

# The Complexity of Perfect Matching Problems on Dense Hypergraphs

Marek Karpiński<sup>1</sup> and Andrzej Ruciński<sup>\*2</sup> and Edyta Szymańska<sup>\*\*2</sup>

<sup>1</sup> Department of Computer Science, University of Bonn. Email: marek@cs.uni-bonn.de

<sup>2</sup> Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Poznań. Email: rucinski,edka@amu.edu.pl

**Abstract.** In this paper we consider the computational complexity of deciding the existence of a perfect matching in certain classes of dense  $k$ -uniform hypergraphs. Some of these problems are known to be notoriously hard. There is also a renewed interest recently in the very special cases of them. One of the goals of this paper is to shed some light on the tractability barriers for those problems.

It has been known that the perfect matching problems are NP-complete for the classes of hypergraphs  $H$  with *minimum*  $((k-1)$ -wise) *vertex degree*  $\delta$  at least  $c|V(H)|$  for  $c < \frac{1}{k}$  and trivial for  $c \geq \frac{1}{2}$ , leaving the status of the problems with  $c$  in the interval  $[\frac{1}{k}, \frac{1}{2})$  widely open. In this paper we show, somehow surprisingly, that  $\frac{1}{2}$ , in fact, is not a *threshold* for the 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  $\delta$  at least  $(\frac{1}{2} - \epsilon)|V(H)|$  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 in view of the known results for graphs.

## 1 Introduction

In recent years hypergraphs gained a lot of interest as a natural generalization of graphs as well as a model for certain discrete optimization problems. For instance, Asadpour, Feige and Saberi [AFS08] reduced a max-min allocation problem, known as *the Santa Claus Problem*, to finding a perfect matching in a class of bipartite hypergraphs. Since they relied on a rather non-constructive, Hall-type sufficient condition of Haxell [H95], they 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. [RRS09]. Recall that the celebrated

---

\* Research supported by grant N201 036 32/2546.

\*\* Research supported by grant N206 017 32/2452

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 [RRS09], the authors used the minimum degree of a  $(k-1)$ -tuple of vertices in a  $k$ -uniform hypergraph and determined the best possible bound on this parameter guaranteeing a perfect matching. If  $n$  is sufficiently large and divisible by  $k$ , then the threshold values turned out to be close to  $\frac{1}{2}n$ .

As a consequence, the decision problem asking whether a given  $k$ -uniform hypergraph with the minimum  $(k-1)$ -wise degree above  $\frac{1}{2}n$  contains a perfect matching is trivial. Szymańska observed in [S09] that the argument presented in [RRS09] can be transformed into a deterministic polynomial time algorithm. Moreover, she also showed that answering the question whether a  $k$ -uniform hypergraph with minimum  $(k-1)$ -wise vertex degree at least  $c|V(H)|$ ,  $c < \frac{1}{k}$ , contains a perfect matching is NP-complete.

Those results leave a “hardness gap” between  $\frac{1}{k}$  and  $\frac{1}{2}$ . By the counterexamples introduced in [RRS09] it is apparent that in this case there exist hypergraphs of minimum  $(k-1)$ -wise vertex degree below  $\frac{1}{2}|V(H)|$  without a perfect matching. 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, when still both answers, *yes* and *no* are possible, the problem is polynomial. Indeed, in this paper we provide a polynomial time algorithm which for every hypergraph of the minimum  $(k-1)$ -wise vertex degree at least  $(\frac{1}{2} - \epsilon)|V(H)|$  constructs a perfect matching if one exists and otherwise exhibits a certificate for non-existence (cf. Theorem 2).

Our second result regards the parallelization of the problem. Parallel algorithms have experienced a lot of attention in late eighties and nineties of last century. Many interesting algorithms were given, including several deterministic algorithms placing the maximal independent set in the class  $NC$ , and 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. For regular graphs however, the problem is  $AC^0$ -equivalent with the general perfect matching problem for arbitrary graphs [DK92]. 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 with minimum degree  $\delta \geq \frac{n}{2}$ .

Dalhaus, Hajnal and Karpiński in [DKH93] gave an  $NC^2$  parallel algorithm finding a perfect matching in such graphs and showed that for the minimum degree at least  $cn$ ,  $c < \frac{1}{2}$ , the problem is as hard as for all graphs. Recently, Sárközy [Sa09] 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 [DKH93] and [Sa09], 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 5 implies that the problem of deciding whether a given  $k$ -uniform hypergraph  $H$ , with minimum  $(k-1)$ -wise vertex 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  $(k-1)$ -wise degrees (cf. Theorems 3,4). These algorithms serve as subroutines in the main algorithm.

In the following subsections we introduce our notation, define formally the problems in question and state our results (Theorems 2, 3, 4 and 5, and Proposition 1). Section 2 contains three parallel algorithms and their analysis which proves Theorems 3, 4 and 5. The last section is devoted to an outline of the proof of Theorem 2. A complete proof of Theorem 2, as well as a proof of Proposition 1, will be presented in a full version of the paper.

### 1.1 Basic definitions and notation

**Hypergraphs.** A *hypergraph*  $H = (V, E)$  is a finite set of vertices  $V$  together with a family  $E$  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  $|V(H) \setminus V(M)|$  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  $n \equiv q \pmod{k}$ , then  $r$ -deficient matchings are possible if and 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  $\frac{1}{k}|V(H)|$ . So, a necessary condition for the existence of a perfect matching in  $H$  is that  $|V(H)| \equiv 0 \pmod{k}$ .

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  $\delta(H) = \min_S \deg_H(S)$  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.

**Three classes of computational problems.** For  $k \geq 2$ , by  $\mathbf{PM}(k)$  we denote the problem of deciding whether a  $k$ -graph  $H = (V, E)$  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 [Ed65]. 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  $k \geq 3$  [GJ79].

Given integers  $k \geq 3$  and  $r \geq 0$ , let  $\mathbf{PM}(k, r)$  denote the problem of deciding whether a  $k$ -graph  $H = (V, E)$  with  $|V(H)| \equiv r \pmod{k}$  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  $\mathbf{PM}(k, 0) = \mathbf{PM}(k)$ .

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  $H = (V, E)$  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 Known results

**Existential results.** Let us begin with the perfect case  $r = 0$ . For  $k = 2$  (graphs), it is very easy to show that, for even  $n$ ,  $t(2, n, 0) = \frac{1}{2}n$ . For all integers  $k \geq 3$  and sufficiently large  $n \geq k$ , the value of  $t(k, n, 0)$  is exactly determined in [RRS09]. It is 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  $\frac{1}{k}n$ , and satisfying  $\frac{3}{2} \leq c_{k,n} \leq 3$ . Hence, in particular,  $\frac{1}{2}n - k + \frac{3}{2} \leq t(k, n, 0) \leq \frac{1}{2}n - k + 3 \leq \frac{1}{2}n$ . In [RRSS08] 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.

As for the deficient matchings (case  $r > 0$ ), a striking difference between perfect and almost perfect matchings was observed in [RRS09]. Indeed, it was proved there that for  $n \equiv r \pmod{k}$  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  $\frac{1}{k}n$ , while in the perfect case it is around  $\frac{1}{2}n$ .

**Computational results.** An immediate consequence of the results in [RRS09] 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.)

In [S09], it was shown 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  $\mathbf{PM}(k)$ . 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  $\mathbf{PM}(k, 0, c)$ .

On the positive side, [S09] provided a polynomial time algorithm for the corresponding search problem when  $c > \frac{1}{k}$ , and  $r > 0$ . It was also observed in [S09] that the existential proof from [RRS09] can be turned into a polynomial time algorithm finding a perfect matching when  $c \geq \frac{1}{2}$ .

### 1.3 New 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 2 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 [RRSS08]. In fact, we turned that proof into a parallel algorithm (see Theorem 5 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 in detail.

**Hardness taxonomy.** Concerning the problem  $\mathbf{PM}(k, 0, c)$ , the results from [RRS09] and [S09] described in Section 1.2 have left a hardness gap for  $c \in [\frac{1}{k}, \frac{1}{2})$ .

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

We present two results which suggest different answers to this problem. To put the first of them into a right context, recall that by [RRS09] 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 1.** *For all  $k \geq 3$ ,  $\mathbf{PM}(k)$  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 3 we describe an algorithm, called PERFECTMATCH, which, for some  $c < \frac{1}{2}$  places  $\mathbf{PM}(k, 0, c)$  in  $\mathcal{P}$ .

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

*Remark 1.* Theorem 2 reveals an interesting feature: it provides a polynomial time algorithm which, unlike the algorithms in [DKH93], [Sa09], [S09], 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).

**Parallel Algorithms.** As the model of computation we choose the version EREW PRAM. Recall that, as shown in [RRS09], 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 [S09], the existential proof from [RRS09] 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  $NC = \bigcup_i NC^i$ , 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,  $\text{PAR-PERFECTMATCH}$ , is based on the existential proof in [RRSS08] and uses as subroutines two other parallel algorithms of independent interest,  $\text{PAR-LARGEDEFMATCH}(r)$  and  $\text{PAR-SMALLDEFMATCH}(r)$ , 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 3.** *For every  $k \geq 3$  and  $r \geq (k-2)k$  there exists a constant  $n_0$ , and a parallel algorithm, called  $\text{PAR-LARGEDEFMATCH}(r)$ , which in every  $k$ -graph  $H$  on  $n \geq n_0$  vertices with  $n \equiv r \pmod{k}$  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  $\mathbf{PM}(k, r, c)$  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  $\text{PAR-SMALLDEFMATCH}(r)$ , given below, uses the algorithm from Theorem 3 as a subroutine.

**Theorem 4.** *For every  $k \geq 3$  and  $0 < r < (k-2)k$  there exist constants  $n_0$  and  $C > 0$ , and a parallel algorithm, called  $\text{PAR-SMALLDEFMATCH}(r)$ , which in every  $k$ -graph on  $n \geq n_0$  vertices with  $n \equiv r \pmod{k}$  and  $\delta(H) \geq \frac{r}{k} + C \log n$  finds an  $r$ -deficient matching in a polylogarithmic number of rounds using a polynomial number of processors. It follows that the search version of  $\mathbf{PM}(k, r, c)$  is in the class  $NC$  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 5.** *For every  $k \geq 3$  there exists constant  $n_0$ , and a parallel algorithm, called  $\text{PAR-PERFECTMATCH}$ , which in every  $k$ -uniform hypergraph on  $n \geq n_0$  vertices with  $n$  divisible by  $k$  and such that  $\delta(H) \geq \frac{n}{2} + \frac{k}{4}$  finds a perfect matching in a polylogarithmic number of rounds using a polynomial number of processors. It follows that the search version of  $\mathbf{PM}(k, 0, c)$  is in the class  $NC$  for  $c > \frac{1}{2}$ .*

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.

## 2 Description and analysis of parallel algorithms

In this section we prove Theorems 3–5. Each proof consists of a description of the algorithm followed by a proof of its correctness.

**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.

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

### 2.1 Proof of Theorem 3

The construction below generalizes the ideas from [DKH93] 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 a *MIS algorithm*, we always mean a parallel algorithm from [L86] which places the maximal independent set problem in  $NC^2$ .

---

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

---

**In:**  $k$ -graph  $H$  with  $n \geq n_0$ ,  $n \equiv r \pmod{k}$  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 a MIS algorithm to the intersection graph of  $H$ . Let  $W := V(H) \setminus V(M_1)$ .
2. Repeat while  $|W| > r$ 
  - (a) Arbitrarily divide  $W$  into  $t := \lfloor \frac{|W|}{(k-1)k} \rfloor$  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 a parallel 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  $W := V(H) \setminus V(M_1)$ .
3. Return  $M_1$ .

---

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

**Fact 6.** Any maximal matching  $M_2$  in the bipartite graph  $G$  defined in the algorithm saturates every vertex of  $V_2$ , that is,  $V(M_2) \supseteq V_2$ .

*Proof.* Fix a set  $S \in V_2$  and divide it into  $k$  disjoint subsets of  $k-1$  vertices each,  $S_1, \dots, S_k$ . Next, for each  $e \in M_1$ , let

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

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  $(e' \cap e, S_j)$  cannot all involve the same vertex of  $e$ , or the same set  $S_j$ , 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 remains to show that  $b \geq |V_2|$ . Once we prove that  $\deg_G(S) \geq |V_2|$  for all  $S \in V_2$ , it will follow trivially that any maximal matching  $M_2$  of  $G$  contains all vertices of  $V_2$ .

Recall the definition of the set  $W$  and the number  $t = |V_2|$  given in the algorithm. The number of vertices outside  $V(M_1) \cup \bigcup_{S \in \mathcal{S}} S$  is

$$n - k|M_1| - t(k-1)k < (k-1)k.$$

Since the expression on the left hand side equals  $r$  modulo  $k$  and  $r \geq (k-2)k$ , we have

$$n - k|M_1| - t(k-1)k \leq r. \quad (1)$$

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,  $\pi_S \leq ka + k^2b = k|M_1| + (k^2 - k)b$ ; on the other hand, by the co-degree assumption on  $H$ ,  $\pi_S \geq k\delta(H) \geq n - r$ . Therefore

$$b \geq \frac{n - r - k|M_1|}{(k-1)k} \geq t = |V_2|,$$

where the last inequality follows by (1).

To see that `ALGORITHM PAR-LARGEDEFMATCH`( $r$ ) finds an  $r$ -deficient matching, let  $a_i$  denote the number of unsaturated by  $M_1$  vertices at the start of the  $i$ th loop of the algorithm in Step 2. Let  $t_i$  be the value of  $t$  computed in the  $i$ th loop. Note that  $t_i(k-1)k \geq a_i - r$  and that exactly  $t_i k$  more vertices become saturated after step  $i$ . Note also that  $a_i - r$  is divisible by  $k$ . Thus, for each  $i \geq 1$ ,

$$a_{i+1} - r = a_i - t_i k - r \leq \frac{k-2}{k-1}(a_i - r),$$

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

Note that in the case of graphs discussed in [DKH93], only one iteration in step 2 was sufficient, saving one logarithmic factor in time complexity.

## 2.2 Proof of Theorem 4

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  $r_i \equiv n \pmod{k}$ ,  $i = 1, 2$ , 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 [RRS09], we will need another parallel subroutine which computes a so called powerful matching. Its success relies on the fact that if  $r > 0$  and  $n \equiv r \pmod{k}$  then any matching with deficiency larger than  $r$  must necessarily have at least  $k + 1$  unsaturated vertices.

We now recall the necessary definitions after [RRS09].

**Definition 7** (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  $|e'' \cap e| = k - 2$ .*

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 there are  $\Theta(n^k)$  of them for every set  $S$  in the input hypergraph  $H$  (see Fact 2.2 in [RRS09]).

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

**Definition 8** (powerful matching). *A matching  $M$  in a  $k$ -graph  $H$  with 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  $k - 2$ .*

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 the families  $\mathcal{F}_S$  of all matchings of size  $k - 2$  consisting of  $k - 2$   $S$ -absorbing edges, where  $S$  runs through all subsets of vertices of size  $k + 1$ . The  $xy$  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  $D$  is the desired powerful matching in  $H$ . This can be done efficiently in parallel if

$$\deg_G(x) \geq c|Y| \text{ for all } x \in X \text{ and } \Delta(G[Y]) = o\left(\frac{1}{\log n}|Y|\right). \quad (2)$$

---

ALGORITHM PAR-INDDOMSET

---

**In:** graph  $G = (X \cup Y, E)$ ,  $G[X] = \emptyset$ , satisfying (2)

**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  $\deg_G(y, X) \geq \frac{c}{2}|X|$ ;

- (b) Set  $D := D \cup \{y_0\}$ ;  $X := X \setminus \{x : xy_0 \in E\}$ ,  
 $Y := Y \setminus (\{y_0\} \cup \{y : yy_0 \in E\})$

2. Return  $D$ .

---

Note that at every step,  $X$  decreases by at least a  $\frac{c}{2}$ -fraction, and so, it becomes empty after at most  $O(\log n)$  steps. Thus, throughout the algorithm the set  $Y$  maintains the size of at least  $|Y| - o(|Y|)$  which implies that for each  $x \in X$ ,  $\deg_G(x) \geq \frac{c}{2}|Y|$ , which, in turn, guarantees the existence of  $y_0$ .

Finally, note that  $|X| = \binom{n}{k+1} = \Theta(n^{k+1})$ ,  $\deg_G(x) = \Theta(n^{(k-2)k})$ ,  $|Y| = \Theta(n^{(k-2)k})$ , and  $\deg_G(y, Y) \leq n^{(k-2)k-1}$ , so that (2) holds.

Now, we are in position to describe our algorithm.

---

ALGORITHM PAR-SMALLDEFMATCH( $r$ ),  $0 < r \leq k$

---

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

**Out:**  $r$ -deficient matching  $M$

1. Compute a powerful matching  $M_0$  ( $|M_0| \leq \frac{1}{k}C \log n$ ), as in Definition 8, 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  $H'$ .
4.  $T := V(H) \setminus (V(M_1) \cup V(M_0))$ , [notice that  $|T| = k(k-2) + r$ ].
5. Repeat until  $|T| = r$ : [ $k-2$  sequential iterations]
  - (a) for an arbitrary set  $S \subseteq T$ ,  $|S| = k+1$ , and an  $S$ -absorbing edge  $e \in M_0$ , set  $M_0 := M_0 \setminus \{e\} \cup \{e', e''\}$ , where  $e', e''$  are as in Definition 7;
  - (b)  $T := V(H) \setminus V(M_0' \cup M_1)$ .
6. Return  $M := M_0 \cup M_1$ .

---

It is clear that the above algorithm returns an  $r$ -deficient matching in a polylogarithmic number of steps.

### 2.3 Proof of Theorem 5

In our construction we will apply an *absorbing configuration* motivated by the proof in [RRSS08].

**Definition 9** (absorbing configuration). Given a set  $S = \{x_1, x_2, \dots, x_k\}$  of  $k$  vertices, a triplet of vertex disjoint edges  $e_1, e_2, e_3 \in H$  is called  *$S$ -absorbing configuration* if there are four disjoint edges  $f_1, f_2, f_3$  and  $f_4$  in  $H$  such that  $f_1 \cap e_1 = \{v\}$ ,  $|f_1 \cap e_2| = k-1$ ,  $f_2 \cap e_1 = \{w\}$ ,  $f_2 = \{u\} \cup \{x_1, \dots, x_{k-1}\}$  and  $f_3 \cap e_3 = T$  and  $f_4 = \{x_k\} \cup (e_3 - T) \cup (e_1 - \{v, w\})$ .

---

ALGORITHM PAR-PERFECTMATCH

---

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

**Out:** perfect matching  $M$

1. Compute a  $k$ -deficient matching  $M_1$  using the parallel algorithm PAR-SMALLDEFMATCH( $k$ ) in  $H$ .

2.  $T := V(H) - V(M_1)$ .
3. For every triple of edges  $e_1, e_2, e_3 \in M_1$ , **in parallel** check if they span an absorbing configuration as in Definition 9.
4. Use an absorbing configuration found in step 3 to absorb the vertices of  $T$  and obtain a perfect matching  $M$ .
5. Return  $M$ .

---

The existence of an absorbing configuration in a hypergraph  $H$  with  $\delta(H) \geq \frac{n}{k} + \frac{k}{4}$ , searched for in step 3, is guaranteed by the proof in [RRSS08].

### 3 Toward understanding the hardness gap (the proof of Theorem 2)

Here we give a sketch of the proof of Theorem 2. We will describe a decision algorithm PERFECTMATCH, which is based on a modification of the proof from [RRS09]. With some extra work, it can be turned into a search algorithm.

Given (not necessarily disjoint) sets  $N_1, \dots, N_k \subseteq V(H)$ , denote by  $E_H(N_1, \dots, N_k)$  the set of ordered  $k$ -tuples of *distinct* vertices  $(v_1, \dots, v_k)$  such that  $v_i \in N_i$ ,  $i = 1, \dots, k$ , and  $\{v_1, \dots, v_k\} \in H$ . For  $\gamma > 0$ , let  $A(\gamma) = \{(v_1, \dots, v_{k-1}) : \deg_H(v_1, \dots, v_{k-1}) > (\frac{1}{2} + 2\gamma)n\}$ , and consider the following pair of conditions.

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

Claim 5.2 in [RRS09] 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 [RRS09],  $\gamma$  was chosen to be  $\frac{1}{\log n}$  but it was also observed that the proof goes through for a sufficiently small constant  $\gamma > 0$ . We will take  $\epsilon := \gamma$ .

We say that a partition  $V(H) = A \cup B$  is *even-complete* [*odd-complete*] if for all even [odd]  $r$ , the subhypergraphs  $E_r := E_r(A, B)$  and  $K_r(A, B)$  differ only by an  $\epsilon'$ -fraction of edges, where  $\epsilon'$  is a function of  $\epsilon$  which tends to zero when  $\epsilon$  does. (For the definitions of  $E_r := E_r(A, B)$  and  $K_r(A, B)$ , as well as of  $c$ -small vertices in  $E_r$ , see [RRS09], Section 4.1)

---

#### ALGORITHM PERFECTMATCH

---

**In:**  $k$ -graph  $H$  with  $\delta(H) \geq (\frac{1}{2} - \epsilon)n$  and  $n \geq n_0$ ,  $n \equiv 0 \pmod{k}$ .

**Out:** YES if  $H$  has a perfect matching, NO otherwise.

1. If (i) or (ii) hold, return YES.
2. Otherwise, let  $A := N_H(S_1)$ ,  $B := V \setminus A$ , where  $(S_1, \dots, S_k)$  is a  $k$ -tuple violating (i).
3. If  $k$  is odd and  $(A, B)$  is odd-complete, swap  $A$  and  $B$  around;
4. If  $(A, B)$  is 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  $\bigcup_{r \text{ odd}} E_r \neq \emptyset$ , return YES
  - (c) Return NO
5. If  $(A, B)$  is 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  $|A| \equiv \frac{n}{k} \pmod{2}$  or  $\bigcup_{r \text{ even}} E_r \neq \emptyset$ , return YES.
  - (c) Return NO.

---

We will now verify the correctness of the algorithm. 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 4(c)  $|A|$  is odd, while all edges of  $H$  intersect  $A$  in an even number of vertices. In step 5(c),  $|A| \not\equiv \frac{n}{k} \pmod{2}$  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 even number of vertices of  $A$ , and so  $|V(M) \cap A| \equiv \frac{n}{k} \pmod{2}$ , a contradiction.

It is a lot harder to verify the correctness of the YES answers. Because of the lack of space we will give that part of the proof in a full version of the paper.

## References

- [AFS08] A. Asadpour, U. Feige and A. Saberi. Santa Claus Meets Hypergraph Matchings Proceedings of APPROX-RANDOM 2008: 10–20.
- [DK92] E. Dalhaus, M. Karpiński, Perfect matching for regular graphs is  $AC^0$ -hard for the general matching problem, J. of Computer and System Sciences, Volume 44, Issue 1, (1992), 94 – 102
- [DKH93] E. Dalhaus, P. Hajnal, M. Karpiński, On the parallel complexity of Hamiltonian cycle and matching problem on dense graphs, J. Algorithms 15 (1993)
- [Ed65] J. Edmonds, Paths, trees and flowers, Canad J. Math. 17 (1965), 449-467
- [GJ79] M.R. Garey, D.S, Johnson, Computers and intractability, Freeman, 1979
- [H95] P.E. Haxell. A Condition for Matchability in Hypergraphs, Graphs and Combinatorics (1995), Vol 11, 245-248.
- [KO06] D. Kühn, D. Osthus, Critical chromatic number and the complexity of perfect packings in graphs, 17th ACM-SIAM Symposium on Discrete Algorithms 2006 (SODA), 851-859
- [L86] M. Luby, A simple parallel algorithm for the maximal independent set problem, SIAM J. Comput., vol 15(4), 1986, pp. 1036-1053.
- [RRS09] V. Rödl, A. Ruciński, E. Szemerédi, Perfect matchings in large uniform hypergraphs with large minimum collective degree, JCT A 116(3) (2009) 613-636.
- [RRSS08] 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): 633-636, (2008)
- [S09] E. Szymańska, The Complexity of Almost Perfect Matchings in Uniform Hypergraphs with High Codegree, Proc. of IWCCA 2009, to appear.
- [Sa09] G. Särközy, A fast parallel algorithm for finding Hamiltonian cycles in dense graphs. Discrete Mathematics 309, 2009, pp. 1611–1622.