# COMPUTATIONAL COMPLEXITY OF THE PERFECT MATCHING PROBLEM IN HYPERGRAPHS WITH SUBCRITICAL DENSITY


MAREK KARPIŃSKI*

*Department of Computer Science, University of Bonn*
*Römerstrasse 164, 53117 Bonn, Germany*
*marek@cs.uni-bonn.de*

ANDRZEJ RUCIŃSKI†

*Faculty of Mathematics and Computer Science, Adam Mickiewicz University,*
*Umultowska 87, 61-614 Poznań, Poland*
*rucinski@amu.edu.pl*

EDYTA SZYMAŃSKA‡

*Faculty of Mathematics and Computer Science, Adam Mickiewicz University,*
*Umultowska 87, 61-614 Poznań, Poland*
*edka@amu.edu.pl*



In this paper we consider the computational complexity of deciding the existence of a perfect matching in certain classes of dense $k$-uniform hypergraphs. It has been known that the perfect matching problem for the classes of hypergraphs $H$ with *minimum* $((k-1)-$wise) *vertex degree* $\delta(H)$ at least $c|V(H)|$ is NP-complete for $c < \frac{1}{k}$ and trivial for $c \geq \frac{1}{2}$, leaving the status of the problem with $c$ in the interval $[\frac{1}{k}, \frac{1}{2})$ widely open. In this paper we show, somehow surprisingly, that $\frac{1}{2}$ is not the *threshold* for tractability of the perfect matching problem, and prove the existence of an $\epsilon > 0$ such that the perfect matching problem for the class of hypergraphs $H$ with $\delta(H) \geq (\frac{1}{2} - \epsilon)|V(H)|$ is solvable in *polynomial time*. This seems to be the first polynomial time algorithm for the perfect matching problem on hypergraphs for which the existence problem is nontrivial. In addition, we consider parallel complexity of the problem, which could be also of independent interest.



*Keywords*: Hypergraph; perfect matching; complexity.

1991 Mathematics Subject Classification: 05C65, 68Q25, 68Q22

*Research supported by DFG grants and the Hausdorff Center grant EXC59-1.
†Research supported by grant N201 036 32/2546.
‡Research supported by grant N206 017 32/2452.

## 1. Introduction

A *hypergraph* $H = (V, E)$ is a finite set of vertices $V := V(H)$ together with a family $E := E(H)$ of distinct, nonempty subsets of $V$, called edges. In this paper we consider $k$-*uniform hypergraphs* (or, shortly, $k$-*graphs*) in which, for a fixed $k \geq 2$, each edge is of size $k$.

*A matching* in a hypergraph $H$ is a set $M \subseteq E$ of disjoint edges. We often treat $M$ as a subhypergraph of $H$ and identify $M$ with $E(M)$. The number $|M|$ of edges in a matching $M$ is called *the size* of the matching, while the number of vertices missing from $M$, that is, the number $|V(H)| - |V(M)|$ is called *the deficiency* of $M$ in $H$. Note that the deficiency of any matching in $H$ equals $n$ modulo $k$. In other words, if $n \equiv q \mod k$, then $r$-deficient matchings are possible only if $r = q + \ell k$ for some $\ell \geq 0$, and such matchings have, of course, size $\lfloor n/k \rfloor - \ell$. A matching is *perfect* if its deficiency is 0, or equivalently if its size is $\frac{1}{k}|V(H)|$. Therefore, a necessary condition for the existence of a perfect matching in $H$ is that $|V(H)| \equiv 0 \mod k$.

For a $k$-graph $H$ and a set of $k-1$ vertices $S$, let $N_H(S)$ be the set of edges of $H$ containing $S$ and put $deg_H(S) = |N_H(S)|$. We define $\delta(H) = \min_S deg_H(S)$ over all $S \in \binom{V}{k-1}$ and refer to it as the $(k-1)$-*wise, collective minimum degree of* $H$, or simply, *minimum co-degree*, as we will not consider any other kinds of degrees in hypergraphs.

Furthermore, for all integers $k \geq 2$, $r \geq 0$, and $n \geq k$, denote by $t(k, n, r)$ the smallest integer $t$ such that every $k$-graph $H$ on $n$ vertices and with $\delta(H) \geq t$ contains an $r$-deficient matching.

### 1.1. *Three classes of computational problems*

In this paper we will consider three computational problems which are defined below.

For $k \geq 2$, by $\mathbf{PM}(k)$ we denote the problem of deciding whether a $k$-graph $H$ contains a perfect matching. The problem $\mathbf{PM}(2)$ is the classical problem of deciding the existence of a perfect matching in a graph, and is known to be in the polynomial class $\mathcal{P}$ since the paper by Edmonds [3]. For all $k \geq 2$, $\mathbf{PM}(k)$ is equivalent to a decision problem called *exact cover by k-sets*, which is known to be NP-complete for $k \geq 3$, [4].

Having defined the notion of matching deficiency we can formulate a more general problem.

Given integers $k \geq 3$ and $r \geq 0$, let $\mathbf{PM}(k, r)$ denote the problem of deciding whether a $k-$graph $H$ with $|V(H)| \equiv r \mod k$ contains an $r$-deficient matching. In particular, when $0 < r < k$, $\mathbf{PM}(k, r)$ asks for a matching in $H$ which, although non-perfect, is as perfect as one can get. Note also that $\mathbf{PM}(k, 0)$=$\mathbf{PM}(k)$.

Finally, we define a special, "density sensitive", case of the above problem. Given integers $k \geq 3$, $r \geq 0$ and a real $c > 0$, by $\mathbf{PM}(k, r, c)$ we denote the same problem as $\mathbf{PM}(k, r)$ but restricted to $k-$graphs $H$ with minimum co-degree $\delta(H) \geq c|V(H)|$.

When $r = 0$, $\mathbf{PM}(k, 0, c)$ can be viewed as the perfect matching problem for dense $k$-graphs.

### 1.2. *Background and motivation*

In recent years hypergraphs gained a lot of interest as a natural generalization of graphs. Many of the standard graph problems, however, became much more complicated after translating them to hypergraphs. It is indeed the case of the perfect matching problem considered in this paper. Hall's theorem gives a necessary and sufficient condition for the existence of a perfect matching in bipartite graphs. The result has been extended to bipartite hypergraphs by Haxell [6], but turned out to be rather computationally noneffective. Recently, Asadpour, Feige and Saberi [1] reduced a max-min allocation problem, known as *the Santa Claus Problem*, to finding a perfect matching in a class of bipartite hypergraphs but could not solve their problem efficiently.

From the computational point of view, more satisfactory is another, Dirac-type sufficient condition given by Rödl et al. [11]. Recall that the celebrated Dirac theorem for graphs guarantees a Hamilton cycle in every $n$-vertex graph with minimum degree at least $\frac{1}{2}n$, and thus, a perfect matching when $n$ is even. In fact, it is very easy to show that $t(2, n, 0) = \frac{1}{2}n$.

In [11], the authors determined exactly the value of $t(k, n, 0)$ for all integers $k \geq 3$ and sufficiently large $n$. They proved there that $t(k, n, 0) = \frac{1}{2}n - k + c_{k,n}$, where $c_{k,n}$ is an explicit constant depending on the parities of $k$, $n$ and $\frac{n}{k}$, and satisfying $\frac{3}{2} \leq c_{k,n} \leq 3$. Hence, in particular, $t(k, n, 0) \leq \frac{1}{2}n - k + 3 \leq \frac{1}{2}n$. In [12] only a slightly weaker upper bound, $t(k, n, 0) \leq \frac{1}{2}n + \frac{1}{4}k$, but with a simpler proof, was shown.

For deficient matchings, i.e. the case of $r > 0$, a striking difference between perfect and almost perfect matchings was observed in [11]. It was shown there that for $n \equiv r \mod k$ and $k \geq 3$, $t(k, n, r) = \frac{n-r}{k}$ for $r \geq (k-2)k$, and $\frac{n-r}{k} \leq t(k, n, r) \leq \frac{n}{k} + O(\log n)$ for $0 < r < (k-2)k$. Thus, in all cases other than the perfect one, the threshold value of $\delta(H)$ for the existence of an $r$-deficient matching in $H$ is around $\frac{n}{k}$, while in the perfect case it is around $\frac{1}{2}n$.

An immediate consequence of the results in [11] is that the decision problem $\mathbf{PM}(k, 0, c)$ is trivial for every $c \geq \frac{1}{2}$, while $\mathbf{PM}(k, r, c)$, $r > 0$, is trivial already for $c > \frac{1}{k}$. (By trivial we mean that the answer is YES for every instance.)

Szymańska showed in [13] by a polynomial reduction of $\mathbf{PM}(k)$ to $\mathbf{PM}(k, r, c)$ that for all $k \geq 3$, $r \geq 0$, and every constant $c < \frac{1}{k}$, $\mathbf{PM}(k, r, c)$ is NP-complete. It follows that $\mathbf{PM}(k, r)$ is NP-complete too, although this can be derived by a direct reduction from $\mathbf{PM}(k)$.

On the positive side, it was observed in [13] that the argument presented in [11] can be transformed into a constructive one and a polynomial time algorithm for the corresponding search problem when $c > \frac{1}{k}$ and $r > 0$ was provided. In [14] the existential proof from [11] was turned into a polynomial time algorithm finding a perfect matching when $c \geq \frac{1}{2}$.

Those results have established a "phase transition" at $c = \frac{1}{k}$ for $\mathbf{PM}(k, r, c)$, $r > 0$, but left a "hardness gap" of $[\frac{1}{k}, \frac{1}{2})$ for $\mathbf{PM}(k, 0, c)$.

**Problem 1.** *What is the computational complexity of* $\boldsymbol{PM}(k, 0, c)$ *when* $c \in [\frac{1}{k}, \frac{1}{2})$*?*

By the counterexamples introduced in [11] it is apparent that there exist $k$-graphs of minimum co-degree below $\frac{1}{2}|V(H)|$ without a perfect matching, so both answers, YES and NO are possible. This motivated us to investigate the complexity of the existence problem for hypergraphs in the gap interval. Interestingly, it turned out that at least in the upper end of this interval the problem is polynomial. Indeed, in this paper we provide a polynomial time algorithm which for every hypergraph with minimum co-degree at least $(\frac{1}{2} - \epsilon)|V(H)|$ constructs a perfect matching if one exists, and otherwise it exhibits a certificate for non-existence (cf. Theorem 3 and Algorithm PerfectMatch).

Our second result concerns parallelization of the problem. Parallel algorithms have experienced a lot of attention in the late eighties and nineties of the last century. Many interesting algorithms were given, including several deterministic $NC$ algorithms for the maximal independent set as well as randomized algorithms placing the maximum matching problem in the class $RNC$. At the same time some questions are still open. In particular, very small progress has been made in deterministic parallelization of such natural graph problems, like the perfect matching problem.

While the perfect matching problem in graphs can be decided and computed in polynomial time, the parallel complexity of the decision problem remains unknown. Apart from randomized results, only some special classes of graphs have efficient parallel algorithms. This includes dense graphs, in particular Dirac's graphs, that is, graphs $G$ with minimum degree $\delta \geq \frac{1}{2}|V(G)|$. Dalhaus, Hajnal and Karpiński gave in [2] an $NC^2$ parallel algorithm finding a perfect matching in such graphs and showed that for the minimum degree at least $c|V(G)|$, $c < \frac{1}{2}$, the problem is as hard as for all graphs. Recently, Särközy [15] proved that the $\frac{1}{2}$-density barrier breaks down even for the harder problem of Hamiltonian cycle, in a special class of graphs called $\eta-$Chvátal graphs.

Motivated by the results of [2] and [15], we investigate the parallel complexity of the perfect matching problem in dense hypergraphs. Besides being interesting in their own right, we treat parallel algorithms as a tool providing alternative, conceptually easier proofs of existential results. Our Theorem 6 implies that the problem of deciding whether a given $k-$uniform hypergraph $H$, with minimum co-degree at least $c|V(H)|$, $c > \frac{1}{2}$, contains a perfect matching admits an $NC$ algorithm. Along the way, we also design parallel algorithms for constructing *almost* perfect matchings in graphs with restricted minimum co-degrees (cf. Theorems 4 and 5). These algorithms serve as subroutines in the main perfect matching algorithm.

In Section 2 we formally state our results (Theorems 3, 4, 5 and 6, and Proposition 2). After that, in Section 3 the parallel algorithms together with their analysis, which proves Theorems 4, 5 and 6, are presented. The last section is devoted to the proofs of Proposition 2 and Theorem 3. A conference version of this paper appeared as [7].

## 2. Our Results

One goal of this paper is an attempt to understand the complexity of $\mathbf{PM}(k, 0, c)$ in the gap interval $c \in [\frac{1}{k}, \frac{1}{2})$. Theorem 3 below shows that at least in the upper end of the interval the decision problem $\mathbf{PM}(k, 0, c)$ is polynomial in time. Another part of this paper is devoted to an alternative, constructive proof of the bound $t(k, n, 0) \leq \frac{1}{2}n + \frac{1}{4}k$ from [12]. In fact, we turned that proof into a parallel algorithm (see Theorem 6 below), showing that $\mathbf{PM}(k, 0, c)$ is not only in $\mathcal{P}$ but also in the $NC$ class. In the next two subsections we formulate our results precisely.

### 2.1. *Hardness taxonomy*

Concerning the problem $\mathbf{PM}(k, 0, c)$, the results from [11] and [13] described in Section 1.2 have left a hardness gap for $c \in [\frac{1}{k}, \frac{1}{2})$.

We present two results which suggest different answers to Problem 1. To put the first of them into a right context, recall that by [11] we know already that $\mathbf{PM}(k, k, c)$ is trivial for $c > \frac{1}{k}$. In other words, every $k$-graph $H$ with $\delta(H) \geq c|V(H)|$, where $c > \frac{1}{k}$ and $|V(H)|$ is divisible by $k$, has a $k$-deficient matching.

**Proposition 2.** *For every $k \geq 3$, $\boldsymbol{PM}(k)$ is $NP$-complete even when restricted to $k$-graphs containing a $k$-deficient matching.*

It means that knowing that a $k$-graph has a matching just one edge short from a perfect one, does not help in deciding the existence of the latter. This could suggest that $\mathbf{PM}(k, 0, c)$ is NP-complete for all $c \in [\frac{1}{k}, \frac{1}{2})$. However, it turns out that it is not so. Indeed, in Section 4 we describe an algorithm, called PerfectMatch, which, for some $c < \frac{1}{2}$, but sufficiently close to $\frac{1}{2}$, places $\mathbf{PM}(k, 0, c)$ in $\mathcal{P}$.

**Theorem 3.** *For all $k \geq 3$ there exists $\epsilon > 0$ such that if $c \geq \frac{1}{2} - \epsilon$, then $\boldsymbol{PM}(k, 0, c)$ as well as its search version are in $\mathcal{P}$.*

**Remark 1.** *Theorem 3 reveals an interesting feature: it provides a polynomial time algorithm which, unlike the algorithms in [2], [15], [13], or those described in the next section, takes as inputs instances which may not possess a desired matching, and decides whether they indeed have one. If the answer is YES, the algorithm, in fact, computes in polynomial time a perfect matching, while when the answer is NO, it provides an evidence (in a form of a witness partition).*

### 2.2. *Parallel algorithms*

As the model of computation we choose the EREW version of PRAM. Recall that, as shown in [11], the problem $\mathbf{PM}(k, 0, \frac{1}{2})$ is trivial, that is, for all $H$ with $\delta(H) \geq \frac{1}{2}n$, $H$ has a perfect matching. As observed in [13], the existential proof from [11] can be turned into a polynomial time search algorithm of complexity $O(n^{k^2+2k} \log^4 n)$. Here we present a parallel algorithm which places the search version of $\mathbf{PM}(k, 0, c)$, $c > \frac{1}{2}$, in the class $NC$. Recall that $NC = \bigcup_{i \geq 0} NC^i$, and a problem is in $NC^i$ if it admits an algorithm of running time $O(\log^i n)$, using a polynomial number of processors.

Our algorithm, PAR-PERFECTMATCH, is based on the existential proof in [12] and uses as subroutines two other parallel algorithms of independent interest, PAR-LARGEDEFMATCH($r$) and PAR-SMALLDEFMATCH($r$), which find $r$-deficient matchings for, resp., large and small, positive values of $r$, under increasingly restrictive conditions on $\delta(H)$.

The properties of these algorithms are presented in the following theorems. The first of them provides a parallel algorithm which finds an $r$-deficient matching for large $r$, but relatively small $\delta$.

**Theorem 4.** *For every $k \geq 3$ and $r \geq (k-2)k$ there exists a constant $n_0$, and a parallel algorithm, called* PAR-LARGEDEFMATCH($r$), *which in every $k$-graph $H$ on $n \geq n_0$ vertices with $n \equiv r \mod k$ and $\delta(H) \geq \frac{n-r}{k}$ finds an $r$-deficient matching in $O(\log^3 n)$ rounds using a polynomial number of processors. It follows that the search version of $\boldsymbol{PM}(k, r, c)$ is in the class $NC^3$ for $r \geq (k-2)k$ and $c \geq \frac{1}{k}$.*

If the degree condition is strengthened just a little, we can find in parallel a matching of any smaller, but positive, deficiency $r$. The algorithm PAR-SMALLDEFMATCH($r$), given below, uses the algorithm from Theorem 4 as a subroutine.

**Theorem 5.** *For every $k \geq 3$ and $0 < r < (k-2)k$ there exist constants $n_0$ and $C > 0$, and a parallel algorithm, called* PAR-SMALLDEFMATCH($r$), *which in every $k$-graph on $n \geq n_0$ vertices with $n \equiv r \mod k$ and $\delta(H) \geq \frac{n}{k} + C \log n$ finds an $r$-deficient matching in $O(\log^3 n)$ rounds using a polynomial number of processors. It follows that the search version of $\boldsymbol{PM}(k, r, c)$ is in the class $NC^3$ for $0 < r < (k-2)k$ and $c > \frac{1}{k}$.*

Finally, if $\delta(H)$ exceeds $\frac{1}{2}n$, then we are in position to compute in parallel a perfect matching in $H$. This is the main result of this section.

**Theorem 6.** *For every $k \geq 3$ there exists constant $n_0$, and a parallel algorithm, called* PAR-PERFECTMATCH, *which in every $k$-graph on $n \geq n_0$ vertices with $n$ divisible by $k$ and such that $\delta(H) \geq \frac{n}{2} + \frac{k}{4}$ finds a perfect matching in $O(\log^3 n)$ rounds using a polynomial number of processors. It follows that the search version of $\boldsymbol{PM}(k, 0, c)$ is in the class $NC^3$ for $c > \frac{1}{2}$.*

Table 1. The complexity of $\mathbf{PM}(k, r, c)$ with $k \geq 3$. For every *t=trivial* problem there exists an *NC parallel* algorithm finding an $r$-deficient matching.

| $r$ \ $c$ | $c < \frac{1}{k}$ | $\frac{1}{k}$ | $(\frac{1}{k}, \frac{1}{2} - \epsilon)$ | $[\frac{1}{2} - \epsilon, \frac{1}{2}]$ | $c > \frac{1}{2}$ |
|---|---|---|---|---|---|
| $r \geq (k-2)k$ | NP-com | t | t | t | t |
| $0 < r < (k-2)k$ | NP-com | ? | t | t | t |
| $r = 0$ | NP-com | ? | ? | $\mathcal{P}$ | t |

The above three theorems will be proved in the next section. A summary of all computational results about $\mathbf{PM}(k, r, c)$ is displayed in Table 1.

## 3. Description and Analysis of Parallel Algorithms

In this section we prove Theorems 4–6. Each proof consists of a description of the algorithm followed by a proof of its correctness.

### 3.1. *Proof of Theorem 4*

The construction below generalizes the ideas from [2] to hypergraphs. *The intersection graph* of a hypergraph $H$ has the edges of $H$ as its vertices, and two vertices are adjacent if the corresponding edges of $H$ intersect. Observe that the matchings in $H$ map one-to-one with the independent sets of the intersection graph. When we refer to *MIS algorithm*, we always mean the parallel algorithm from [9] which places the maximal independent set problem in $NC^2$.

In short, the idea of the construction is the following. First we compute a maximal matching $M_1$ in $H$ using MIS algorithm in the intersection graph of $H$. This leaves a set $W$ of unmatched vertices in $H$, which is next divided into groups of size $(k-1)k$ each, with a possible remainder of a smaller size. Then, as shown in Fig. 1, an auxiliary bipartite graph $G$ is constructed. The vertices of $G$ correspond to the edges of $M_1$ on one side and the groups of vertices on the other. We put an edge in $G$ connecting a vertex corresponding to an edge $e \in M_1$ with a vertex corresponding to a subset $S$ of $W$ if and only if there exist two disjoint edges in $H$, each of them intersecting $e$ in one vertex and containing $k-1$ vertices from $S$. In the next step a maximal matching $M_2$ in $G$ is computed. Each edge of $M_2$ is then used for absorbing $2(k-1) - (k-2) = k$ vertices into $M_1$ and enlarging its size by one. The whole process is repeated until there are $r$ vertices remaining in $W$.

---

ALGORITHM PAR-LARGEDEFMATCH$(r)$, $r \geq (k-2)k$

---

**In:** $k$-graph $H$ with $n \geq n_0$, $n \equiv r \mod k$, and $\delta(H) \geq \frac{n-r}{k}$
**Out:** $r$-deficient matching $M_1$

(1) Compute **in parallel** a maximal matching $M_1$ in $H$ applying MIS algorithm to the intersection graph of $H$. Let $W := V(H) - V(M_1)$.

(2) Repeat while $|W| > r$

(a) Arbitrarily divide $W$ into $t := \left\lfloor \frac{|W|}{(k-1)k} \right\rfloor$ disjoint sets $S$ of size $|S| = (k-1)k$. Call this family of sets $\mathcal{S}$. Define an auxiliary bipartite graph $G = (V_1, V_2, E(G))$ as follows:

- $V_1 = M_1$ and $V_2 = \mathcal{S}$; thus $|V_2| = t$.
- For each $e \in V_1$ and $S \in V_2$ put **in parallel** an edge $\{e, S\} \in E(G)$ if and only if there are two vertices $u_e, v_e \in e$, $u_e \neq v_e$ and two disjoint $(k-1)-$element subsets $X_S, Y_S$ of $S$ such that $e'_{e,S} := X_S \cup \{u_e\} \in H$ and $e''_{e,S} := Y_S \cup \{v_e\} \in H$.

(b) Compute **in parallel** a maximal matching $M_2$ in $G$ using MIS algorithm.

(c) For every edge $(e, S) \in M_2$ **in parallel** absorb into $M_1$ the set of vertices $X_S \cup Y_S$, by replacing $e$ with $e'_{e,S}$ and $e''_{e,S}$, at the same time releasing from $M_1$ the remaining $k-2$ vertices of $e$, i.e., $M_1 := (M_1 - \{e\}) \cup \{e'_{e,S}, e''_{e,S}\}$. Set $W := V(H) - V(M_1)$.

(3) Return $M_1$.

---

To show that the above algorithm computes a desired matching we need the following fact.

**Fact 7.** *Any maximal matching $M_2$ in the bipartite graph $G$ defined in the algorithm saturates at least*

$$\left\lceil \frac{n - r - k|M_1|}{(k-1)k} \right\rceil$$

*vertices of $V_2$.*

Note that for $r < (k-1)k$ the above quantity is at least as large as $t = |V_2|$, that is, in this case $V(M_2) \supseteq V_2$.

**Proof.** Fix a set $S \in V_2$ and divide it into $k$ disjoint subsets of $k-1$ vertices each, $T_1, \ldots, T_k$. Next, for each $e \in M_1$, let

$$T(e) = \{e' \in E(H) : |e' \cap e| = 1 \text{ and } e' \cap S = T_j \text{ for some } j\}.$$

Further, let $a$ be the number of edges $e \in M_1$ with $|T(e)| \leq k$, and let $b = |M_1| - a$ denote the number of edges $e \in M_1$ with $|T(e)| \geq k+1$. Note that for every $e$ with $|T(e)| \geq k+1$, we have $\{e, S\} \in E(G)$. Indeed, the $k+1$ pairs $(e' \cap e, T_j)$ cannot all involve the same vertex of $e$, or the same set $T_j$, simply because there are only $k$ such vertices and $k$ such sets. Thus, the degree of $S$ in $G$ satisfies $deg_G(S) \geq b$, and it follows trivially that any maximal matching $M_2$ of $G$ contains at least $b$ vertices of $V_2$. It remains to bound $b$ from below.

To prove the required bound on $b$, we estimate the number $\pi_S$ of pairs $(e, e')$, where $e \in M_1$ and $e' \in T(e)$. On the one hand, $\pi_S \leq ka + k^2 b = k|M_1| + (k^2 - k)b$; on

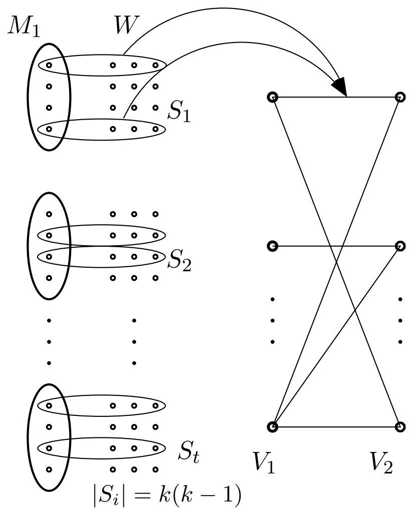


Fig. 1. Constructing the auxiliary bipartite graph ($k = 4$).

the other hand, by the minimum co-degree assumption on $H$, $\pi_S \geq k\delta(H) \geq n - r$. Therefore

$$b \geq \frac{n - r - k|M_1|}{(k-1)k}.$$ □

To see that Algorithm par-LargeDefMatch($r$) finds an $r$-deficient matching, let $u_i$ denote the number of unsaturated by $M_1$ vertices at the start of the $i$th loop of the algorithm in Step (2). Let $b_i$ be the value of $b$ computed in the $i$th loop. Note that the bound on $b$ obtained in the proof of Fact 7 is equivalent to $b_i(k-1)k \geq u_i - r$. Also, since $|M_2| \geq b_i$, at least $b_i k$ more vertices have become saturated after step $i$. Thus, for each $i \geq 1$,

$$u_{i+1} - r = u_i - b_i k - r \leq \tfrac{k-2}{k-1}(b_i - r),$$

and after at most $O(\log n)$ steps the quantity $u_i - r$ will vanish. At this point, $M_1$ becomes an $r$-deficient matching. Hence, the time complexity is $O(\log^3 n)$.

We remark that in the case of graphs discussed in [2], only one iteration in Step (2) was sufficient, saving one logarithmic factor in time complexity.

### 3.2. *Proof of Theorem 5*

Let us begin by noting that without loss of generality we may restrict the range of $r$ to $0 < r \leq k$. Indeed, if $r_1 < r_2$ and $r_i \equiv n \mod k$, $i = 1, 2$, then any $r_1$-deficient matching contains an $r_2$-deficient matching.

The algorithm par-SmallDefMatch presented below uses as subroutine par-LargeDefMatch. In addition, following the absorbing technique introduced in

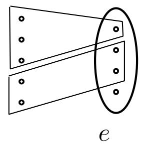

Fig. 2. Absorbing edge.

[11], we will need another parallel subroutine which computes a so called powerful matching. Its success relies on the fact that if $r > 0$ and $n \equiv r \mod k$ then any matching with deficiency larger than $r$ must necessarily leave out at least $k+1$ unsaturated vertices, as opposed to only $k$ when $r = 0$.

We now recall the necessary definitions from [11].

**Definition 8 (absorbing edge)** *Given a set $S$ of $k+1$ vertices, an edge $e \in H$ is called* $S$-absorbing *if there are two disjoint edges $e'$ and $e''$ in $H$ such that $|e' \cap S| = k-1$, $|e' \cap e| = 1$, $|e'' \cap S| = 2$ and $|e'' \cap e| = k-2$. (See Fig. 2.)*

Clearly, if the set $S$ is outside a matching $M$ which contains an $S$-absorbing edge $e$, then $M$ can "absorb" $S$ by swapping $e$ for $e'$ and $e''$ (one vertex of $e$ will become unmatched).

The key feature of the absorbing edge is that if $\delta(H) = \Theta(n)$, then there are $\Theta(n^k)$ of them for every set $S$, $S \subseteq V$, $|S| = k+1$ (see Fact 2.2 in [11]).

For the absorbing technique to work, we need a small matching $M$ (of size $O(\log n)$) containing several absorbing edges for each set $S$. It will be then altered in the absorbing procedure, extending any sufficiently large matching until it becomes $r$-deficient. We call such a matching $M$ *powerful* and define it formally below.

**Definition 9 (powerful matching)** *A matching $M$ in a $k$-graph $H$ is called* powerful *if for every set $S \subset V$ of size $k+1$ the number of $S$-absorbing edges in $M$ is at least $k-2$.*

We need $k-2$ absorbing edges per $S$, because PAR-LARGEDEFMATCH finds a matching with deficiency between $k(k-2)$ and $k(k-1)$. Therefore, in the worst case we might need to absorb $k-3$ other sets before a given set $S_0$, using possibly as many $S_0$-absorbing edges (an edge in a powerful matching is typically $S$-absorbing for many sets $S$).

To construct a small, powerful matching in $H$, we first create an auxiliary graph $G = (X \cup Y, E)$, where $X$ is an independent set. The vertices in $Y$ represent all matchings in $H$ of size $k-2$, while the vertices in $X$ represent all subsets $S$ of vertices of size $k+1$. Let $\mathcal{F}_S$ be the family of all matchings of size $k-2$ consisting of $S$-absorbing edges. The $\{x, y\}$ edges of $G$, where $x \in X$ and $y \in Y$, exhibit the membership of the matchings in the families $\mathcal{F}_S$, while the $\{y', y''\}$ edges, where $y', y'' \in Y$, indicate whether the two matchings represented by $y'$ and $y''$ have a vertex in common. Now, our goal is to construct an independent subset $D$ of $Y$

of size $O(\log n)$ which dominates all vertices of $X$. Then the union of the $(k-2)$-matchings represented by the vertices of $D$ forms the desired powerful matching in $H$. As we will see, this can be done efficiently in parallel if for some $d > 0$

$$deg_G(x) \geq d|Y| \text{ for all } x \in X \text{ and } \Delta(G[Y]) = o\left(\frac{1}{\log n}|Y|\right). \tag{1}$$

---

Algorithm par-IndDomSet

---

**In:** graph $G = (X \cup Y, E)$, $G[X] = \emptyset$, satisfying (1)
**Out:** independent subset $D \subseteq Y$ dominating $X$, $|D| = O(\log n)$

(1) Repeat until $X = \emptyset$:
   (a) For all $y \in Y$ compute **in parallel** $deg_G(y, X)$; set $y_0$ for the lexicographically first $y$ for which $deg_G(y, X) \geq \frac{d}{2}|X|$;
   (b) Set $D := D \cup \{y_0\}$; $X := X - \{x : \{x, y_0\} \in E\}$,
   $Y := Y - (\{y_0\} \cup \{y \in Y : \{y, y_0\} \in E\})$
(2) Return $D$.

---

One can see that at every step, $X$ decreases by at least a $\frac{d}{2}$-fraction, and so, it becomes empty after at most $O(\log n)$ steps. Thus, throughout the algorithm the total number of vertices removed from $Y$ is $O(\log n)\Delta(G[Y]) = o(|Y|)$. This implies that for each $x \in X$, $deg_G(x) \geq \frac{d}{2}|Y|$, which, in turn, guarantees the existence of $y_0$ in the next step unless $X = \emptyset$.

Finally, note that in our application $|X| = \binom{n}{k+1} = \Theta(n^{k+1})$. By Fact 2.2 in [11] we have $deg_G(x) = \Theta(n^{(k-2)k})$. Moreover, $|Y| = \Theta(n^{(k-2)k})$ and $deg_G(y, Y) \leq n^{(k-2)k-1}$, and therefore both conditions in (1) are satisfied. Since $D$ dominates $X$, for every set $S$ of $k+1$ vertices there will be an $S$-absorbing matching contained in the union of matchings of size $k-2$ represented by $D$. Algorithm par-IndDomSet runs in $O(\log^2 n)$ steps.

Now, we are in position to describe our algorithm for small deficient matchings. It will start by constructing a powerful matching $M_0$ in $H$ using the above procedure. Next we remove $M_0$ from $H$ to get a subhypergraph $H'$, with still large minimum co-degree, in which a $(k(k-2)+r)$-deficient matching $M_1$ is computed. Out of the remaining $k(k-2)+r$ vertices, all but $r$ are then sequentially (in $k-2$ steps) absorbed into $M_1$ using absorbing edges from $M_0$.

---

Algorithm par-SmallDefMatch$(r)$, $0 < r \leq k$

---

**In:** $k$-graph $H$ with $\delta(H) \geq \frac{n}{k} + C\log n$ and $n \geq n_0$, $n \equiv r \mod k$.
**Out:** $r$-deficient matching $M$

(1) Compute a powerful matching $M_0$ ($|M_0| \leq \frac{1}{k}C\log n$), as in Definition 9, applying par-IndDomSet to the auxiliary graph $G$ described above.
(2) $H' := H - V(M_0)$ [notice that $\delta(H') \geq \frac{1}{k}|V(H')|$].
(3) Compute a $(k(k-2)+r)$ $-$deficient matching $M_1$ using algorithm par-LargeDefMatch$(k(k-2)+r)$ in $H'$.

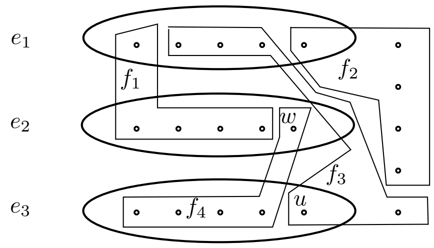


Fig. 3. Absorbing configuration $E = \{e_1, e_2, e_3\}$, $(k = 5)$, $F = \{f_1, f_2, f_3, f_4\}$.

(4) $T := V(H) - (V(M_1) \cup V(M_0))$ [notice that $|T| = k(k-2) + r$].
(5) Repeat until $|T| = r$: [$k - 2$ sequential iterations]

  (a) for an arbitrary set $S \subseteq T$, $|S| = k + 1$, find an $S-$absorbing edge $e \in M_0$ by checking all edges of $M_0$ in **parallel**;
  (b) set $M_0 := M_0 - \{e\} \cup \{e', e''\}$, where $e', e''$ are as in Definition 8;
  (c) $T := V(H) - V(M_0 \cup M_1)$.

(6) Return $M := M_0 \cup M_1$.

---

It is clear that the above algorithm returns an $r$-deficient matching. Its time complexity is dominated by the complexity of PAR-LARGEDEFMATCH and so, it is also $O(\log^3 n)$.

### 3.3. *Proof of Theorem 6*

Our goal now is to build a perfect matching in a $k$-uniform hypergraph on $n \geq n_0$ vertices with $n$ divisible by $k$ and such that $\delta(H) \geq \frac{n}{2} + \frac{k}{4}$ . In our construction we will apply an *absorbing configuration* motivated by the proof in [12].

**Definition 10 (absorbing configuration)** *Given a set $S$ of $k$ vertices, a set of vertex disjoint edges $E \subseteq H - S$ is called* $S$-absorbing configuration *if there is another set of disjoint edges $F \subseteq H$ such that $\bigcup_{f \in F} f = \bigcup_{e \in E} e \cup S$. (See Fig. 3.)*

Observe that if $S$ is outside a matching $M$ which contains an $S$-absorbing configuration $E$ then $M$ can "swallow" $S$ by swapping $E$ for $F$. Note also that it follows from Definition 10 that $|F| = |E| + 1$.

The algorithm will first use the previous procedure from Theorem 5 to construct a $k$-deficient matching $M_1$. Then it will search for an absorbing set of edges in $M_1$ of size 1, 2, or 3, to absorb the remaining vertices into a perfect matching $M$. It will follow from the proof in [12] that such a configuration exists.

---

ALGORITHM PAR-PERFECTMATCH

---

**In:** $k$-graph $H$ with $\delta(H) \geq \frac{n}{k} + \frac{k}{4}$ and $n \geq n_0$, $n \equiv 0 \mod k$.
**Out:** perfect matching $M$

(1) Compute a $k$-deficient matching $M_1$ using the parallel algorithm PAR-SMALLDEFMATCH($k$) in $H$.
(2) $S := V(H) - V(M_1)$. If $S \in H$, $M := M_1 \cup \{S\}$ and go to (5)
(3) For every set of edges $E \subset M_1$, $|E| \leq 3$, **in parallel** check if it forms an $S$-absorbing configuration as in Definition 10.
(4) Use the absorbing configuration found in Step (3) to absorb the vertices of $S$ and obtain a perfect matching $M := (M_1 - E) \cup F$.
(5) Return $M$.

---

It remains to show that an $S$-absorbing configuration searched for in Step (3) does exist. If it does, it then can be found in parallel in constant time with processors assigned to all sets $E \subset M_1$, $|E| \leq 3$.

For every $u \in V(M_1)$, let $e_u$ denote the edge of $M_1$ containing $u$. For every $v \in V(H)$ define the set $T_{M_1}(v) := \{u \in V(M_1) : e_u - \{u\} \cup \{v\} \in H\}$ and set $t_{M_1}(v) := |T_{M_1}(v)|$. Further, set $S = \{x_1, \dots, x_k\}$. By Observation 1 in [12], if $t_{M_1}(x_i) > \frac{n}{2} - \frac{5}{4}k$ for some $1 \leq i \leq k$, then, since $S \notin H$,

$$|N(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_k) \cap T_{M_1}(x_i)| > (\tfrac{n}{2} + \tfrac{k}{4}) + (\tfrac{n}{2} - \tfrac{5}{4}k) - (n - k) = 0.$$

Thus, there exists a vertex $y \in V(M_1)$ such that

$$f_1 = \{y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_k\} \in H \text{ and } f_2 = e_y - \{y\} \cup \{x_i\} \in H.$$

Consequently, $E = \{e_y\}$ is an $S$-absorbing configuration with $F = \{f_1, f_2\}$.

If $t_{M_1}(x_i) \leq \frac{n}{2} - \frac{5}{4}k$ for all $i = 1, \dots, k$, then, by Observation 2 in [12], there is a vertex $w \in V(M_1)$ with $t_{M_1}(w) > \frac{n}{2} - \frac{k}{4}$. Hence, by Observation 3 therein, there exists $e_1 = \{v_1, v_2, \dots, v_k\} \in M_1 - \{e_w\}$ such that $f_1 = e_w \cup \{v_1\} - \{w\} \in H$ and $f_2 = \{x_1, \dots, x_{k-1}, v_2\} \in H$. Set $e_2 := e_w$ and $M' = M_1 \cup \{f_1, f_2\} - \{e_1, e_2\}$, and observe that $w \notin V(M')$ and $t_{M'}(w) > \frac{n}{2} - \frac{5}{4}k$. Again by Observation 1 in [12], there is $u \in N(v_3, \dots, v_k, x_k) \cap T_{M'}(w)$. If $u \in f_1$ then $E = \{e_1, e_2\}$ is $S$-absorbing with $F$ consisting of $f_2$, $f_3 = \{u, v_3, \dots, v_k, x_k\}$ and $f_1 \cup \{w\} - \{u\}$. The case $u \in f_2$ is similar. Finally, if $u \notin f_1 \cup f_2$, then the edge containing $u$ in $M'$ is the same as in $M_1$, that is $e_u$. So, $\{e_1, e_2, e_3\}$ is an $S$-absorbing configuration with $F = \{f_1, f_2, f_3, f_4\}$, where $e_3 := e_u$ and $f_4 := e_3 \cup \{w\} - \{u\}$. This last case is depicted in Fig. 3.

## 4. Toward Understanding the Hardness Gap

### 4.1. *Proof of Proposition 2*

In this section an effective reduction proving Proposition 2 is presented. Given a $k$-graph $H$ with $|V(H)| = n$ divisible by $k$, we construct a gadget $H' = (V', E')$ as follows. Let $M$ be a matching of size $n/k$ disjoint from $H$. We let $V' = V \cup V(M)$ and $E' = E(H) \cup E(M) \cup E^*$, where $E^* = E(M^*)$ is the edge set of a suitably chosen matching of size $2n/k - 1$ whose each edge intersects both, $V(H)$ and $V(M)$ (see Fig. 4). More specifically, let $M = \{e_1, \dots, e_{n/k}\}$ and $e_i = \{v_i^1, \dots, v_i^k\}$, $i = 1, \dots, n/k$.

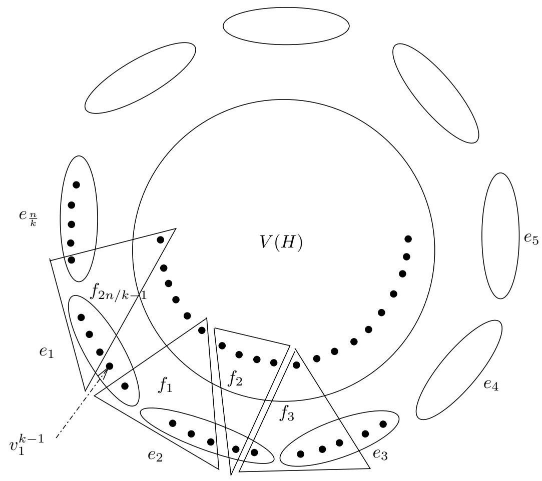


Fig. 4. Gadget for $k$-deficient matchings ($k = 5$).

Further, let $V(H) = \{u_1, \ldots, u_n\}$. Then $M^*$ consists of edges

$$f_1 = \{u_1, v_1^k, v_2^1, \ldots, v_2^{k-2}\}, \qquad f_2 = \{u_2, \ldots, u_k, v_2^{k-1}\},$$
$$f_3 = \{u_{k+1}, v_2^k, v_3^1, \ldots, v_3^{k-2}\}, \ldots, f_{2n/k-1} = \{u_{n-k+1}, v_{n/k}^k, v_1^1, \ldots, v_1^{k-2}\}.$$

Note that vertex $v_1^{k-1}$ is the only vertex of $M$ not belonging to $M^*$. Moreover, $M^*$ is a $k$-deficient matching in $H'$, so that $H'$ is a legal input of the restricted problem.

Clearly, if $H$ has a perfect matching $M_H$ then $M_H \cup M$ is a perfect matching of $H'$. Conversely, if $H'$ has a perfect matching $M'$ then, because of $v_1^{k-1}$, $e_1 \in M'$. Consequently, $f_1$, which intersects $e_1$, is not in $M'$, and thus, $e_2 \in M'$. Iterating this argument, we conclude that $M = \{e_1, \ldots, e_{n/k}\} \subset M'$, which implies that $M' - M$ is a perfect matching of $H$. This shows that the NP-complete problem PM($k$) reduces polynomially to its restricted version.

### 4.2. *Proof of Theorem 3*

Here we give a detailed sketch of the proof of Theorem 3. We will only describe a decision algorithm PerfectMatch, leaving out the additional, quite involved procedure which finds a perfect matching every time the answer is YES. (However, in the course of analysis of the algorithm, at least in some cases we will provide hints to how a perfect matching can be actually found.) The algorithm is based on a modification of the proof from [11], where under the assumption that $\delta(H) \geq t(k, n, 0) = \frac{n}{2} + O(1)$ two cases were studied separately: when $H$ is close, in some sense, to a critical $k$-graph and when it is far from it. In the former case, $H$ is almost complete in a bipartite sense and this fact is used to build a perfect matching "manually". In the latter case, some absorbing configurations are utilized.

First let us recall some definitions and facts from [11]. Given (not necessarily disjoint) sets $N_1, \ldots, N_k \subseteq V(H)$, denote by $E_H(N_1, \ldots, N_k)$ the set of ordered $k$-tuples of *distinct* vertices $(v_1, \ldots, v_k)$ such that $v_i \in N_i$, $i = 1, \ldots, k$, and $\{v_1, \ldots, v_k\} \in H$. For $\gamma > 0$, let

$$\Lambda(\gamma) = \{(v_1, \ldots, v_{k-1}) : deg_H(v_1 \ldots, v_{k-1}) > (\tfrac{1}{2} + 2\gamma)n\}.$$

Consider the following pair of conditions.

(i) For all choices of $(k-1)$-element sets $S_1, \ldots, S_k \subseteq V(H)$, we have $|E_H(N_H(S_1), \ldots, N_H(S_k))| \geq \frac{n^k}{\log^3 n}$;

(ii) $|\Lambda(\gamma)| \geq \frac{n^{k-1}}{\log n}$.

Claim 5.2 in [11] asserts that if $H$ is a $k$-graph on $n > n_0$ vertices, $n$ divisible by $k$, $\delta(H) \geq (\frac{1}{2} - \gamma)n$, and at least one of the above conditions holds, then $H$ has a perfect matching. In [11], $\gamma$ was chosen to be $\frac{1}{\log n}$, but it was also observed that the proof goes through for a sufficiently small constant $\gamma > 0$. As mentioned earlier, Szymańska [14] showed how to turn that proof into an algorithm finding a perfect matching.

Given a $k$-graph $H$ and a partition $V(H) = A \cup B$, we define for each $r = 0, 1, \ldots, k$, the set $E_r := E_r(A, B)$ of all edges of $H$ intersecting $A$ in precisely $r$ vertices (and $B$ in $k - r$ vertices). A $k$-graph, which consists of all $k$-element subsets of $A \cup B$ intersecting $A$ in $r$ vertices is denoted by $K_r := K_r(A, B)$. Further, for a given $\gamma'$ we say that a partition $V(H) = A \cup B$ is *$\gamma'$-even-complete* [*$\gamma'$-odd-complete*] if for all even [odd] $r$, $|K_r - E_r| < \gamma' n^k$. A partition is $\gamma'$-complete if it is $\gamma'$-even-complete or $\gamma'$-odd-complete.

It follows from the proof of Claim 5.1 in [11] that if neither (i) nor (ii) hold then one can find a partition $V(H) = A \cup B$ which is $\gamma'$-complete for some $\gamma' = \gamma'(\gamma)$, where $\gamma'(\gamma)$ is a decreasing function. Having such a partition we will follow the lines of the proof from Section 4 of [11]. It is based on four facts, Facts 4.1-4.4, which require that $\gamma'$ is small enough. In addition, Fact 4.4(b) assumes that $\delta(H) \geq \frac{n}{2} - O(1)$, but this can be relaxed to $\delta(H) \geq (\frac{1}{2} - \epsilon')n$ for sufficiently small $\epsilon'$.

Let $\gamma_0$ be such that $\gamma'(\gamma_0)$ is small enough for Facts 4.1-4.4 in [11] to hold. We will prove our Theorem 3 with $\epsilon = \min(\gamma, \gamma_0, \epsilon')$.

---

Algorithm PerfectMatch

---

**In:** $k$-graph $H$ with $\delta(H) \geq (\frac{1}{2} - \epsilon)n$ and $n \geq n_0$, $n \equiv 0 \mod k$.
**Out:** YES if $H$ has a perfect matching, NO otherwise.

(1) For all $S \subset V$, $|S| = k - 1$, compute $deg_H(S)$ and check condition (ii). If (ii) holds, return YES.

(2) Otherwise, for all $(S_1, \ldots, S_k)$, $S_i \subset V$, $|S_i| = k - 1$, $i = 1, \ldots, k$, compute $|E_H(N_H(S_1), \ldots, N_H(S_k))|$ and check condition (i). If (i) holds, return YES.

(3) Otherwise, set $A := N_H(S_1)$, $B := V - A$, where $(S_1, \ldots, S_k)$ is a $k$-tuple violating condition (i).

(4) Decide if $H$ has a perfect matching using algorithm PerfectMatchInComplete$(H, A, B, \gamma')$.

---

For the description of PerfectMatchInComplete we need one more notation. Given a partition $V = A \cup B$, a vertex $v \in V$ is called *$\alpha$-small* in $E_r$, $0 \le r \le k$, $0 < \alpha < 1$, if $deg_{E_r}(v) \le \alpha \cdot deg_{K_r}(v)$, and is called *$\alpha$-large* otherwise.

---

Algorithm PerfectMatchInComplete

---

**In:** $k$-graph $H$ with $\delta(H) \ge (\frac{1}{2} - \epsilon)n$ and $n \ge n_0$, $n \equiv 0 \mod k$ and an $\gamma'$-complete partition $V(H) = A \cup B$.
**Out:** YES if $H$ has a perfect matching, NO otherwise.

(1) If $k$ is odd and $(A, B)$ is $\gamma'$-odd-complete, swap $A$ and $B$ around;
(2) If $(A, B)$ is $\gamma'$-even-complete, set $k' = k - 1$ if $k$ is odd and $k' = k - 2$ otherwise and do:
   (a) Identify the set $S$ of all 0.3-small vertices of $E_{k'}$ and move them to the other side, that is, reset $A := A \triangle S$ and $B := B \triangle S$.
   (b) If $|A|$ is even or $\bigcup_{r\ odd} E_r \ne \emptyset$, return YES
   (c) Return NO
(3) If $(A, B)$ is $\gamma'$-odd-complete (and so $k$ is even) set $k' = \frac{k}{2} + 1$ if $k$ is divisible by 4 and $k' = \frac{k}{2}$ otherwise and do:
   (a) Identify the set $S$ of all 0.3-small vertices of $E_{k'}$; reset $A := A \triangle S$ and $B := B \triangle S$.
   (b) If $|A| \equiv \frac{n}{k} \mod 2$ or $\bigcup_{r\ even} E_r \ne \emptyset$, return YES.
   (c) Return NO.

---

First, let us verify the correctness and complexity of PerfectMatch.

Our algorithm first checks if either (i) or (ii) holds. (Note that in [11], all $k$-tuples of sets $N_1, \ldots, N_k$ of size $|N_i| \ge (\frac{1}{2} - \gamma)n$ were checked to verify (i); here, in order to be efficient, we look only at the neighborhood sets which is sufficient.) If (i) or (ii) holds, then, by Claim 5.2 in [11] with constant $\gamma > 0$ the answer is YES. Otherwise, we have found sets $N_i = N_H(S_i)$, $i = 1, \ldots, k$, where $S_i$'s violate $(i)$, and we know that $|\Lambda(\gamma)| \le \frac{n^{k-1}}{\log n}$. It can be deduced from the proof of Claim 5.1 in [11] that then, for all $i$, $|N_i| < n/2 + 2\gamma n$ and, taking, say, $A = N_1$ and $B = V - N_1$, we obtain a $\gamma'$-complete partition. The most time consuming Step is (2), where we have to compute $|E_H(N_H(S_1), \ldots, N_H(S_k))|$ for, roughly, $\binom{n}{k-1}^k$ instances.

We now verify the correctness of PerfectMatchInComplete. The answers NO are easy to explain, because they are accompanied by a witness in the form of an $(A, B)$ partition which prevents the existence of a perfect matching in $H$. In Step 2(c) $|A|$ is odd, while all edges of $H$ intersect $A$ in an even number of vertices. In Step 3(c), $|A| \not\equiv \frac{n}{k} \mod 2$ and every edge of $H$ intersects $A$ in an odd number of vertices. If there existed a perfect matching $M$ in $H$, then every edge of

$M$ would saturate an odd number of vertices of $A$, and so $|V(M) \cap A| \equiv \frac{n}{k} \mod 2$, a contradiction.

Next we will move to the explanation of the answer YES in Steps 2(b) and 3(b). To do this we will follow a modified proof from Section 4 of [11]. This modification is necessary, because in [11] there was a stronger assumption $\delta(H) \geq t(k, n, 0)$ under which all $k$-graphs $H$ do have a perfect matching. We need another notion, strongly related to that of an $\alpha$-small vertex. We say that $v \in V$ is *$\alpha$-deficient* in $E_r$, $0 \leq r \leq k$, $0 < \alpha < 1$, if $deg_{E_r}(v) \leq deg_{K_r}(v) - \alpha m^{k-1}$.

Assume first that $(A, B)$ is $\gamma'$-even-complete and $k$ is odd. If $|A|$ is odd but there exists in $H$ an edge $e_0$ such that $|e_0 \cap A|$ is odd, then reset $A := A - \{e_0\}$, $B := B - \{e_0\}$ to get the size of $A$ even. If $|A|$ is even, one can build a perfect matching $M$ in $H$ from the following ingredients. One of them is the edge $e_0$ if it was indeed needed. Let $N$ be the set of all $\sqrt{\gamma'}$-deficient vertices in $E_0$, $E_{k-3}$ or $E_{k-1}$. By Fact 4.2 in [11], $|N| \leq 3\sqrt{\gamma'} \cdot kn$, and by Fact 4.4 therein, all vertices of $E_{k-1}$ are 0.2-large, so we may apply Fact 4.3 to $N$, obtaining a matching $M_1 \subset E_{k-1}$ of size $|N|$ which matches all vertices of $N$. Reset $A := A - V(M_1)$ and $B := B - V(M_1)$. Let $a = |A|$, $b = |B|$ and $a + b = sk$. Note that $0 \leq n - sk \leq 3\sqrt{\gamma'} \cdot k^2 n + k$.

The rest of $M$ will be composed of partial matchings $M_2 \subset E_0$, $M_3 \subset E_{k-3}$, and $M_4 \subset E_{k-1}$. Their existence is guaranteed by Fact 4.1 from [11] which we quote here in a suitable form.

**Fact 11 (Fact 4.1, [11])** *Let $\alpha < (2k)^{-2k}$. If $|A| = 0 \mod r$, $|B| = 0 \mod k - r$, $|A| + |B| = 0 \mod k$ and no vertex is $\alpha$-deficient in $E_r$, then $E_r$ has a perfect matching.*

We are in position to apply Fact 11 because after removing $V(M_1)$, there are no $\sqrt[3]{\gamma'}$-deficient vertices in $E_0$, $E_{k-3}$ or $E_{k-1}$. (We write $\sqrt[3]{\gamma'}$ instead of $\sqrt{\gamma'}$, because we apply Fact 11 to smaller and smaller sets $V$.)

To obtain $M_2$, $M_3$ and $M_4$ we just need to find the right proportions of these three matchings. This, however, boils down to solving a system of equations. Let $x = |M_2|$, $y = |M_3|$, and $z = |M_4|$. Then, we must have $(k-3)y + (k-1)z = a$ and $kx + 3y + 2z = b$. Expressing $y, z$ in terms of $x$, we obtain the solution

$$y = \frac{k-1}{2}(s - x) - \frac{1}{2}a, \qquad z = \frac{1}{2}a - \frac{k-3}{2}(s - x).$$

For $y$ and $z$ to be nonnegative, we need to choose a nonnegative integer $x$ so that

$$\frac{a}{k-1} \leq s - x \leq \frac{a}{k-3}.$$

This is feasible, because $s$ is close to $2a/k$ (and thus much bigger than $a/(k-1)$). Note that $y$ and $z$ are integer too. This checks that for $k$ odd the answer YES in 2(b) is correct.

For $k$ even, Step 2(b) is very similar. This time we let $N$ be the set of all $\sqrt{\gamma'}$-deficient vertices in $E_0$, $E_{k-4}$ or $E_{k-2}$, and build $M$ of $e_0$ (if needed), $M_1 \subset E_{k-2}$, $M_2 \subset E_0$, $M_3 \subset E_{k-4}$, and $M_4 \subset E_{k-2}$. The system of equations is $(k-4)y + (k-$

$2)z = a$ and $kx + 4y + 2z = b$ and has a positive, integer solution

$$y = \left(\frac{k}{2} - 1\right)(s - x) - \frac{a}{2}, \qquad z = \frac{a}{2}\left(\frac{k}{2} - 2\right)(s - x),$$

where

$$\frac{a}{k-2} \le s - x \le \frac{a}{k-4}.$$

Slightly more involved is Step 3(b), where we assume that $(A, B)$ is $\gamma'$-odd-complete. By Step (1) we know that $k$ must be even, since otherwise, swapping $A$ and $B$ around would result in a $\gamma'$-even-complete partition.

If $|A| \not\equiv \frac{n}{k} \mod 2$, but there exists in $H$ an edge $e_0$ such that $r := |e_0 \cap A|$ is even, then reset $A := A - \{e_0\}$, $B := B - \{e_0\}$. Note that after removing $e_0$, $n/k$ has decreased by one, while $a$ has decreased by $r$, an even number. Thus, for this new set $A$ and with $n := n - k$, we have $|A| \equiv \frac{n}{k} \mod 2$.

Assume first that $k$ is divisible by 4. Define $N$ to be the set of all $\sqrt{\gamma'}$-deficient vertices in $E_{k/2-1}$ or $E_{k/2+1}$, and build $M$ of $e_0$ (if needed), $M_1 \subset E_{k/2-1}$, $M_2 \subset E_{k/2-1}$, and $M_3 \subset E_{k/2+1}$. Denoting $x = |M_2|$, $y = |M_3|$, and $k = 2\ell = 4t$, we thus have a system of equations $(\ell - 1)x + (\ell + 1)y = a$ and $(\ell + 1)x + (\ell - 1)y = b$ with the solution

$$x = \frac{1}{2}\left(s + \frac{b-a}{2}\right) = ts + \frac{s-a}{2}, \quad y = \frac{1}{2}\left(s + \frac{a-b}{2}\right) = \frac{s+a}{2} - ts.$$

Note that removing an edge of $E_{\ell-1}$ changes the parity of both, $a$ and $s$, and hence we do have the congruence $a \equiv s \mod 2$. Consequently, $x$ and $y$ are integer, and also nonnegative, because $|a - b|$ is small.

If $k \equiv 2 \mod 4$, we need to consider two further subcases: $a \ge b$ and $a < b$. If $a \ge b$, let $N$ be the set of all $\sqrt{\gamma'}$-deficient vertices in $E_{k/2}$ or $E_{k/2+2}$. Build $M_1 \subseteq E_{k/2}$ as before and note that, again, $a \equiv s \mod 2$. We find $M_2 \subset E_{k/2+2}$ by Fact 4.3 from [11] and then $M_3 \subset E_{k/2}$ using Fact 11 with

$$x = \frac{a+b}{k} - y, \qquad y = \frac{a-b}{4}.$$

Since $4|(a - b)$ and $a \le b(k + 4)/(k - 4)$, both $x$ and $y$ are nonnegative integers. (We could not apply Fact 11 to obtain $M_2$ because $y$ is too small.)

Finally, if $a < b$, we replace $E_{k/2+2}$ with $E_{k/2-2}$ but otherwise proceed as before. Now, $x = (a+b)/k - y$ and $y = (b-a)/4$, both, again, nonnegative integers, because $b \le a(k + 4)/(k - 4)$.

## Concluding Remarks

**Remark 2.** *We have not tried to optimize the value of $\epsilon$ for which Theorem 3 remains true. Certainly, it decreases with $k$, but even for $k = 3$ our analysis forces $\epsilon$ to be quite small. Note that if for $k = 3$ we could push $\epsilon$ up to $\frac{1}{6}$, then we would completely cover the gap and show that $\boldsymbol{PM}(k, 0, c)$ is in $\mathcal{P}$ for all $c > 1/k$.*

**Remark 3.** *One can formulate a problem similar to PM*$(k, 0, c)$*, but with* $\delta(H)$ *replaced by other versions of minimum hypergraph degrees. For* $1 \le l \le k-1$*, let* $\delta_l(H)$ *be the largest integer* $d$ *such that every* $l$*-element subset of vertices is contained in at least* $d$ *edges of* $H$*. Recently, it was proved in [5] that if* $\delta_1(H) > (\frac{5}{9} + \epsilon)\binom{n}{2}$*, then* $H$ *contains a perfect matching. This was complemented in [14], where it was shown that the problem of deciding if a* $k$*-graph* $H$ *with* $\delta_1(H) \ge c\binom{|V(H)|}{2}$ *contains a perfect matching is NP-complete for* $c < \frac{5}{9}$*. So, unlike for PM*$(k, 0, c)$*, there is no hardness gap left here. Also, this could be related to the fact that there are 3-graphs* $H$ *with* $\delta_1(H) \sim \frac{5}{9}\binom{n}{2}$ *with no* $r$*-deficient matching for any* $r = o(n)$*.*

**Remark 4.** *There are similar results regarding the complexity of the problem* $\boldsymbol{HAM}(k, c)$ *deciding the existence of a Hamilton cycle (as defined in, e.g., [10]) in a* $k$*-graph with* $\delta(H) \ge c|V(H)|$*. It was shown in [10] that for all* $k \ge 3$*,* $c > \frac{1}{2}$*, and sufficiently large* $n$*, every* $k$*-graph* $H$ *with* $|V(H)| = n$ *and* $\delta(H) \ge cn$ *contains a Hamilton cycle. Hence,* $\boldsymbol{HAM}(k, c)$ *is trivial for all* $c > \frac{1}{2}$*. On the other hand, for* $c < \frac{1}{k}$ *we were able to prove recently (cf. [8]) that* $\boldsymbol{HAM}(k, c)$ *is NP-complete. Interestingly, it leaves a similar hardness gap* $(\frac{1}{k}, \frac{1}{2})$ *as for the problem* $\boldsymbol{PM}(k, 0, c)$*. Note that this gap collapses for graphs (*$k = 2$*), see [2].*

## References

[1] A. Asadpour, U. Feige and A. Saberi. Santa Claus Meets Hypergraph Matchings, APPROX-RANDOM 2008 10–20.

[2] E. Dahlhaus, P. Hajnal, M. Karpiński, On the parallel complexity of Hamiltonian cycle and matching problem on dense graphs, J. Algorithms 15(3) (1993) 367-384.

[3] J. Edmonds, Paths, trees and flowers, Canad J. Math. 17 (1965) 449-467.

[4] M.R. Garey, D.S. Johnson, Computers and intractability, Freeman, 1979

[5] H. Han, Y. Person, and M. Schacht, On perfect matchings in uniform hypergraphs with large minimum vertex degree, SIAM Journal on Discrete Mathematics 23(2) (2009) 732-748.

[6] P.E. Haxell. A Condition for Matchability in Hypergraphs, Graphs and Combinatorics 11 (1995) 245-248.

[7] M. Karpiński, A. Ruciński, and E. Szymańska, The Complexity of Perfect Matching Problems on Dense Hypergraphs, ISAAC 2009 LNCS 5878 626–636.

[8] M. Karpiński, A. Ruciński, and E. Szymańska, Computational Complexity of the Hamiltonian Cycle Problem in Dense Hypergraphs, LATIN 2010 662-673.

[9] M. Luby, A simple parallel algorithm for the maximal independent set problem, SIAM J. Comput. 15(4) (1986) 1036-1053.

[10] V. Rödl, A. Ruciński and E. Szemerédi, An approximate Dirac-type theorem for $k$-uniform hypergraphs, Combinatorica 28 (2) (2008) 229-260.

[11] V. Rödl, A. Ruciński, E. Szemerédi, Perfect matchings in large uniform hypergraphs with large minimum collective degree, JCT A 116(3) (2009) 613-636.

[12] V. Rödl, A. Ruciński, M. Schacht, and E. Szemerédi, A note on perfect matchings in uniform hypergraphs with large minimum collective degree, Commen. Math. Univ. Carol. 49(4) (2008) 633-636

[13] E. Szymańska, The Complexity of Almost Perfect Matchings in Uniform Hypergraphs with High Codegree, IWOCA 2009 LNCS 5874 438–449.

[14] E. Szymańska, The Complexity of Almost Perfect Matchings and Other Packing Problems in Uniform Hypergraphs with High Codegree, *submitted*

[15] G. Särközy, A fast parallel algorithm for finding Hamiltonian cycles in dense graphs. Discrete Mathematics 309 (2009) 1611–1622.