value of the weighted $k-$matching problem, M_R the optimal value of its LP-Relaxation and by $M(x) := \sum_{i=1}^{m} w_i x_i$ $(x \in [0, 1]^m)$ the objective function. Linear Programming gives a rational solution vector $\tilde{x} \in [0, 1]$ whose encoding length is a polynomial in L, n and m, so the encoding length of M_R is also polynomially bounded. Let L' be the maximum of L and the encoding length of M_R. Since L' will appear only in the encoding length of numbers we have to compute in our algorithms, but has no influence on the running time, we neglect the exact degree of the polynomial bounding L'.

Raghavan and Thompson [19] analysed the algorithm P-HYPERMATCH in the unweighted case ($w \equiv 1$) and showed that for a certain scaling factor $0 < \epsilon < 1$ the algorithm finds a

3

$k-$matching M with $M \geq (1 - \frac{\epsilon}{2})M_{opt} - O(\sqrt{(1 - \frac{\epsilon}{2})M_R})$, provided that $k \geq 6 \ln n$. It is easy to check that their proof is also valid in the weighted case, so we obtain the same result. But given an $\epsilon \in (0, 1)$ we are interested in an approximation $M \geq (1 - \epsilon)M_{opt}$, because such a statement explicitly shows instances of the problem, where an arbitrary or at least in ϵ measurable approximation is possible. This can be proved when k is at least $\frac{24 \ln n}{\epsilon^2}$.

Lemma 1.1 (Angluin/Valiant[4]. For a proof see [15]) *Let $\psi_1, \ldots, \psi_m$ be independent random variables, with $0 \leq \psi_j \leq 1$ and $E(\psi_j) = p_j$ for all $j = 1, \ldots, m$. Let $\psi = \frac{1}{m}\sum_{j=1}^{m}\psi_j$, $p = \frac{1}{m}\sum_{j=1}^{m}p_j$ and $0 < \beta \leq 1$. Then*

(i) $Prob(\psi > (1 + \beta)mp) \leq exp(-\frac{\beta^2 mp}{3})$

(ii) $Prob(\psi < (1 - \beta)mp) \leq exp(-\frac{\beta^2 mp}{2})$.

Theorem 1.2 *Let $\epsilon \in (0, 1)$, $k \geq \frac{24 \ln n}{\epsilon^2}$ and $M_R \geq \frac{18}{\epsilon^2}$. Then $P-HYPERMATCH$ finds a weighted $k-$matching M such that $M \geq (1 - \epsilon)M_{opt}$ with probability at least 0.73.*

Proof. Let $n \geq 8$ (otherwise solve the problem by enumeration) and run $P-HYPERMATCH$ with output vector $x \in \{0, 1\}^m$. We first show the following two inequalities:

(a) $Prob(\sum_{j=1}^{m}a_{ij}x_j > k) \leq \frac{1}{8n}$ for all $i = 1, \ldots, n$.

(b) $Prob(M(x) < (1 - \frac{\epsilon}{2})M_{opt} - 2\sqrt{(1 - \frac{\epsilon}{2})M_R}) \leq e^{-2}$.

With $n \geq 8$ and the assumption on k we have

$$\ln 8n \leq \frac{k\epsilon^2}{12(1 - \frac{\epsilon}{2})}. \tag{1}$$

Choose $\beta := \frac{\epsilon}{2 - \epsilon}$. Trivially $0 < \beta < 1$ and the Angluin-Valiant inequality (Lemma 1.1 (i)) and (1) prove (a):

$$Prob(\sum_{j=1}^{m}a_{ij}x_j > k) = Prob(\sum_{j=1}^{n}a_{ij}x_j > (1 + \beta)(1 - \frac{\epsilon}{2})k) \leq \frac{1}{8n}.$$

With $\beta_0 := \sqrt{\frac{4}{(1 - \frac{\epsilon}{2})M_R}}$ we may assume that $0 < \beta_0 \leq 1$, because if $\beta_0 > 1$, we have the zero probability event "$M < 0$". Finally $M_R \geq \frac{18}{\epsilon^2}$ implies $(1 - \frac{\epsilon}{2})M_R - \sqrt{4(1 - \frac{\epsilon}{2})M_R} \geq (1 - \epsilon)M_R$. $\square$

Corollary 1.3 *Let $w \equiv 1$, $\epsilon \in (0, 1)$ and $k \geq \frac{24 \ln n}{\epsilon^2}$. Then $P-HYPERMATCH$ finds a $k-$matching M such that $M \geq (1 - \epsilon)M_{opt}$ with probability at least 0.73.*

Proof. We may assume that $n \geq 4$. This implies $\frac{18}{\epsilon^2} \leq \frac{24 \ln n}{\epsilon^2} \leq k$. But in the unweighted case we always have $M_R \geq M_{opt} \geq k$, hence Theorem 1.2 proves Corollary 1.3. $\square$

Remark We assumed that k is at least $\frac{24 \ln n}{\epsilon^2}$ which differs by the $4\epsilon^{-2}$ factor from Raghavan and Thompson's assumption on k. Note that Theorem 1.2 can be proved under less restrictive assumptions on k, if we accept an only exponentially small success probability for the algorithm P-HYPERMATCH. But even in that case the probabilistic analysis requires $k = \Omega(\ln n)$. Furthermore we assumed in the weighted case $M_R \geq \frac{18}{\epsilon^2}$. We saw in the unweighted case that this condition is automatically satisfied (Corollary 1.3). Again if we allow small success probabilities, we obtain less restrictive assumptions, but even then M_R must be greater than 4. This is due to the probabilistic analysis, in particular to the Angluin-Valiant inequality: The probabilistic algorithm guarantees $M \geq (1 - \frac{\epsilon}{2})M_R - 2\sqrt{(1 - \frac{\epsilon}{2})M_R}$. The right hand side must be positive, otherwise the algorithm does not give any guarantee. In the best case ϵ is zero and the lower bound for M is positive if $M_R > 4$. Let us proceed to the main problem, the derandomization of the results above.

4

2 Derandomized Algorithms

We briefly sketch the derandomization idea of Spencer [2]. Let $\epsilon \in (0,1)$ and let $k \geq \frac{24 \ln n}{\epsilon^2}$. Denote by E_ϵ^c the event

$$\text{``}\exists i \quad (Ax)_i > k \quad \text{or} \quad M(x) < (1-\epsilon)M_{opt}\text{''}.$$

If $M_R \geq \frac{18}{\epsilon^2}$ Theorem 1.2 gives $\text{Prob}(E_\epsilon^c) \leq 0.27$. The method of conditional probabilities seeks for a vector for which the event E_ϵ holds, sequentially selecting the values of the x_i's from $\{0,1\}$ by minimizing the conditional probability that E_ϵ^c will occur, if x_i is chosen to be 0 (resp. 1):

Algorithm WALK(E_ϵ^c)

(a) Initial Step ($l = 1$)
Set $x_1 = z_1$, where z_1 minimizes the function

$$z \to \text{Prob}(E_\epsilon^c | z), \quad z \in \{0,1\}.$$

(b) Induction step ($l+1$)
If $x_1 \ldots, x_l$ have been selected, set $x_{l+1} = z_{l+1}$, where z_{l+1} minimizes the function

$$z \to \text{Prob}(E_\epsilon^c | x_1 \ldots, x_l, z), \quad z \in \{0,1\}.$$

(c) Stop, when $l = m$

$\square$

The striking observation is that the output vector x satisfies the event E_ϵ, because the inequalities $1 > \text{Prob}(E_\epsilon^c) \geq \text{Prob}(E_\epsilon^c | x_1) \geq \ldots \geq \text{Prob}(E_\epsilon^c | x_1 \ldots, x_m)$ and the fact that $\text{Prob}(E_\epsilon^c | x_1, \ldots, x_m)$ is either zero or one imply $\text{Prob}(E_\epsilon^c | x_1, \ldots, x_m) = 0$. The WALK procedure is a deterministic algorithm, but to run the algorithm on the usual finite machine models of computation, like the RAM model or the Turing machine model, we must be able to compute the conditional probabilities $\text{Prob}(E_\epsilon^c | x_1, \ldots, x_l)$. The complexity of the computation of these conditional probabilities determines the time complexity of the algorithm. Unfortunately, there is no general way of computing the conditional probabilities. Raghavan circumvented this obstacle in some examples constructing easier computable upper bounds for the conditional probabilities, the so called pessimistic estimators, which mimic the role of the conditional probabilities.

Definition 2.1 (Pessimistic Estimator, [20]) *Let $U(E_\epsilon^c)$ denote a sequence of functions $U_1(x_1), U_2(x_1, x_2), \ldots, U_m(x_1, \ldots, x_m)$ satisfying the following properties:*

(i) $Prob(E_\epsilon^c | x_1, \ldots, x_l) \leq U_l(x_1, \ldots, x_l)$ for all $x_1, \ldots, x_l \in \{0,1\}, l \leq m$.

(ii) $U_{l+1}(x_1, \ldots, x_l, x_{l+1}) \leq U_l(l_1, \ldots, x_l)$ given any $x_1, \ldots, x_l \in \{0,1\}$ for some $x_{l+1} \in \{0,1\}$.

(iii) $U_1(x_1) < 1$ for some $x_1 \in \{0,1\}$.

(iv) $U_l(x_1, \ldots, x_l)$ can be computed on the RAM model of computation in time bounded by a polynomial in n, m and $\log \frac{1}{\epsilon}$ for each l.

Then $U(E_\epsilon^c)$ is called a pessimistic estimator for the event E_ϵ^c.

If we replace the conditional probabilities in the WALK procedure by the functions of the pessimistic estimator, we get indeed a polynomial-time algorithm finding the desired vector x.

Remark: Raghavan [20] constructed for *unweighted* $k-$matching a family of functions, which satisfies the conditions (i)–(iii) of Definition 2.1. But his approach raises two computational problems: The computation of the pessimistic estimator requires the computation of exponential terms of the form s^{w_j}, where s is a real number, $s \geq 1$.

(a) In case of rational edge weights w_j the term s^{w_j} cannot be computed on the RAM model in polynomial-time.

(b) In the unweighted case $w_j \in \{0, 1\}$ is an integer and s^{w_j} is computable iff s is computable. In [20] (see Theorem 7) s was defined to be $s := D((1 - \frac{\epsilon}{2})M_R, \frac{1}{n}) - 1$, where $D((1 - \frac{\epsilon}{2})M_R, \frac{1}{n})$ is a positive root of the equation

$$z - (1 - z)\ln(1 + z) + \frac{\ln n}{(1 - \frac{\epsilon}{2})M_R} = 0.$$

But it is not known, how to find a root of this analytic equation in polynomial-time.

While the second problem is only a minor technical obstacle, as we will show using parameters defined in McDiarmid's proof of the Angluin-Valiant inequality, the presence of rational edge weights w_j cause the more serious problem, which requires some work. We follow the proof of the Angluin-Valiant inequality of McDiarmid [15] and derive upper bounds on the conditional probabilities. Then we show that these upper bounds can be replaced by $O(m^2 \log m)$-degree polynomials evaluated at a rational number depending on the edge weights.

The following "conditional probability" formulation of the Angluin-Valiant inequality can be extracted from the proof of corollary 5.1 and 5.2 in [15].

Lemma 2.2 *Let $a_1, \ldots, a_m$ be real numbers with $0 \leq a_j \leq 1$ for each j and let $\psi_1, \ldots, \psi_m$ be independent $0 - 1$ valued random variables. Let $\tilde{p}_j = E(\psi_j)$,*
$\tilde{q}_j = 1 - \tilde{p}_j$, $\psi = \sum_{j=1}^{m} a_j \psi_j$, $p = \frac{1}{m}E(\psi)$, $q = 1 - p$ and $0 < \beta < 1$.
Define $s^+ = \frac{q(1+\beta)}{q - p\beta}$, $s^- = \frac{q + p\beta}{q(1-\beta)}$ and for $1 \leq l \leq m$ let $x_1, \ldots, x_l \in \{0, 1\}$.
Then we have

$$(i) \quad Prob(\psi \ > \ (1 + \beta)mp | \psi_1 = x_1, \ldots, \psi_l = x_l)$$
$$\leq \quad e^{-(1+\beta)pm \ln s^+} e^{\sum_{j=1}^{l} a_j x_j \ln s^+} \prod_{j=l+1}^{m} [\tilde{p}_j e^{a_j \ln s^+} + 1 - \tilde{p}_j]$$

$$(ii) \quad Prob(\psi \ < \ (1 - \beta)mp | \psi_1 = x_1, \ldots, \psi_l = x_l)$$
$$\leq \quad e^{(1-\beta)pm \ln s^-} e^{-\sum_{j=1}^{l} a_j x_j \ln s^-} \prod_{j=l+1}^{m} [\tilde{p}_j e^{-a_j \ln s^-} + 1 - \tilde{p}_j].$$

$\square$

Lemma 2.2 motivates the definition of the basic functions for the construction of the pessimistic estimator $U(E_\epsilon^c)$. In the following let $y_1, \ldots, y_m$ denote the scaled variables $y_j := (1 - \frac{\epsilon}{2})\tilde{x}_j$. (Recall that $(\tilde{x}_j)$ was the solution of the LP-Relaxation (see Algorithm P−HYPERMATCH)). Before we define the events of interest, we choose the deviation factors β so that s always will be a rational number. With binary search in the interval $[0, \sqrt{\frac{8}{(2-\epsilon)M_R}}]$ we can find a rational number β_0 with $0 \leq \beta_0 - \sqrt{\frac{8}{(2-\epsilon)M_R}} \leq \frac{2}{m}$ in $O(\log m)$ steps. Since $M_R \leq m \leq m^2$, we get

$$\sqrt{\frac{8}{(2 - \epsilon)M_R}} \leq \beta_0 \leq \sqrt{\frac{9}{(2 - \epsilon)M_R}}. \tag{2}$$

(a) The event "$M < (1 - \beta_0)(1 - \frac{\epsilon}{2})M_R$ "
Let β_0 be as in (2), $p_0 = \frac{(1 - \frac{\epsilon}{2})M_R}{m}$, $q_0 = 1 - p_0$ and $s_0 = \frac{q_0 + p_0 \beta_0}{q_0(1 - \beta_0)}$.

6

For $l = 0$ define

$$V_0^{(0)} = e^{(1-\beta_0)p_0 m \ln s_0} \prod_{j=1}^{m} [y_j e^{-w_j \ln s_0} + 1 - y_j] \tag{3}$$

For $l \geq 1$ and $z_1, \ldots, z_l \in \{0, 1\}$ define

$$V_l^{(0)} = e^{(1-\beta_0)p_0 m \ln s_0} e^{-\sum_{j=1}^{l} w_j z_j \ln s_0} \prod_{j=l+1}^{m} [y_j e^{-w_j \ln s_0} + 1 - y_j] \tag{4}$$

(b) The event "$\exists i \quad (Ax)_i > k$"

Let $1 \leq i \leq n$ be arbitrary. Let $p = \frac{(1-\frac{\epsilon}{2})k}{m}$, $q = 1 - p$, $\beta = \frac{\epsilon}{2-\epsilon}$ and $s = \frac{q(1+\beta)}{q-p\beta}$.

For $l = 0$ define

$$V_0^{(i)} = e^{-(1+\beta)pm \ln s} \prod_{j=1}^{m} [y_j e^{a_{ij} \ln s} + 1 - y_j] \tag{5}$$

For $l \geq 1$ and $z_1, \ldots, z_l \in \{0, 1\}$ define

$$V_l^{(i)}(z_1, \ldots, z_l) = e^{-(1+\beta)pm \ln s} e^{\sum_{j=1}^{l} a_{ij} z_j \ln s} \prod_{j=l+1}^{m} [y_j e^{a_{ij} \ln s} + 1 - y_j]. \tag{6}$$

(c) The event "$M < (1 - \beta_0)(1 - \frac{\epsilon}{2})M_R$ or $\exists i \quad (Ax)_i > k$"

Let us denote by E_ϵ^c the event defined in (c). Let for $l \geq 1$

$$V_l(z_1, \ldots, z_l) := \sum_{i=0}^{n} V_l^{(i)}(z_1, \ldots, z_l) \tag{7}$$

and for $l = 0$

$$V_0 := V_0^{(0)} + \sum_{i=1}^{n} V_0^{(i)}. \tag{8}$$

We first show that the functions V_l satisfy the condition (i)–(iii) of Definition 2.1.

Lemma 2.3 *We have for each integer l, $1 \leq l \leq m$*

(i) $\mathrm{Prob}(E_\epsilon^c | x_1, \ldots, x_l) \leq V_l(x_1, \ldots, x_l)$ for all $x_1, \ldots, x_l \in \{0, 1\}$.

(ii) $V_{l+1}(x_1, \ldots, x_l, x_{l+1}) \leq V_l(x_1, \ldots, x_l)$ given any $x_1, \ldots, x_l$ for some $x_{l+1} \in \{0, 1\}$.

(iii) $V_1(x_1) \leq 0.27$ for some $x_1 \in \{0, 1\}$.

Proof.
(i) The inequality $\mathrm{Prob}(E_\epsilon^c | x_1, \ldots, x_l) \leq V_l(x_1, \ldots, x_l)$ follows from Lemma 2.2.

(ii) Let $l \geq 0$ and let $z \in \{0, 1\}$. For $i = 0$ define

$$f_l^{(0)}(z) := V_l^{(0)}(x_1, \ldots, x_l)[y_{l+1} e^{-w_{l+1} \ln s_0} + 1 - y_{l+1}]^{-1} e^{-z w_{l+1} \ln s_0}$$

and for $i \geq 1$

$$f_l^{(i)}(z) := V_l^{(i)}(x_1, \ldots, x_l)[y_{l+1} e^{a_{i,l+1} \ln s} + 1 - y_{l+1}]^{-1} e^{z a_{i,l+1} \ln s}.$$

7

Let
$$f_l(z) := \sum_{i=0}^{n} f_l^{(i)}(z).$$

Then we have with x_{l+1} being the minimizer of $z \mapsto f_l(z)$

$$
\begin{aligned}
V_l(x_1, \ldots, x_l) &= y_{l+1} f_l(1) + (1 - y_{l+1}) f_l(0) \\
&\geq f_l(x_{l+1}) \\
&= V_{l+1}(x_1, \ldots, x_{l+1})
\end{aligned}
$$

(iii)

$$V_1(x_1) = y_1 f_0(1) + (1 - y_1) f_0(0) = \sum_{i=0}^{n} V_0^{(i)}.$$

From Theorem 1.2 and $\sqrt{\frac{8}{(2-\epsilon) M_R}} \leq \beta_0$ (see (2)) it follows that

$$V_0^{(0)} \leq \exp(-\frac{1}{2}\beta_0^2 p_0 m) \leq e^{-2}.$$

For $i \geq 1$ we observe

$$
\begin{aligned}
y_j e^{a_{ij} \ln s} + 1 - y_j &= 1 + y_j(e^{a_{ij} \ln s} - 1) \\
&\leq 1 + y_j a_{ij}(e^{\ln s} - 1).
\end{aligned}
$$

Since $\sum_{j=1}^{m} a_{ij} y_j \leq (1 - \frac{\epsilon}{2}) k = mp$ we have

$$V_0^{(i)} \leq \exp(-\frac{1}{3}\beta^2 pm) \leq \frac{1}{8n}.$$

The last inequality follows from our assumption $k \geq \frac{24 \ln n}{\epsilon^2}$. Hence for some $x_1 \in \{0, 1\}$

$$V_1(x_1) = \sum_{i=0}^{n} V_0^{(i)} \leq \frac{1}{8} + e^{-2} \leq 0.27.$$

$\square$

Let us consider the l-th step ($l \geq 1$) of the algorithm WALK(E_ϵ^c). We wish to compute the function $V_l(x_1, \ldots, x_l)$. Now $V_l(x_1, \ldots, x_l) = \sum_{i=0}^{n} V_l^{(i)}(x_1, \ldots, x_l)$. If $i \geq 1$, then

$$V_l^{(i)}(x_1, \ldots, x_l) = s^{k + \sum_{j=1}^{l} a_{ij} x_j} \prod_{j=l+1}^{m} [y_j s^{a_{ij}} + 1 - y_j].$$

Since $x_j, a_{ij} \in \{0, 1\}$ and $s = \frac{q(1+\beta)}{q - p\beta}$ is a rational number by definition of p and β, $V_l^{(i)}(x_1, \ldots, x_l)$ is efficiently computable on the RAM model of computation. So the only problem is the computation of $V_l^{(0)}(x_1, \ldots, x_l)$, in particular the handling of the terms $s_0^{w_j}$ for rational edge weights w_j. We show how to approximate $V_l^{(0)}(x_1, \ldots, x_l)$ by a polynomial of degree $O(m^2 \log m)$ We need a technical lemma which is a special case of Lemma 2.9 in [23].

Lemma 2.4 *Let $A_1, \ldots, A_m, B, \gamma$ be rational numbers with encoding length at most L, $B \geq 1$ and $0 < \gamma \leq 1$. Let $0 < \epsilon \leq 1$ and suppose that $\sum_{i=1}^{m} |A_i| \leq \alpha_1 m$ and $b \leq \frac{\alpha_2 n}{\epsilon}$ for some non-negative constants α_1, α_2. Let $N = 18 \lceil \alpha_1 m \rceil \lceil \log \alpha_2 m \rceil + 2m + \lceil \log_3 \frac{m+1}{\gamma} \rceil$ and let T_N be the N-th degree Taylor polynomial of the exponential function.*

8

(i) *Then a rational number b approximating $\ln B$ and for each $i = 1, \ldots, m$ the number $T_N(A_i b)$ can be computed in $O(m \log(\frac{m}{\gamma \epsilon}))$-time such that*

$$|\prod_{i=1}^{m} e^{A_i \ln B} - \prod_{i=1}^{m} T_N(A_i b)| \leq \gamma$$

uniformly for all A_i as above.

(ii) *The encoding length of $T_N(A_i b)$ is $O(L[m \log(\frac{m}{\epsilon \gamma})]^2)$.*

We are ready to prove the main results.

Theorem 2.5 *Let $\epsilon \in (0, 1)$, $k \geq \frac{24 \ln n}{\epsilon^2}$ and $M_R \geq \frac{18}{\epsilon^2}$. Given the optimal LP solution $\tilde{x} \in [0, 1]^m$, derandomization finds a k-matching M such that $M \geq (1 - \epsilon) M_R$ in $O(m^2 \log m + mn)$-time.*

Proof. We approximate the function $V_l^{(0)}(x_1, \ldots, x_l)$ in order to define the pessimistic estimator. Given $x_1, \ldots, x_l \in \{0, 1\}$ set $Z_j = -w_j X_j$ for $j \geq l + 1$ and $Z_l = (1 - \beta_0) p_0 m - \sum_{i=1}^{l} x_i w_i$. Then

$$V_l^{(0)}(x_1, \ldots, x_l) = \prod_{j=l}^{m} \mathbb{E}(e^{Z_j \ln s_0}).$$

We will use Lemma 2.4. Since $M_R \geq \frac{18}{\epsilon^2}$, we have $\beta_0 \leq \frac{m-1}{m}$, which implies $s_0 \leq \frac{4m}{\epsilon}$. Set $\alpha_1 = 2$, $\alpha_2 = 4$ and $\gamma = \frac{1}{8(4m-1)}$ and let N be as in Lemma 2.4. Let T_N be the Taylor polynomial of the exp-function with degree N. By Lemma 2.4 the approximation of $\ln s_0$ by a rational number b as well as the approximation of $\prod_{j=l}^{m} e^{Z_j \ln s_0}$ can be done uniformly for all $x_1, \ldots, x_l$, hence we have

$$||\prod_{j=l}^{m} e^{Z_j \ln s_0} - \prod_{j=l}^{m} T_N(Z_j b)||_\infty \leq \gamma.$$

This implies taking expectation and using the independence of the random variables Z_j for $j \geq l+1$

$$|V_l^{(0)}(x_1, \ldots, x_l) - \prod_{j=l}^{m} \mathbb{E}(T_N(Z_j b))| \leq \gamma.$$

Set

$$T_l^{(0)}(x_1, \ldots, x_l) := \prod_{j=l}^{m} \mathbb{E}(T_N(Z_j b))$$

and let $U_l(x_1, \ldots, x_l)$ be the functions defined by

$$U_l(x_1, \ldots, x_l) := T_l^{(0)}(x_1, \ldots, x_l) + \sum_{i=1}^{n} V_l^{(i)}(x_1, \ldots, x_l) + 2(2m - l)\gamma. \tag{9}$$

We show that this family of functions defines a pessimistic estimator for the event E_ϵ^c.

Condition (i): With Lemma 2.3 (i)

$$Prob(E_\epsilon^c | x_1, \ldots, x_l) \leq \sum_{i=1}^{n} V_l^{(i)}(x_1, \ldots, x_l) + V_l^{(0)}(x_1, \ldots, x_l).$$

But

$$V_l^{(0)}(x_1, \ldots, x_l) \quad \leq \quad T_l^{(0)}(x_1, \ldots, x_l) + \gamma.$$

9

Hence $Prob(E_\epsilon^c | x_1, \ldots, x_l) \leq U_l(x_1, \ldots, x_l)$ for all $x_1, \ldots, x_l$.

Condition (ii): With Lemma 2.3(ii) and the fact

$$|V_l^{(0)}(x_1, \ldots, x_l) - T_l^{(0)}(x_1, \ldots, x_l)| \leq \gamma,$$

it is straight forward to prove the inequality

$$U_{l+1}(x_1, \ldots, x_{l+1}) \leq U_l(x_1, \ldots, x_l)$$

for some x_{l+1} given any $x_1, \ldots, x_l$.

Condition (iii): We show $U_1(x_1) \leq \frac{1}{2}$ for some $x_1 \in \{0, 1\}$. Using Lemma 2.3 (ii) we have for some x_1 equals either 0 or 1

$$
\begin{aligned}
U_1(x_1) &= \sum_{i=1}^{n} V_1^{(i)}(x_1) + T_1^{(0)}(x_1) + 2(2m-1)\gamma \\
&\leq \sum_{i=0}^{n} V_1^{(i)}(x_1) + \frac{1}{8(4m-1)} + \gamma + 2(2m-1)\gamma \\
&\leq \frac{1}{2}.
\end{aligned}
$$

The computation of the running time goes as follows:

Let us first consider the approximation of $V_l^{(0)}(x_1, \ldots, x_l)$. Note that

$$V_l^{(0)}(x_1, \ldots, x_l) = V_{l-1}^{(0)}(x_1, \ldots, x_{l-1}) \frac{1}{\mathbb{E}(e^{Z_l \ln s_0})} e^{-w_l x_l \ln s_0}. \tag{10}$$

By Lemma 2.4 we can compute each $\mathbb{E}(T_N(Z_j b))$ in $O(m \log(\frac{m}{\gamma \epsilon}))$-time. In the first step of the WALK procedure $(l = 1)$ we have to approximate $\mathbb{E}(\prod_{j=1}^{m} e^{Z_j \ln s_0})$ for some $x_1 \in \{0, 1\}$, and this requires the computation of m polynomials, therefore $O(m^2 \log(\frac{m}{\gamma \epsilon}))$-time. But according to the recursion (10), in the forthcoming steps $(l \geq 2)$ we have to do only one update computing two polynomials, which can be done in $O(m \log(\frac{m}{\gamma \epsilon}))$-time per step, hence summing up over all the m steps we need $O(m^2 \log \frac{m}{\gamma \epsilon})$-time. Now for each $i \geq 1$ the total computation time of $V_l^{(i)}(x_1, \ldots, x_l)$ over all m steps is $O(m)$, using a recursion argument as above and the fact that we can do exact computations. As i runs from 1 to n we have for the computation of all the nm numbers $V_l^{(i)}(x_1, \ldots, x_l)$ a total time of $O(nm)$, hence the total time of the rounding algorithm is $O(m^2 \log(\frac{m}{\gamma \epsilon}) + nm)$. We can assume that $\epsilon \geq \frac{1}{m}$. Otherwise we would get $k \geq 24m^2 \ln m$, and the hypergraph k-matching problem would become trivial. This gives us together with $\gamma = \frac{1}{8(4m-1)}$ the total time of $O(m^2 \log m + nm)$.

$\square$

Corollary 2.6 *Let $\epsilon \in (0, 1)$ and $k \geq \frac{24 \ln n}{\epsilon^2}$. Let $w_1 \geq \ldots \geq w_m$ be the edge weights with $w_1 + \ldots + w_k \geq \frac{18}{\epsilon^2}$. Then with linear programming and derandomization we can find in polynomial-time a k-matching M with $M \geq (1 - \epsilon)M_{opt}$.*

Proof. $w_1 + \ldots + w_k \geq \frac{18}{\epsilon^2}$ implies $M_R \geq \frac{18}{\epsilon^2}$ and Theorem 2.5 gives $M \geq (1 - \epsilon)M_{opt}$.

$\square$

In the unweighted case we have

Corollary 2.7 *Let $w \equiv 1$, $\epsilon \in (0, 1)$ and $k \geq \frac{24 \ln n}{\epsilon^2}$. Then with linear programming and derandomized rounding we can find in polynomial-time a k-matching with $M \geq (1 - \epsilon)M_{opt}$.*

Proof. $k \geq \frac{24 \ln n}{\epsilon^2}$ implies $M_R \geq \frac{18}{\epsilon^2}$ and Theorem 2.5 proves Corollary 2.7. $\quad\square$

For arbitrary weighted $k-$matching Theorem 2.5 implies

Corollary 2.8 *Let $\epsilon \in (0,1)$ and $k \geq \frac{24 \ln n}{\epsilon^2}$. Let $w_1 \geq \ldots \geq w_m$ be the edge weights with $w_1 > 1$ with $w_1 + \ldots + w_k \geq \frac{18 w_1}{\epsilon^2}$. Then with linear programming and derandomization we can find in polynomial-time a k-matching M with $M \geq (1-\epsilon)M_{opt}$.*

Remark In [1] Aharoni, Erdős, Linial proved for unweighted $1-$matching in hypergraphs that $M_{opt} \geq \frac{M_R^2}{n}$. An examination of their proof shows that a similar inequality holds for k-matching:

$$M_{opt} \geq \frac{M_R^2}{k^2 n}. \tag{11}$$

This can be seen as follows. Let $d := \min \{|e_i|; \ i = 1,\ldots,m; \ e_i \in E\}$ be the minimal edge cardinality. The proof of Theorem 2 in [1] shows for every vector $x \in \mathbb{R}^m$, $x \geq 0$,

$$x^T A^T A x \geq (\sum_{i=1}^{m} x_i)^2 \left[\frac{(d-1)}{m} + \frac{1}{M_{opt}^{(1)}} \right], \tag{12}$$

where $M_{opt}^{(1)}$ is the maximal unweighted 1-matching number. Taking the fractional unweighted k-matching vector associated to M_R, inequality (12) implies $M_{opt} \geq \frac{M_R^2}{k^2 n}$. A comparison of the "randomized" lower bound for $\frac{M_{opt}}{M_R}$ and the bound in (11) shows that the randomized result is much better. By the proof of Theorem 2.1 we have

$$M_{opt} \geq (1 - \frac{\epsilon}{2})M_R - \sqrt{4(1 - \frac{\epsilon}{2})M_R}. \tag{13}$$

Let $R_c := \frac{M_R}{nk^2}$ and $R_p := (1 - \frac{\epsilon}{2}) - 2\sqrt{\frac{(1-\frac{\epsilon}{2})}{M_R}}$. Using the trivial estimate $M_R \leq kn$ and the assumption $M_R \geq k \geq \frac{24 \ln n}{\epsilon^2}$, it is straightforward to show that $\frac{R_p}{R_c} \geq \frac{k}{4}$. In other words, the bound in (11) is $\frac{k}{4}$-times worse than the randomized bound. This is not very much surprising, since in (11) we divide M_R by k^2.

A major improvement on the inequality (11) would be the proof of an inequality of the form

$$M_{opt} \geq \frac{M_R^2}{ckn}$$

with some positive constant c not depending on k or n, m.

3 Concluding Remarks

(a) In section 2 we assumed $k \geq \frac{24 \ln n}{\epsilon^2}$. We saw in the discussion at the end of section 1 that our type of probabilistic analysis always requires $k = \Omega(\ln n)$. It would be interesting to exhibit the best approximation factor, when k is small, i.e. $k = O(\ln n)$.

(b) Is it true that for k-matching in hypergraphs the inequality

$$M_{opt} \geq \frac{M_R^2}{ckn}$$

holds with a positive constant c independent of k, n, m ?

(c) A challenging problem in the context of derandomization is the problem of finding parallel derandomized algorithms (see Berger and Rompel [7], Motwani, Naor and Naor [17] and Alon [3]). For the hypergraph matching problem such an algorithm is not known.

(d) The probabilistic analysis presented in this paper is based on the fact that the objective function is linear. But many combinatorial optimization problems can be formulated in a direct and natural way as 0-1 quadratic optimization problems. In [22] it is shown that a theory of randomized rounding and derandomization can be developed for the graph partitioning problem. More examples of derandomization in integer programming can be found in [23] and [24].

References

[1] R. Aharoni, P. Erdős, N. Linial; *Dual integer linear programs and the relationship between their optima.* Proceedings of the 17th ACM Symposium on the Theory of Computing, ACM, New York (1985), 476-483.

[2] N. Alon, J. Spencer, P. Erdős; *The prababilistic method.* John Wiley & Sons, Inc. 1992.

[3] N. Alon; *A parallel algorithmic version of the Local Lemma.* Random Structures and Algorithms, Vol.2, No.4 (1991), 367-378.

[4] D. Angluin, L.G. Valiant: *Fast probabilistic algorithms for Hamiltonion circuits and matchings.* J. Computer and System Sciences, Vol. 18, (1979), 155–193.

[5] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy; *On the intractability of approximation problems.* (early draft), UC Berkeley and Stanford University, (1992).

[6] J. Beck; *An algorithmic approach to the Lovász Local Lemma I.* Random Structures and Algorithms, Vol.2, No.3.,(1991), 343-365.

[7] B. Berger, J. Rompel; *Simulating* $(\log^c n)-wise$ *independence in NC.* Proc. IEEE Conference on Foundations of Computer Science 1989, IEEE Computer Society Press, Los Alamitos, CA, 2-8.

[8] P. Erdös, J.L. Selfridge; *On a combinatorial game.* Jour.Comb.Th., Ser.A 14, 298-301, (1973).

[9] V. Faber, L. Lovász; *Problem 18* in Lecture Notes in Mathematics 411, p.284. C. Berge and D.K. Ray-Chaudhri(eds.), Springer Verlag, Berlin/New York (1974).

[10] Z. Füredi; *Maximum degree and fractional matchings in uniform hypergraphs.* Combinatorica 1 (1981), 155 - 162.

[11] Z. Füredi, J. Kahn, P.D. Seymour; *On the fractional matching polytope of a hypergraph.* Combinatorica 13 (2), (1993), 167 - 180.

[12] M. Grötschel, L. Lovász, A. Schrijver; *Geometric algorithms and combinatorial optimixation.* Springer-Verlag (1988)

[13] N.N. Kuzyurin; *On the relationship between the optima of linear and integer programming.* Discrete Math. Appl., Vol.2, No.3, (1992), 305 - 311.

[14] L. Lovász; *On the ratio of optimal and fractional covers.* Discrete Mathematics, 13 (1975), 383-390.

[15] C. McDiarmid; *On the method of bounded differences.* Surveys in Combinatorics, 1989. J. Siemons, Ed.: London Math. Soc. Lectures Notes, Series 141, Cambridge University Press, Cambridge, England 1989.

[16] K. Mehlhorn; *Data structures and algorithms 1: Sorting and Searching.* Springer-Verlag (1984)

[17] R. Motwani, J. Naor, M. Naor; *The probabilistic method yields deterministic parallel algorithms.* Proc. of the 30th IEEE conference on Foundation of Computer Science (1989), 8–13.

[18] C. H. Papadimitriou, K. Steiglitz; *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, Englewood Cliffs NJ, (1982).

[19] P. Raghavan, C. D. Thompson; *Randomized Rounding: A technique for provably good algorithms and algorithmic proofs.* Combinatorica 7 (4), (1987), 365-373.

[20] P. Raghavan; *Probabilistic construction of deterministic algorithms: Approximating packing integer programs.* Jour. of Computer and System Sciences 37, (1988), 130-143.

[21] J. Spencer; *Ten lectures on the probabilistic method.* SIAM, Philadelphia (1987).

[22] A. Srivastav, P. Stangier; *The relationship between fractional and integral graph partitioning.* Preprint, Research Institute of Discrete Mathematics, University of Bonn, (1993).

[23] A. Srivastav, P. Stangier; *Algorithmic Chernoff-Hoeffding inequalties in integer programming.* Preprint, Institut für Informatik, Universität zu Köln, (1993), Submitted.

[24] A. Srivastav, P. Stangier; *Integer multicommodity flows with reduced demands.* in T. Lengauer (eds.), Lecture Notes in Computer Science, 1993, Springer-Verlag, pp. 360 - 372. (Proc. First Annual European Symposium on Algorithms 1993 (ESA'93), Bonn.)