University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
Z. Parsaeian, G. Schmid

Distributed Graph Algorithms
Exercise Sheet 4

The Subgraph Detection Problem

In general, the *subgraph detection problem* is defined as follows. We are given some graph S of fixed size and the task is to find out if S is a subgraph of the network graph G. The nodes only have two possible outputs: 'yes' or 'no'. If G contains some copy of S then *at least one node* must output 'yes'. Otherwise, *all nodes* must output 'no'. Note that the nodes do not need to explicitly output a copy of S in G. In the following, we are considering the subgraph detection problem for the subgraph $S = C_4$, i.e., when S is a cycle of length 4. That is, we have to determine if G contains at least one 4-cycle or if G is has no 4-cycles (in which case we call G C_4-free).

Exercise 1: An Algorithm for C_4-Detection *(10 Points)*

In the first exercise, we construct a $\mathsf{CONGEST}$ model algorithm to solve the C_4-detection problem in $O(\sqrt{n})$ rounds. Assume that the network graph is $G = (V, E)$. Also assume that all nodes of G know the exact value of n.

(a) We start with a basic graph-theoretic statement. Let $G = (V, E)$ be an n-node graph and assume that for a node $u \in V$, $N(u)$ denotes the set of neighbors of u and $\deg(u) = |N(u)|$ is the degree of u. Show that if the following property is true for some node $u \in V$, then u is contained in some 4-cycle in G:

$$\sum_{v \in N(u)} (\deg(v) - 1) \geq n.$$

In the following, we partition the set of nodes V into the low-degree node V_L and the high-degree node V_H as follows:

$$V_L := \{v \in V \,:\, \deg(v) < \sqrt{n} + 1\}, \quad V_H := V \setminus V_L = \{v \in V \,:\, \deg(v) \geq \sqrt{n} + 1\}.$$

The algorithm now works as follows. First, every node $v \in V_H$ that has at least $\sqrt{n}$ neighbors in V_H outputs 'yes'. Then, every node in V_L sends the list of neighbor IDs to all its neighbors. Further, every node $v \in V_H$ that has at most $\sqrt{n}$ neighbors in V_H sends the list of its V_H-neighbors to all its neighbors. Every node that learns all four edges of some 4-cycle outputs 'yes'. All remaining nodes output 'no'.

(b) Show that the described algorithm can be implemented in $O(\sqrt{n})$ rounds in the $\mathsf{CONGEST}$ model.

(c) Show that if at least one node outputs 'yes', then G contains at least one 4-cycle. You can use the result of (a) to show this.

(d) Show that if G does contains a 4-cycle, then at least one node outputs 'yes'.

Hint: Show that four every 4-cycle, either the cycle contains a node $v \in V_H$ with at least $\sqrt{n}$ neighbors in V_H (and then v outputs 'yes') or one of the nodes of the cycle learns about the existence of all four edges of the cycle.

Exercise 2: A Lower Bound for C_4-Detection *(10 Points)*

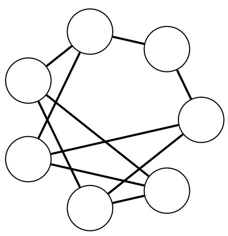
The goal of this exercise is to show that the algorithm of Exercise 1 is almost tight by showing that $\Omega(\sqrt{n}/\log n)$ rounds are needed to solve the C_4-detection problem in the $\mathsf{CONGEST}$ model. For solving the exercise, you can use the following two facts.

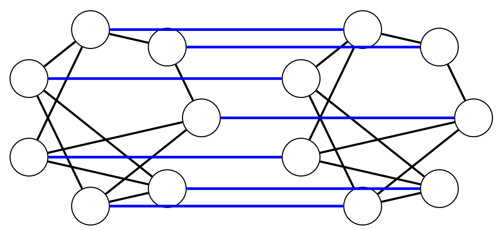
Communication Complexity of Set Disjointness: Assume that two players Alice and Bob obtain private inputs X and Y, where both X and Y are subsets of the set $\{1,\ldots,K\}$ for some given integer parameter K. Alice and Bob have to determine if X and Y are disjoint. If $X \cap Y = \emptyset$, they both have to output 'yes' and otherwise they both have to output 'no'. To achieve this, Alice and Bob can communicate over a bidirectional communication channel. It is known that even if Alice and Bob can use randomization and even if they only need to succeed with probability 2/3, then Alice and Bob have to exchange $\Omega(K)$ bits (in expectation).

Dense C_4-Free Graphs: There exists a constant $c > 0$ such that for every positive integer N, there exists a C_4-free N-node graph H that has at least $c \cdot N^{3/2}$ edges.

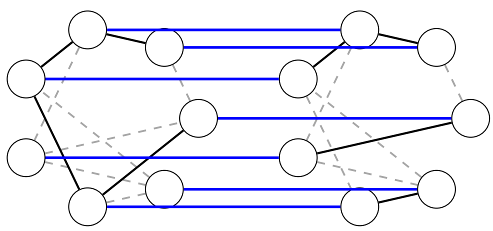
Lower Bound Graph Construction: To obtain a lower bound, we have to construct a family of graphs on which C_4-detection is hard. To achieve this, we pick some positive integer N and we construct a collection $\mathcal{G}_N$ of graphs consisting of $n = 2N$ nodes. The graphs consist of the nodes $U = \{u_1, u_2, \ldots, u_N\}$ and $V = \{v_1, v_2, \ldots, v_N\}$. Let H be a C_4-free N-node graph with exactly $\lceil c \cdot N^{3/2} \rceil$ edges for some constant $c > 0$. We use two disjoint copies H_1 and H_2 of H such that the set of nodes of H_1 is U and the set of nodes of H_2 is V. The nodes in H_1 and H_2 are labeled in a consistent manner such that for every $i, j \in \{1,\ldots,N\}$, $\{u_i, u_j\}$ is an edge of H_1 if and only if $\{v_i, v_j\}$ is an edge of H_2. The graph $G = (U \cup V, E)$ of the family $\mathcal{G}_N$ are now constructed as follows. For every $i \in \{1,\ldots,n\}$, E contains an edge between u_i and v_i (i.e., U and V are connected by a perfect matching). In addition, E consists of an arbitrary subset of the edges of H_1 and H_2. An illsutration is given below (not with the correct H graph, but a simpler one for illusstration purposes).

(1) A C_4-free graph H **(2) Two copies H_1, H_2 with matching**





(3) Take a subset of the edges (dashed edges do were not chosen)



(a) Assume that we are given sets $X, Y \subseteq \{1, \ldots, \lceil c \cdot N^{3/2} \rceil\}$. Show that you can construct a graph $G = (U \cup V, E) \in \mathcal{G}_N$ such that the edges among nodes in U only depend on X, the edges among node in V only depend on Y and such that G is C_4-free if and only if X and Y are disjoint.

(b) To prove the desired lower bound for C_4-detection, assume (for the sake of contradiction) that there exists a $\mathsf{CONGEST}$ algorithm $\mathcal{A}$ that solves the C_4-detection problem in graphs of $\mathcal{G}_N$ in $o(\sqrt{N}/\log N) = o(\sqrt{n}/\log n)$ rounds. Show that if Alice and Bob are given sets $X, Y \subseteq \{1, \ldots, K\}$ for $K = \lceil c \cdot N^{3/2} \rceil$, they can use this protocol to determine if X and Y are disjoint by communication $o(K)$ bits (which is in contradiction to the known lower bound on the communication needed to solve set disjointness).