DISCRETE APPLIED MATHEMATICS

ELSEVIER Discrete Applied Mathematics 57 (1995) 255–269

Weighted fractional and integral k-matching in hypergraphs

Anand Srivastav*,[a], Peter Stangier[b]

[a]*Forschungsinstitut für Diskrete Mathematik, Universität Bonn, Nassestr. 2, 53113 Bonn, Germany*
[b]*Institut für Informatik, Universität zu Köln, Pohligstr. 1, 50969 Köln, Germany*

Received 14 September 1992; revised 25 February 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 et al. (1985) and Alon et al. (1992).

Keywords: Hypergraph matching; Integer and linear programming; Randomized algorithm; Derandomization

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 $w_i \geqslant 0$ be rational weights of the hyperedges, $i = 1, \ldots, m$. 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

*Corresponding author.

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 can a (deterministic) polynomial time approximation algorithm 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 $S_{\mathrm{opt}} \leqslant (1 + \log n) S_{\mathrm{R}}$. Recently this inequality was generalized by Kuzyurin [13] to general integer programming (minimization problem) with non-negative integer data. Kuzyurin's result implies in particular for weighted k-set covering with non-negative integer weights the inequality $S_{\mathrm{opt}} \leqslant (1 + \log kn) S_{\mathrm{R}}$. In the unweighted case of 1-set covering Aharoni et al. [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, $M_{\mathrm{opt}}/M_{\mathrm{R}} \geqslant M_{\mathrm{R}}/n$. Recently Füredi et al. [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 $\mathscr{M}$ obeying the stronger inequality

$$M_{\mathrm{R}} \leqslant \sum_{e \in \mathscr{M}} \left(|e| - 1 + \frac{1}{|e|} \right) w(e).$$

A different direction of research was undertaken by Raghavan and Thompson [20] who gave in the unweighted case, assuming that $k \geqslant 6 \ln n$, a probabilistic approximation algorithm finding a k-matching M such that $M \geqslant (1 - \delta) M_{\mathrm{R}} - \mathrm{O}(\sqrt{(1 - \delta) M_{\mathrm{R}}})$, 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 [19].

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, in view of the results of Arora et al. [5] we may raise the conjecture that for arbitrary instances of weighted k-matching even a polynomial-time approximation scheme is 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 the following theorem.

Theorem (Corollaries 2.6–2.8). (i) *Let* $0 < \varepsilon < 1$ *and let the edge weights be rational numbers in* $[0, 1]$. *Suppose that* $k \geqslant 24 \ln n / \varepsilon^2$ *and* k *edges have total weight at least* $18 \varepsilon^{-2}$.

Then with a derandomized algorithm we can find in polynomial time a k-matching M such that $M \geqslant (1-\varepsilon)M_{\mathrm{opt}}$.

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

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

The essential methods we use are randomized rounding and derandomization. While randomized rounding can be performed as in [20], 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 circumvent this problem constructing a new class of pessimistic estimators for the conditional probabilities under consideration which can be derived from McDiarmid's [15] proof of the Angluin–Valiant inequality bounding 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 for 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 Spencer [21] 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 with 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 $\mathrm{O}(nm^2 \log(mn/\varepsilon))$-time derandomized rounding algorithm. In this paper we show that at least for the weighted hypergraph matching problem derandomized rounding only needs $\mathrm{O}(m^2 \log m + mn)$-time, so we have a strongly polynomial-time rounding algorithm. At this moment we do not have an 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 [20] and consists of essentially two steps: randomized rounding and scaling down the probability of setting the hyperedge assignment variable x_i to 1 by a factor of $1 - \varepsilon/2$ for all $i = 1, \dots, m$.

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

$$\max \left\{ \sum_{i=1}^{m} w_i x_i;\ Ax \leqslant k, x \in [0,1]^m \right\}$$

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

(2) (Scaling) Choose $\varepsilon \in [0,1]$ and replace $\tilde{x}$ by $(1 - \varepsilon/2)\tilde{x}$.
(3) (Randomized rounding) For $i = 1, \dots, m$ set independently $x_i = 1$ with probability $(1 - \varepsilon/2)\tilde{x}_i$ and $x_i = 0$ with probability $1 - (1 - \varepsilon/2)\tilde{x}_i$
(4) Output the vector $x = (x_1, \dots, x_m) \in \{0,1\}^m$.

Let us denote by M_{opt} the optimal value of the weighted k-matching problem, by 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 may neglect the exact degree of the polynomial bounding L'.

Raghavan and Thompson [20] analysed the algorithm P-HYPERMATCH in the unweighted case ($w \equiv 1$) and showed that for a certain scaling factor $0 < \varepsilon < 1$ the algorithm finds a k-matching M with $M \geqslant (1 - \varepsilon/2)\, M_{\text{opt}} - \mathrm{O}(\sqrt{(1 - \varepsilon/2) M_{\mathrm{R}}})$, provided that $k \geqslant 6 \ln n$. It is easy to check that their proof is also valid in the weighted case, so we obtain the same result. But given an $\varepsilon \in (0,1)$ we are interested in an approximation $M \geqslant (1 - \varepsilon) M_{\text{opt}}$, 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 $24 \ln n/\varepsilon^2$.

Lemma 1.1 (Angluin and Valiant [4]; for a proof see [15]). *Let $\psi_1, \ldots, \psi_m$ be independent random variables, with $0 \leqslant \psi_j \leqslant 1$ and $E(\psi_j) = p_j$ for all $j = 1, \ldots, m$. Let $\psi = (1/m)\sum_{j=1}^{m} \psi_j$, $p = (1/m)\sum_{j=1}^{m} p_j$ and $0 < \beta \leqslant 1$. Then:*
(i) $\mathrm{Prob}(\psi > (1+\beta)mp) \leqslant \exp(-\beta^2 mp/3)$.
(ii) $\mathrm{Prob}(\psi < (1-\beta)mp) \leqslant \exp(-\beta^2 mp/2)$.

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

Proof. Let $n \geqslant 8$ (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) $\mathrm{Prob}(\sum_{j=1}^{m} a_{ij}x_j > k) \leqslant 1/8n$ for all $i = 1, \ldots, n$.
(b) $\mathrm{Prob}(M(x) < (1-\varepsilon/2)M_{\mathrm{opt}} - 2\sqrt{(1-\varepsilon/2)M_{\mathrm{R}}}) \leqslant \mathrm{e}^{-2}$.
With $n \geqslant 8$ and the assumption on k we have

$$\ln 8n \leqslant \frac{k\varepsilon^2}{12(1-\varepsilon/2)}. \tag{1}$$

Choose $\beta := \varepsilon/(2-\varepsilon)$. Trivially $0 < \beta < 1$ and the Angluin–Valiant inequality (Lemma 1.1(i)) and (1) prove (a):

$$\mathrm{Prob}\left(\sum_{j=1}^{m} a_{ij}x_j > k\right) = \mathrm{Prob}\left(\sum_{j=1}^{m} a_{ij}x_j > (1+\beta)\left(1-\frac{\varepsilon}{2}\right)k\right) \leqslant \frac{1}{8n}.$$

With $\beta_0 := \sqrt{4/(1-\varepsilon/2)M_{\mathrm{R}}}$ we may assume that $0 < \beta_0 \leqslant 1$, because if $\beta_0 > 1$, we have the zero probability event "$M < 0$". Finally $M_{\mathrm{R}} \geqslant 18/\varepsilon^2$ implies $(1-\varepsilon/2)M_{\mathrm{R}} - \sqrt{4(1-\varepsilon/2)M_{\mathrm{R}}} \geqslant (1-\varepsilon)M_{\mathrm{R}}$. □

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

Proof. We may assume that $n \geqslant 4$. This implies $18/\varepsilon^2 \leqslant 24 \ln n/\varepsilon^2 \leqslant k$. But in the unweighted case we always have $M_{\mathrm{R}} \geqslant M_{\mathrm{opt}} \geqslant k$, hence Theorem 1.2 proves Corollary 1.3. □

Remark. We assumed that k is at least $24 \ln n/\varepsilon^2$ which differs by the $4\varepsilon^{-2}$ 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 $M_{\mathrm{R}} \geqslant 18/\varepsilon^2$. 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 $M \geqslant (1 - \varepsilon/2)M_{\mathrm{R}} - 2\sqrt{(1 - \varepsilon/2)M_{\mathrm{R}}}$. 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 $M_{\mathrm{R}} > 4$. Let us proceed to the main problem, the derandomization of the results above.

2. Derandomized algorithms

We briefly sketch the derandomization idea of Spencer [21]. Let $\varepsilon \in (0, 1)$ and let $k \geqslant 24 \ln n/\varepsilon^2$. Denote by E_ε^c the event

"$\exists i (Ax)_i > k$ or $M(x) < (1 - \varepsilon)M_{\mathrm{opt}}$".

If $M_{\mathrm{R}} \geqslant 18/\varepsilon^2$ Theorem 1.2 gives $\mathrm{Prob}(E_\varepsilon^c) \leqslant 0.27$. The method of conditional probabilities seeks 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 $z \to \mathrm{Prob}(E_\varepsilon^c \mid z)$,
    $z \in \{0, 1\}$.
(b) Induction step ($l + 1$). If $x_1, \dots, x_l$ have been selected, set $x_{l+1} = z_{l+1}$, where
    $z_{l+1}$ minimizes the function $z \to \mathrm{Prob}(E_\varepsilon^c \mid x_1, \dots, x_l, z)$, $z \in \{0, 1\}$.
(c) Stop, when $l = m$.
```

The striking observation is that the output vector x satisfies the event E_ε, because the inequalities $1 > \mathrm{Prob}(E_\varepsilon^c) \geqslant \mathrm{Prob}(E_\varepsilon^c \mid x_1) \geqslant \cdots \geqslant \mathrm{Prob}(E_\varepsilon^c \mid x_1, \dots, x_m)$ and the fact that $\mathrm{Prob}(E_\varepsilon^c \mid x_1, \dots, x_m)$ is either zero or one imply $\mathrm{Prob}(E_\varepsilon^c \mid x_1, \dots, x_m) = 0$. 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 $\mathrm{Prob}(E_\varepsilon^c \mid x_1, \dots, x_l)$. 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 $U(E_\varepsilon^c)$ *denote a family 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) $\mathrm{Prob}(E_\varepsilon^c | x_1, \ldots, x_l) \leqslant U_l(x_1, \ldots, x_l)$ for all $x_1, \ldots, x_l \in \{0, 1\}$, $l \leqslant m$.

(ii) $U_{l+1}(x_1, \ldots, x_l, x_{l+1}) \leqslant U_l(x_1, \ldots, x_l)$ 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 $\log 1/\varepsilon$ for each l.

Then $U(E_\varepsilon^c)$ 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 [19] constructed for *unweighted* k-matching a family of functions which satisfies 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, $s \geqslant 1$.

(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 is 0 or 1 and s^{w_j} is computable iff s is computable. In [19] (see Theorem 7) s was defined as $s := D((1 - \varepsilon/2)M_{\mathrm{R}}, 1/n) - 1$, where $D((1 - \varepsilon/2)M_{\mathrm{R}}, 1/n)$ is a positive root of the equation

$$z - (1 - z)\ln(1 + z) + \frac{\ln n}{(1 - \varepsilon/2)M_{\mathrm{R}}} = 0.$$

Unfortunately, it is not known how to find a root of such an 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 $\mathrm{O}(m^2 \log m)$-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 Corollaries 5.1 and 5.2 in [15].

Lemma 2.2. *Let* $a_1, \ldots, a_m$ *be real numbers with* $0 \leqslant a_j \leqslant 1$ *for each* j *and let* $\psi_1, \ldots, \psi_m$ *be independent 0–1 valued random variables. Let* $\tilde{p}_j = E(\psi_j)$, $\tilde{q}_j = 1 - \tilde{p}_j$, $\psi = \sum_{j=1}^m a_j \psi_j$, $p = (1/m)E(\psi)$, $q = 1 - p$ *and* $0 < \beta \leqslant 1$. *Define*

$s^+ = q(1+\beta)/(q-p\beta)$, $s^- = (q+p\beta)/q(1-\beta)$ and for $1 \leqslant l \leqslant m$ let $x_1, \dots, x_l \in \{0,1\}$. Then we have

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

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

Lemma 2.2 motivates the definition of the basic functions for the construction of the pessimistic estimator $U(E_\varepsilon^c)$. In the following let $y_1, \dots, y_m$ denote the scaled variables $y_j := (1-\varepsilon/2)\tilde{x}_j$. (Recall that $(\tilde{x}_j)$ is 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 $[0, \sqrt{8/(2-\varepsilon)M_\mathrm{R}}]$ we can find a rational number β_0 with $0 \leqslant \beta_0 - \sqrt{8/(2-\varepsilon)M_\mathrm{R}} \leqslant 2/m$ in $\mathrm{O}(\log m)$ steps. Since $M_\mathrm{R} \leqslant m \leqslant m^2$, we get

$$\sqrt{\frac{8}{(2-\varepsilon)M_\mathrm{R}}} \leqslant \beta_0 \leqslant \sqrt{\frac{9}{(2-\varepsilon)M_\mathrm{R}}}. \tag{2}$$

(a) *The event "$M < (1-\beta_0)(1-\varepsilon/2)M_\mathrm{R}$"*: Let β_0 be as in (2), $p_0 = (1-\varepsilon/2)M_\mathrm{R}/m$, $q_0 = 1 - p_0$ and $s_0 = (q_0 + p_0\beta_0)/q_0(1-\beta_0)$. For $l = 0$ define

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

For $l \geqslant 1$ and $z_1, \dots, z_l \in \{0,1\}$ define

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

(b) *The event "$\exists i\ (Ax)_i > k$"*: Let $1 \leqslant i \leqslant n$ be arbitrary. Let $p = (1-\varepsilon/2)k/m$, $q = 1 - p$, $\beta = \varepsilon/(2-\varepsilon)$ and $s = q(1+\beta)/(q-p\beta)$. For $l = 0$ define

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

For $l \geqslant 1$ and $z_1, \dots, z_l \in \{0,1\}$ define

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

(c) *The event* "$M < (1-\beta_0)(1-\varepsilon/2)M_{\mathrm{R}}$ *or* $\exists i\,(Ax)_i > k$": Let us denote by $E_\varepsilon^{\mathrm{c}}$ the event defined in (c). Let for $l \geqslant 1$

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

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 conditions (i)–(iii) of Definition 2.1.

Lemma 2.3. *We have for each integer l with $1 \leqslant l \leqslant m$:*

(i) $\operatorname{Prob}(E_\varepsilon^{\mathrm{c}} \mid x_1,\dots,x_l) \leqslant V_l(x_1,\dots,x_l)$ *for all* $x_1,\dots,x_l \in \{0,1\}$.

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

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

Proof. (i) The inequality $\operatorname{Prob}(E_\varepsilon^{\mathrm{c}} \mid x_1,\dots,x_l) \leqslant V_l(x_1,\dots,x_l)$ follows from Lemma 2.2.

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

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

and for $i \geqslant 1$

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

Let

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

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

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

(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 $\sqrt{8/(2-\varepsilon)M_{\mathrm{R}}} \leqslant \beta_0$ (see (2)) it follows that

$$V_0^{(0)} \leqslant \exp(-\tfrac{1}{2}\beta_0^2 p_0 m) \leqslant \mathrm{e}^{-2}.$$

For $i \geqslant 1$ we observe

$$y_j e^{a_{ij} \ln s} + 1 - y_j = 1 + y_j(e^{a_{ij} \ln s} - 1)$$
$$\leqslant 1 + y_j a_{ij}(e^{\ln s} - 1).$$

Since $\sum_{j=1}^{m} a_{ij} y_j \leqslant (1 - \varepsilon/2)k = mp$ we have

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

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

$$V_1(x_1) = \sum_{i=0}^{n} V_0^{(i)} \leqslant \tfrac{1}{8} + e^{-2} \leqslant 0.27. \qquad \square$$

Let us consider the lth step ($l \geqslant 1$) 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 $i \geqslant 1$, 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 $s = q(1 + \beta)/(q - p\beta)$ 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, $B \geqslant 1$ and $0 < \gamma \leqslant 1$. Let $0 < \varepsilon \leqslant 1$ and suppose that $\sum_{i=1}^{m} |A_i| \leqslant \alpha_1 m$ and $b \leqslant \alpha_2 m/\varepsilon$ for some non-negative integer constants α_1, α_2. Let $N = 10\lceil \alpha_1 m \rceil \lceil \log \lceil \alpha_2 m \rceil \rceil + m + \lceil \log \lceil (m+1)/\gamma \rceil \rceil$ and let T_N be the Nth degree Taylor polynomial of the exponential function.*

(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 $O(m \log(m/\gamma\varepsilon))$-time such that the inequality*

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

holds uniformly for all A_i as above.

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

We are ready to prove the main results.

Theorem 2.5. *Let $\varepsilon \in (0,1)$, $k \geqslant 24 \ln n/\varepsilon^2$ and $M_{\mathrm{R}} \geqslant 18/\varepsilon^2$. Given an optimal LP-solution $\tilde{x} \in [0,1]^m$, derandomization finds a k-matching M such that $M \geqslant (1-\varepsilon)M_{\mathrm{R}}$ in $\mathrm{O}(m^2 \log m + mn)$-time.*

Proof. We approximate the function $V_l^{(0)}(x_1,\dots,x_l)$ in order to define the pessimistic estimator. Given $x_1,\dots,x_l \in \{0,1\}$ set $Z_j = -w_j X_j$ for $j \geqslant l+1$, $Z_0 = (1-\beta_0)p_0 m$ and $Z_j = -w_j x_j$ for $j = 1,\dots,l$. Then

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

We will use Lemma 2.4. Since $M_{\mathrm{R}} \geqslant 18/\varepsilon^2$, we have $\beta_0 \leqslant (m-1)/m$ which implies $s_0 \leqslant 4m/\varepsilon$. Set $\alpha_1 = 2$, $\alpha_2 = 4$ and $\gamma = 1/8(4m-1)$ 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 $\prod_{j=l}^{m} \mathrm{e}^{Z_j \ln s_0}$ can be done uniformly for all $x_1,\dots,x_l$. Hence we have

$$\left\| \prod_{j=0}^{m} \mathrm{e}^{Z_j \ln s_0} - \prod_{j=0}^{m} T_N(Z_j b) \right\|_{\infty} \leqslant \gamma.$$

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

$$\left| V_l^{(0)}(x_1,\dots,x_l) - \prod_{j=0}^{m} \mathbb{E}(T_N(Z_j b)) \right| \leqslant \gamma.$$

Set

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

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

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

We show that this family of functions defines a pessimistic estimator for the event $E_\varepsilon^{\mathrm{c}}$.

Condition (i): With Lemma 2.3(i)

$$\mathrm{Prob}(E_\varepsilon^{\mathrm{c}} | x_1,\dots,x_l) \leqslant \sum_{i=1}^{n} V_l^{(i)}(x_1,\dots,x_l) + V_l^{(0)}(x_1,\dots,x_l).$$

But

$$V_l^{(0)}(x_1,\dots,x_l) \leqslant T_l^{(0)}(x_1,\dots,x_l) + \gamma.$$

Hence $\mathrm{Prob}(E_\varepsilon^{\mathrm{c}} | x_1,\dots,x_l) \leqslant U_l(x_1,\dots,x_l)$ for all $x_1,\dots,x_l$.

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

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

it is straightforward to prove the inequality

$$U_{l+1}(x_1,\dots,x_{l+1}) \leqslant U_l(x_1,\dots,x_l)$$

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

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

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

The computation of the running time goes as follows: Let us first consider the approximation of $V_l^{(0)}(x_1,\dots,x_l)$. Note that

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

By Lemma 2.4 we can compute each $\mathbb{E}(T_N(Z_j b))$ in $O(m\log(m/\gamma\varepsilon))$-time. In the first step ($l = 1$) of the WALK procedure we have to compute $\prod_{j=1}^{m} \mathbb{E}(e^{Z_j \ln s_0})$ for some $Z_1 = x_1 \in \{0,1\}$, and this requires the computation of m polynomials, therefore $O(m^2\log(m/\gamma\varepsilon))$-time. But according to the recursion (10) in the forthcoming steps ($l \geqslant 2$) we have to do only one update computing two polynomials, which can be done in $O(m\log(m/\gamma\varepsilon))$-time per step, hence summing up over all the m steps we need $O(m^2\log(m/\gamma\varepsilon))$-time. Now for each $i \geqslant 1$ the total computation time for $V_l^{(i)}(x_1,\dots,x_l)$ 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 need for the computation of all the nm numbers $V_l^{(i)}(x_1,\dots,x_l)$ a total time of $O(nm)$, hence the total time of the rounding algorithm is $O(m^2\log(m/\gamma\varepsilon) + nm)$. We can assume that $\varepsilon \geqslant 1/m$. Otherwise we would get $k \geqslant 24m^2\ln m$, and the hypergraph k-matching problem would become trivial. This gives us together with $\gamma = 1/8(4m-1)$ the total time of $O(m^2\log m + nm)$. □

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

Proof. $w_1 + \cdots + w_k \geqslant 18/\varepsilon^2$ implies $M_{\mathrm{R}} \geqslant 18/\varepsilon^2$ and Theorem 2.5 gives $M \geqslant (1-\varepsilon)M_{\text{opt}}$. □

In the unweighted case we have the following corollary.

Corollary 2.7. *Let $w \equiv 1$, $\varepsilon \in (0,1)$ and $k \geqslant 24 \ln n/\varepsilon^2$. Then with linear programming and derandomization we can find in polynomial time a k-matching with $M \geqslant (1-\varepsilon)M_{\text{opt}}$.*

Proof. $k \geqslant 24 \ln n/\varepsilon^2$ implies $M_{\mathrm{R}} \geqslant 18/\varepsilon^2$ and Theorem 2.5 proves Corollary 2.7. □

For arbitrary weighted k-matching Theorem 2.5 implies Corollary 2.8.

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

Remark. In [1] Aharoni et al. proved for unweighted 1-matching in hypergraphs the inequality $M_{\text{opt}} \geqslant M_{\mathrm{R}}^2/n$. An examination of their proof shows that a similar inequality holds for k-matching:

$$M_{\text{opt}} \geqslant \frac{M_{\mathrm{R}}^2}{k^2 n}. \tag{11}$$

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 $x \in \mathbb{R}^m$, $x \geqslant 0$,

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

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

$$M_{\text{opt}} \geqslant \left(1 - \frac{\varepsilon}{2}\right) M_{\mathrm{R}} - \sqrt{4\left(1 - \frac{\varepsilon}{2}\right) M_{\mathrm{R}}}. \tag{13}$$

Let $R_{\mathrm{c}} := M_{\mathrm{R}}/nk^2$ and $R_{\mathrm{p}} := (1 - \varepsilon/2) - 2\sqrt{(1-\varepsilon/2)/M_{\mathrm{R}}}$. Using the trivial estimate $M_{\mathrm{R}} \leqslant kn$ and the assumption $M_{\mathrm{R}} \geqslant k \geqslant 24 \ln n/\varepsilon^2$, it is straightforward to show that $R_{\mathrm{p}}/R_{\mathrm{c}} \geqslant k/4$. In other words, the bound in (11) is $(k/4)$-times worse than the randomized bound. This is not very 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

$$M_{\text{opt}} \geqslant \frac{M_{\text{R}}^2}{ckn}$$

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

3. Concluding remarks

(a) In Section 2 we assumed $k \geqslant 24 \ln n/\varepsilon^2$. 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. $k = O(\ln n)$ and to show that even under the condition $k = \Omega(\ln n)$ the k-matching problem is NP-hard or MAX-SNP-hard.

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

$$M_{\text{opt}} \geqslant \frac{M_{\text{R}}^2}{ckn}$$

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 [7, 17, 2]). 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, 24].

References

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

[2] N. Alon, A parallel algorithmic version of the local lemma, Random Structures Algorithms 2 (1991) 367–378.

[3] N. Alon, J. Spencer and P. Erdös, The Probabilistic Method (Wiley, New York, 1992).

[4] D. Angluin and L.G. Valiant, Fast probabilistic algorithms for Hamiltonion circuits and matchings, J. Comput. System Sci. 18 (1979) 155–193.

[5] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, Proof verification and the intractability of approximation problems, in: Proceedings of the 32th IEEE Symposium on the Foundation of Computer Science (1992) 14–23.

[6] J. Beck, An algorithmic approach to the Lovász local lemma I, Random Structures Algorithms 2 (1991) 343–365.

[7] B. Berger and J. Rompel, Simulating ($\log^e n$)-wise independence in NC, in: Proceedings of the IEEE Conference on Foundations of Computer Science (IEEE Computer Society Press, Los Alamitos, CA, 1989) 2–8.
[8] P. Erdös and J.L. Selfridge, On a combinatorial game, J. Combin. Theory Ser. A 14 (1973) 298–301.
[9] V. Faber and L. Lovász, Problem 18, in: C. Berge and D.K. Ray-Chaudhri, eds., Lecture Notes in Mathematics 411 (Springer, Berlin, 1974) 284.
[10] Z. Füredi, Maximum degree and fractional matchings in uniform hypergraphs, Combinatorica 1 (1981) 155–162.
[11] Z. Füredi, J. Kahn and P. D. Seymour, On the fractional matching polytope of a hypergraph, Combinatorica 13 (1993) 167–180.
[12] M. Grötschel, L. Lovász and A. Schrijver, Geometric Algorithms and Combinatorial Optimization (Springer, Berlin, 1988).
[13] N.N. Kuzyurin, On the relationship between the optima of linear and integer programming, Discrete Appl. Math. (1992) 305–311.
[14] L. Lovász, On the ratio of optimal and fractional covers, Discrete Math. 13 (1975) 383–390.
[15] C. McDiarmid, On the method of bounded differenes, in: J. Siemons, ed., Surveys in Combinatorics, London Mathematical Society Lecture Note Series 141 (Cambridge University Press, Cambridge, 1989).
[16] K. Mehlhorn, Data Structures and Algorithms 1: Sorting and Searching (Springer, Berlin, 1984).
[17] R. Motwani, J. Naor and M. Naor, The probabilistic method yields deterministic parallel algorithms, in: Proceedings of the 30th IEEE Conference on Foundation of Computer Science (1989) 8–13.
[18] C.H. Papadimitriou and K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity (Prentice-Hall, Englewood Cliffs, NJ, 1982).
[19] .P. Raghavan, Probabilistic construction of deterministic algorithms: approximating packing integer programs, J. Comput. System Sci. 37 (1988) 130–143.
[20] P. Raghavan and C.D. Thompson, Randomized rounding: a technique for provably good algorithms and algorithmic proofs, Combinatorica 7 (1987) 365–373.
[21] J. Spencer; Ten Lectures on the Probabilistic Method (SIAM, Philadelphia, PA, 1987).
[22] A. Srivastav and P. Stangier, The relationship between fractional and integral graph partitioning, Preprint, Institut für Informatik, Freie Universität Berlin (1994).
[23] A. Srivastav and P. Stangier, Algorithmic Chernoff–Hoeffding inequalities in integer programming, Preprint, Institut für Informatik, Freie Universität Berlin (1994). Extended abstract in: D.-Z. Du and X.-S. Zhang, eds., Proceedings of the 5th International Symposium on Algorithms and Computation 1994 (ISAAC '94), Beijing, Lecture Notes in Computer Science (Springer, Berlin, 1994) 226–233.
[24] A. Srivastav and P. Stangier, Integer multicommodity flows with reduced demands in: T. Lengauer, ed., Proceedings of the 1st Annual European Symposium on Algorithms 1993 (ESA'93), Bonn, Lecture Notes in Computer Science 1993 (Springer, Berlin, 1993) 360–372.